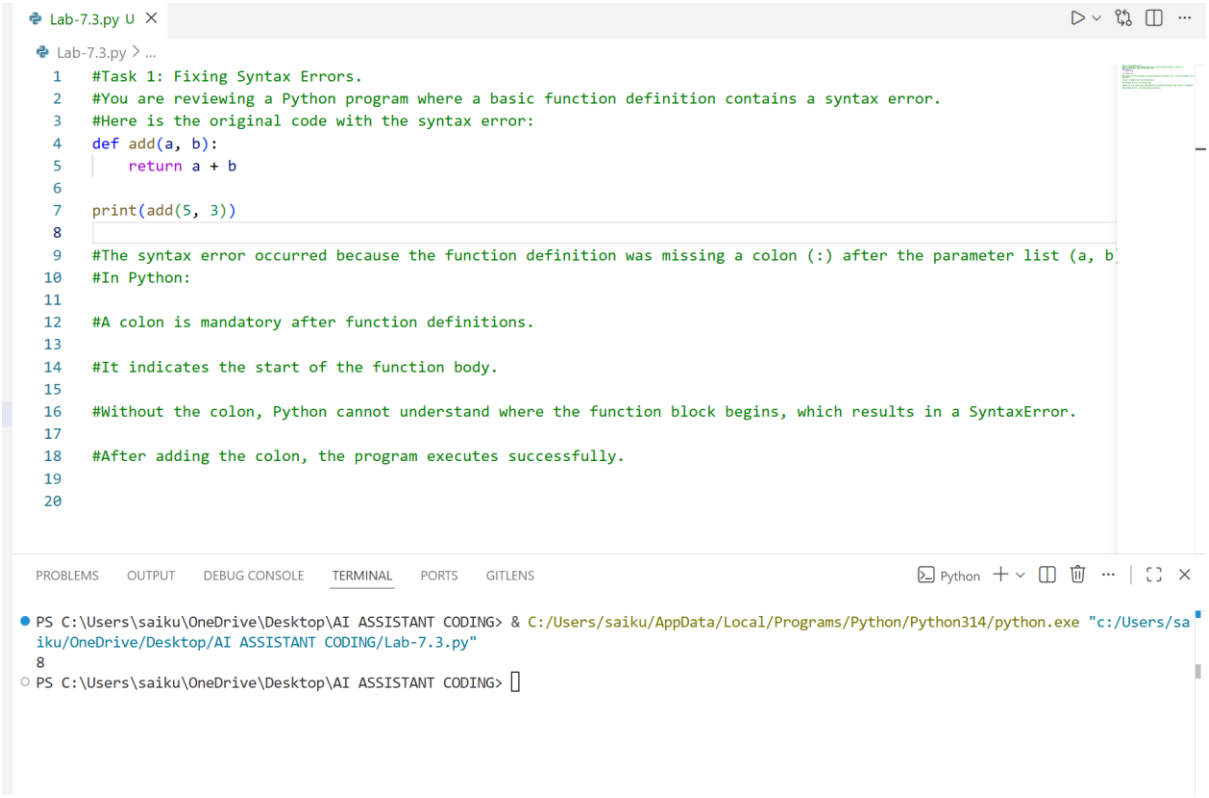
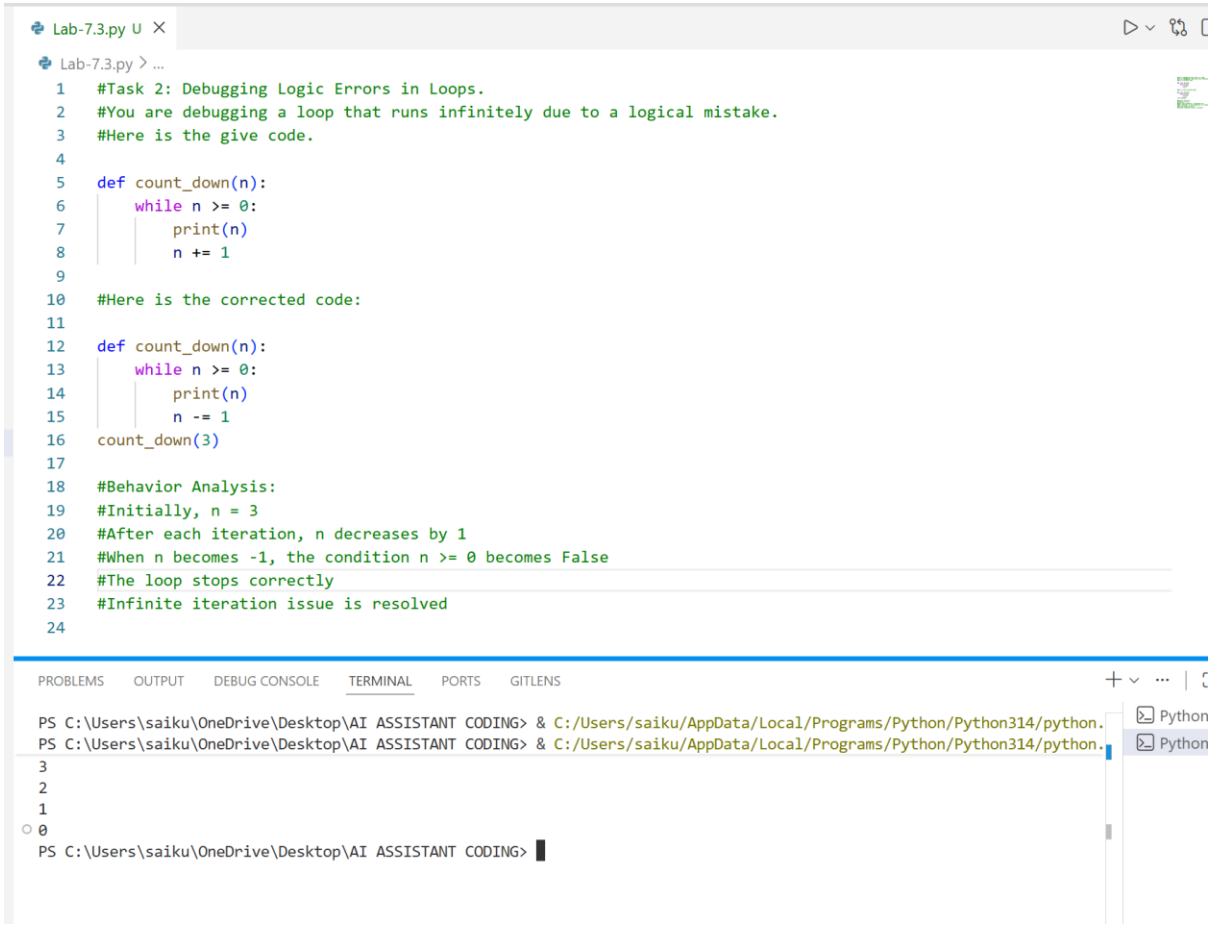
AI ASSISTED CODING LAB-7.3

J.Sai Kumar || Batch:-09 || 2303A51562

Task 1: Fixing Syntax Errors.



Task 2: Debugging Logic Errors in Loops.



Task 3: Handling Runtime Errors (Division by Zero).

Lab-7.3.py U

Lab-7.3.py > ...

```python
#Task 3: Handling Runtime Errors (Division by Zero).
#A Python function crashes during execution due to a division by zero error.
#Here is corrected code with error handling for division by zero:
def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return "Error: Cannot divide by zero"

print(divide(10, 0))

#How the AI Fixed It:
#The risky statement a / b is placed inside a try block.

#If division by zero occurs, Python raises ZeroDivisionError.

#The except block catches the error.

#Instead of crashing, the function returns a user-friendly message.
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/saiku/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/saiku/OneDrive/Desktop/AI ASSISTANT CODING/Lab-7.3.py"
Error: Cannot divide by zero
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING>
```

Task 4: Debugging Class Definition Errors.

Lab-7.3.py U

Lab-7.3.py > ...

```python
#Task 4: Debugging Class Definition Errors.
#The following Python class gives an error when creating an object.
#Identify the issue in the constructor and correct the code.
#Also explain why the fix is required.
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def display(self):
        print("Length:", self.length)
        print("Width:", self.width)

# Creating object
r1 = Rectangle(10, 5)
r1.display()

#AI Explanation of the Object-Oriented Error:
#self represents the current instance (object) of the class.
#It allows access to instance variables and methods.
#When an object is created, Python automatically passes the object as the first argument.
#If self is missing, Python throws a TypeError.
#Without self, the object cannot store or access its own data.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/saiku/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/saiku/OneDrive/Desktop/AI ASSISTANT CODING/Lab-7.3.py"
Length: 10
Width: 5
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING>
```

Task 5: Resolving Index Errors in Lists.

Lab-7.3.py U

Lab-7.3.py > ...

```python
#Task 5: Resolving Index Errors in Lists.
#The following Python code crashes when executed.
#Identify the error and modify the code to safely access the list.
numbers = [1, 2, 3]
index = 5

if index < len(numbers):
    print(numbers[index])
else:
    print("Error: Index out of range")

#AI Explanation of the Error Handling

#Lists in Python use zero-based indexing.

#If an index greater than len(list) - 1 is accessed, Python raises an IndexError.

#Using bounds checking ensures the index is valid before accessing.

#Using try-except prevents the program from crashing.
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/saiku/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/saiku/OneDrive/Desktop/AI ASSISTANT CODING/Lab-7.3.py"
Error: Index out of range
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING>
```