LAB ASSIGNMENT-7.3

2303A51337

BATCH-10

➢ TASK-1:

PROMPT:

I have a Python function with a syntax error. Detect the issue and correct it.

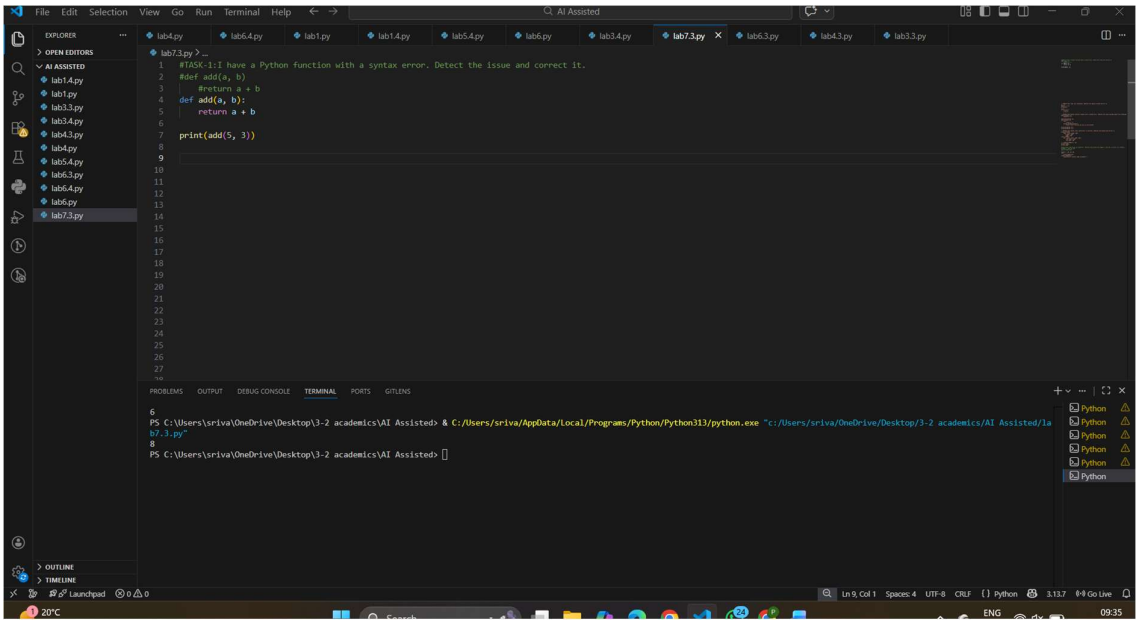
```
def add(a, b)
    return a + b
```

CODE:

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

print(add(5, 3))
```

OUTPUT:



EXPLANATION:

The error is because the function definition is missing a colon and is commented out. It should be written as def add(a, b):. After fixing it, the function runs correctly and returns 8 for add(5, 3).

- TASK-2

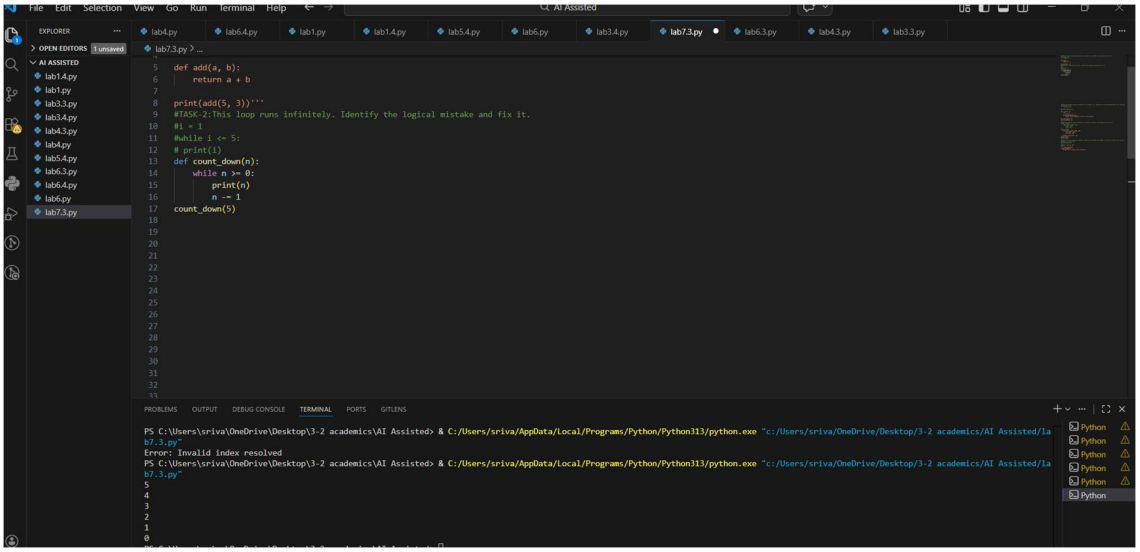
PROMPT:

This loop runs infinitely. Identify the logical mistake and fix it.

CODE:

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

OUTPUT:



EXPLANATION:

The loop runs infinitely because the value of i is never changed inside the loop, so the condition i <= 5 is always true. To fix this, i must be incremented inside the loop using i += 1. After this correction, the loop prints numbers from 1 to 5 and then stops.

❖ TASK-3:

PROMPT:

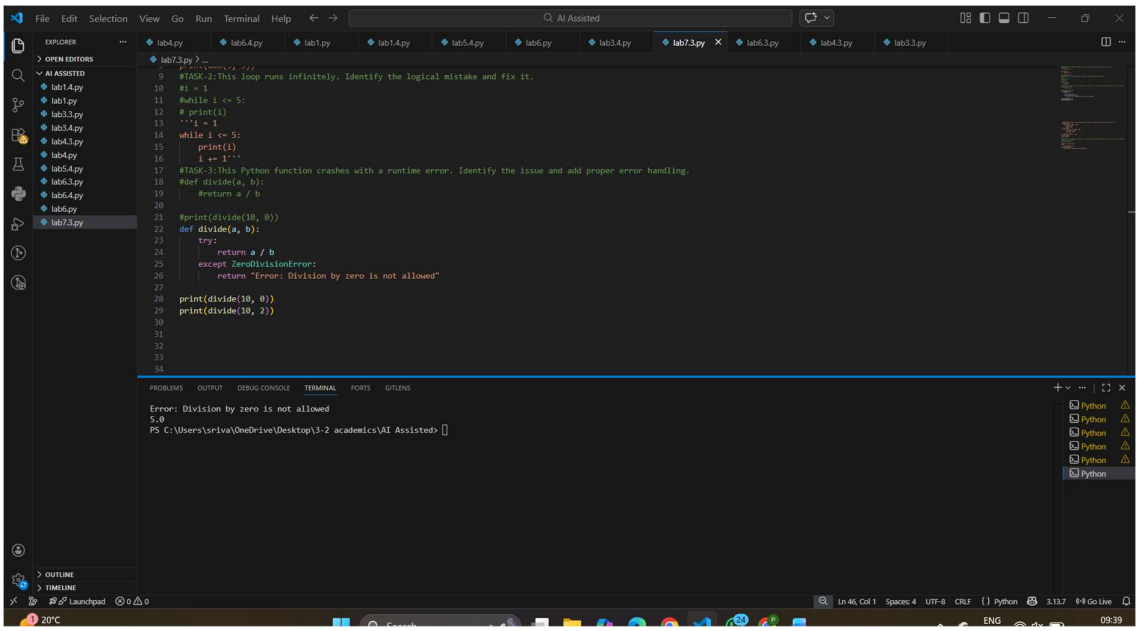
This Python function crashes with a runtime error. Identify the issue and add proper error handling.

def divide(a, b): return a / b print(divide(10, 0))

CODE:

```
def divide(a, b):
  try:
    return a / b
  except ZeroDivisionError:
    return "Error: Division by zero is not allowed"
print(divide(10, 0))
print(divide(10, 2))
```

OUTPUT:



EXPLANATION:

The error occurs because dividing by zero causes a ZeroDivisionError.Using try and except prevents the program from crashing by catching this error and returning a message instead. When the divisor is not zero, the function works normally and returns the correct result.

TASK-4:

PROMPT:

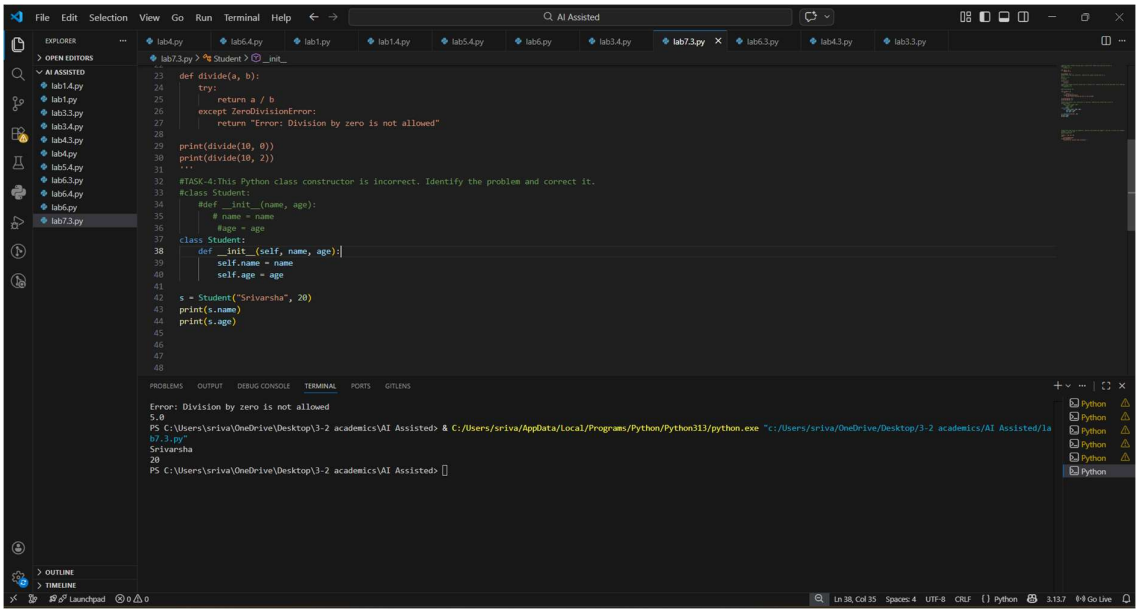
`This Python class constructor is incorrect. Identify the problem and correct it.`

`class Student: def __init__(name, age)  name = name   age = age`

CODE:

```
class Student:
   def __init__(self, name, age):
      self.name = name
      self.age = age
s = Student("Srivarsha", 20)
print(s.name)
print(s.age)
```

OUTPUT:



EXPLANATION:

The constructor is incorrect because it does not use self, so the variables name and age are not stored in the object. self is required to refer to the current object, and attributes must be assigned as self.name and self.age. After fixing this, the constructor correctly initializes the student's name and age, and the object can access them properly.

❖ Task-5

PROMPT:

This code throws an IndexError. Identify the problem and suggest a safe way to access list elements.

#numbers = [10, 20, 30]

#print(numbers[5])

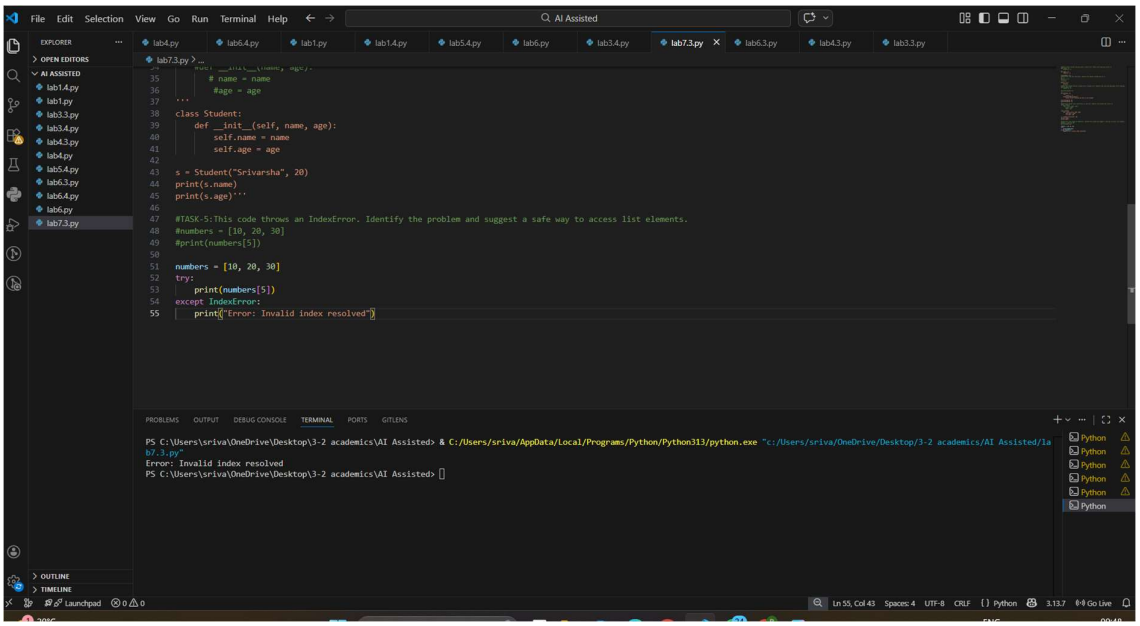
CODE:

```
numbers = [10, 20, 30]

try:
    print(numbers[5])

except IndexError:
    print("Error: Invalid index resolved")
```

OUTPUT:



EXPLANATION:

The error occurs because the list has only 3 elements (indexes 0, 1, and 2), but the code tries to access index 5, which does not exist. This causes an IndexError. Using a try–except block safely handles this situation by catching the error and printing a message instead of crashing the program.