

ASSIGNMENT-7.3

2303A51546

Batch-10

TASK-1:

Prompt:

I wrote a Python function, but it is showing a syntax error.

Please identify the syntax mistake, correct the function definition, and explain what caused the error.

Wrong code with syntax error:

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

CODE:

```
def add(a, b):
    return a + b
print(add(5, 3))
```

Output:

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists several Python files: assignment-1.py, Assignment-3.3.py, lab-1, LAB-1-4.py, Lab-3-4.py, Lab-4-3.py, Lab-5-4.py, lab-6-3.py, and LAB-6-4.py. The file Lab-7.3.py is currently selected. The main editor area displays the following Python code:

```
#TASK-1
def add(a, b):
    return a + b
print(add(5, 3))
```

The Terminal pane at the bottom shows the output of running the code:

```
PS C:\Users\srini\OneDrive\Desktop\AI Assisted> & "c:\Users\srini\AppData\Local\Programs\Python\Python313\python.exe" 'c:\Users\srini\.vscode\extensions\ms-python.python.debugpy-2025.1.8\win32-x64\bundledlibs\debugpy\launcher' '60796' '--' 'c:\Users\srini\OneDrive\Desktop\AI Assisted\Lab-7.3.py'
8
PS C:\Users\srini\OneDrive\Desktop\AI Assisted>
```

Analysis:

In Python, every function definition must end with a colon (:).

The colon tells Python that the indented block below belongs to the function.

Without the colon, Python raises a SyntaxError and the program cannot run.

The AI correctly detected the missing colon and fixed the function.

TASK-2:

Prompt:

I wrote a Python function that is supposed to count down to 0, but it runs forever. Please find the logic error causing the infinite loop, correct the code, and explain the mistake.

Wrong code (Debugging logic errors in loops):

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

CODE:

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

```
count_down(5)
```

Output:

```

File Edit Selection View Go Run Terminal Help < > Q AI Assisted
EXPLORER ... Welcome Lab-7.3.py U Lab-3.4.py U
AI ASSISTED
assignment-1.py U
Assignment-3.3.py U
lab-1
LAB-1.4.py U
Lab-3.4.py U
Lab-4.3.py U
Lab-5.4.py U
lab-6.3.py U
LAB-6.4.py U
Lab-7.3.py U
TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS
Python Debug Console + × ⌂ ...
8.0-win32-x64\bundled\libs\debugpy\launcher' '53196' '--' 'c:\Users\smini\OneDrive\Desktop\AI Assisted\Lab-7.3.py'
5
4
3
2
1
0
PS C:\Users\smini\OneDrive\Desktop\AI Assisted>
Ln 4, Col 20 Spaces: 4 UFT-8 CRLF { } Python 3.13.7 ⌂ Go Live

```

Analysis:

The loop was intended to count down, but the variable was being incremented instead of decremented.

Because n kept increasing, it never became less than 0, so the loop never stopped.

Changing n += 1 to n -= 1 correctly decreases the value each time, allowing the loop to terminate.

TASK-3:

Prompt:

The following Python function crashes during execution.

Please identify the runtime error, fix the code using a try-except block, and explain how your solution prevents the crash.

Wrong code (handling runtime errors division by zero):

```

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

```

CODE:

```

def divide(a, b):
    try:
        return a / b

```

```
except ZeroDivisionError:  
    return "Error: Cannot divide by zero"
```

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

Output:

The screenshot shows the Microsoft Visual Studio Code interface. The left sidebar contains icons for Explorer, Search, Find, Open, Save, Undo, Redo, and Terminal. The Explorer view shows a folder named 'AI ASSISTED' containing several Python files: assignment-1.py, Assignment-3.3.py, lab-1, LAB-14.py, Lab-34.py, Lab-43.py, Lab-54.py, Lab-63.py, LAB-64.py, and Lab-7.3.py. The 'Lab-7.3.py' file is currently selected and open in the main editor area. The code in the editor is as follows:

```
11 count_down(5) ***  
12  
13  
14  
15 #TASK-3  
16 def divide(a, b):  
17     try:  
18         return a / b  
19     except ZeroDivisionError:  
20         return "Error: Cannot divide by zero"  
21  
22 print(divide(10, 0))  
23  
24  
25  
26
```

The bottom right corner of the code editor has a small note from AI Assisted: "I can help you with this code". Below the editor is the terminal window, which shows the output of running the code. The terminal title is 'Python Debug Console'. The output text is:

```
PS C:\Users\srini\OneDrive\Desktop\AI Assisted> & "c:\Users\srini\AppData\Local\Programs\Python\Python313\python.exe" "c:\Users\srini\.vscode\extensions\ms-python.python\2025.8.0-win32-x64\bundled\libs\debug\launcher" "52929" -- "c:\Users\srini\OneDrive\Desktop\AI Assisted\Lab-7.3.py"  
Error: Cannot divide by zero  
PS C:\Users\srini\OneDrive\Desktop\AI Assisted>
```

Analysis:

Division by zero is not allowed in Python and results in a runtime error.

A try block is used to run the risky code.

If a `ZeroDivisionError` occurs, the `except` block handles it safely.

Instead of crashing, the program now shows a user-friendly error message.

TASK-4:

Prompt:

I wrote this Python class, but it is not working correctly.

Please find the mistake in the constructor, fix the class definition, and explain why self is required.

Wrong code (Debugging Class Definition Errors):

```
class Rectangle:
```

```
def __init__(length, width):
```

```
self.length = length
```

```
self.width = width
```

CODE:

```
class Rectangle:
```

```
    def __init__(self, length, width):
```

```
        self.length = length
```

```
        self.width = width
```

```
# Creating an object
```

```
r1 = Rectangle(10, 5)
```

```
print("Length:", r1.length)
```

```
print("Width:", r1.width)
```

Output:

The screenshot shows the Visual Studio Code interface. The code editor displays a Python script named 'Lab-7.3.py'. The script defines a class 'Rectangle' with an __init__ method that initializes length and width. It then creates an object 'r1' of type 'Rectangle' with parameters 10 and 5, and prints the length and width. The terminal below shows the execution of the script and its output: Length: 10 and Width: 5.

```
File Edit Selection View Go Run Terminal Help < - > Q AI Assisted
EXPLORER ... Welcome Lab-7.3.py U Lab-3.4.py U
AI ASSISTED assignment-1.py U Assignment-3.3.py U
lab-1 LAB-1.4.py U Lab-3.4.py
Lab-4.3.py U Lab-5.4.py U lab-3.py U LAB-6.4.py U
Lab-7.3.py U
# TASK-4
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width
# Creating an object
r1 = Rectangle(10, 5)
print("Length:", r1.length)
print("Width:", r1.width)

TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS
PS C:\Users\smini\OneDrive\Desktop\AI Assisted> & 'c:\Users\smini\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\smini\.vscode\extensions\ms-python.python-2025.1.8.0-win32-x64\bundled\libs\debugpy\launcher' '56935' '--' 'c:\Users\smini\OneDrive\Desktop\AI Assisted\Lab-7.3.py'
Length: 10
Width: 5
PS C:\Users\smini\OneDrive\Desktop\AI Assisted>
In 22, Col 24 Spaces:4 UTF-8 CRLF () Python 3.13.7 ⓘ Go Live
```

Analysis:

self represents the **current object** of the class.

When we create an object (r1 = Rectangle(10, 5)), Python automatically passes that object as self.

Using self.length and self.width stores the values inside the object.

Without self, the variables are not linked to the object, leading to errors.

TASK-5:

Prompt:

This Python program crashes with an index error when accessing a list element. Please explain why the error occurs and modify the code to safely access the list using bounds checking or exception handling.

Wrong code (Resolving Index Errors in Lists):

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

CODE:

```
numbers = [1, 2, 3]
```

```
index = 5
```

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

Output:

```
File Edit Selection View Go Run Terminal Help <- > Q AI Assisted
EXPLORER ... Welcome Lab-7.3.py U Lab-3.4.py U
AI ASSISTED
assignment-1.py U
assignment-3.3.py U
lab-1
LAB-1.4.py U
Lab-3.4.py U
Lab-4.3.py U
Lab-5.4.py U
lab-6.3.py U
LAB-6.4.py U
Lab-7.3.py U
35
36 #TASK-5
37 numbers = [1, 2, 3]
38 index = 5
39
40 if index < len(numbers):
41     print(numbers[index])
42 else:
43     print("Error: Index out of range")
44
45
46
47
48
49

TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS
Python Debug Console + × ⚡ ... ×
PS C:\Users\sriini\OneDrive\Desktop\AI Assisted> & 'c:\Users\sriini\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\sriini\.vscode\extensions\ms-python.python-2025.1.0-win32-x64\bundled\libs\debugpy\launcher' '57685' '--' 'c:\Users\sriini\OneDrive\Desktop\AI Assisted\Lab-7.3.py'
Error: Index out of range
PS C:\Users\sriini\OneDrive\Desktop\AI Assisted>

Ln 44, Col 1  Spaces: 4  UTF-8  CRLF  {} Python ⚡ 3.13.7 ⚡ Go Live
```

Analysis:

Python lists have a fixed range of valid positions based on their length.

Trying to access an element outside that range raises an IndexError.

We can prevent crashes by:

Checking the index using `len(list)`

Using a try-except block to catch the error safely