

2303a51924

## Lab assignment – 7.3

### Task 1: Fixing Syntax Errors

#### Scenario

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

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, and other standard options. The title bar shows the current file is "lab assignment 7.3.py". The main editor area displays the following Python code:

```
1 #-----1
2 def add(a, b):
3     return a + b
4
5 # Syntax: Python uses colons to introduce new code blocks (like functions, loops, and if-statements).
6
7 # Comparison: Think of the colon as the word "do" or "then." By writing def add(a, b):, you are essentially saying: "Define a function named add with these parameters, and then do the following."
```

A red squiggly underline is under the colon on line 2, indicating a syntax error. Below the editor, the terminal window shows the command "python.exe" being run and the resulting error message:

```
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python311/python.exe "d:/AI Coding/lab assignment 7.3.py"
File "d:/AI Coding/lab assignment 7.3.py", line 2
    def add(a, b):
          ^
IndentationError: expected an indented block after function definition on line 2
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python311/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python311/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding>
```

In the bottom right corner of the terminal, there is a tooltip: "Extension Biscet is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options." There are also icons for powershell, Python, and Python 3.

## Task 2: Debugging Logic Errors in Loops

### Scenario

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

The screenshot shows a code editor interface with a dark theme. At the top, there's a menu bar with File, Edit, Selection, View, Go, Run, and other options. Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main area displays a Python script named `lab assignment 7.3.py`. The code contains a while loop that increments a variable `i` by 1 each time it runs, starting from 1 and ending at 5. The code is annotated with comments explaining its purpose. The terminal tab at the bottom shows the command line output of running the script, which results in an `IndentationError` because there is no indentation block after the function definition on line 2. A tooltip in the bottom right corner indicates that the Extension Bisection feature is active and has disabled 3 extensions, allowing the user to still reproduce the problem.

```
1 #-----2
2 i = 1
3 while i <= 5:
4     print(i)
5     i += 1 # This increments i by 1 each time the loop runs
6 #Start: i is 1.
7
8 #Check: Is i 5? Yes.
9
10 #Action: Print 1.
11
12 #The Loopback: Without i += 1, the code jumps back to the Check step. Since i is still 1, it enters a "Groundhog Day" scenario where it prints 1 forever.
13
14 #By adding i += 1, you ensure that i reaches 2, then 3, then 4, then 5. Once it hits 6, the condition 6 <= 5 becomes False, the gate closes, and the loop finishes.
```

File "d:\AI Coding\lab assignment 7.3.py", line 2  
def add(a, b):  
 ^ IndentationError: expected an indented block after function definition on line 2  
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"  
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"  
PS D:\AI Coding>  
1  
2  
3  
4  
5  
PS D:\AI Coding>

In 14 Col 2 Spaces: 4 LF-CR F Python Chat stats rear

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

#### Scenario

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

The screenshot shows a code editor window titled "AI Coding". The file "lab assignment 7.3.py" contains the following code:

```
1 #-----3
2 def divide(a, b):
3     if b == 0:
4         return "Cannot divide by zero"
5     return a / b
6 #The try block: Python attempts to execute the division. If b is anything other than zero, it runs normally and returns the result.
7
8 #The except block: If a ZeroDivisionError occurs, Python stops the try block immediately and jumps to this section. Instead of the program dying, it returns a helpful message (or a default value like 0 or No
9
```

The terminal below shows the execution of the script:

```
File "d:/AI Coding/lab assignment 7.3.py", line 2
  def add(a, b):
IndentationError: expected an indented block after function definition on line 2
PS D:/AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:/AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
1
2
3
4
5
PS D:/AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
```

A tooltip in the terminal area states: "Extension Bisect is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options." The bottom status bar shows the date and time as 2/5/2026, 12:28 PM, and the weather as 29°C Sunny.

## Task 4: Debugging Class Definition Errors

### Scenario

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

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. In the top left, there's a navigation bar with File, Edit, Selection, View, Go, Run, etc. Below it is a search bar. The main workspace shows a file named "lab assignment 7.3.py" with the following code:

```
1 class Student:
2     def __init__(self, name, roll):
3         self.name = name
4         self.roll = roll
5 #The self parameter represents the current object instance.
6
7 #Without self, Python cannot associate variables with the object.
8
9 #self is required as the first parameter in all instance methods.
```

Below the code editor is a "PROBLEMS" tab, which is currently active. The terminal at the bottom shows a series of identical command-line outputs, each ending in an error message:

```
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 7.3.py"
```

A tooltip in the terminal area says: "Extension Bisect is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options." Below the terminal are status icons for battery, signal, and time (1:30 PM, 2/5/2026).

## Task 5: Resolving Index Errors in Lists

## Scenario

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

The screenshot shows a Jupyter Notebook interface with the following code in cell 7.3.py:

```
# Lab assignment 7.3.py > ...
numbers = [10, 20, 30]
index = 5
if index < len(numbers):
    print(numbers[index])
else:
    print("Index out of range")
# The list has only 3 elements, so index 5 is invalid.
# Accessing an invalid index raises an IndexError.
# Bounds checking or exception handling prevents crashes and improves robustness.
```

The notebook also displays a terminal window with multiple command-line executions of the same script, all resulting in the output "Index out of range". A status bar at the bottom indicates the current file is "lab assignment 7.3.py".