

## ASSIGNMENT – 7

Name: Abhinav

Hall.no: 2303A52174

### Task 1: Fixing Syntax Errors

CODE:



The screenshot shows a code editor with multiple tabs at the top: as1.py, ii.py, u.py, 3\_3.py, ass7.py (which is the active tab), customer\_data, and config. The ass7.py tab has a red badge with the number 4, indicating four errors. Below the tabs, the code for ass7.py is displayed in a code editor window. The first two lines of code are shown: line 1 contains 'def add(a,b)' and line 2 contains 'return a+b'. A blue arrow points from the start of line 1 to the opening parenthesis '(', indicating a potential error. Both lines have underlines underneath them, with the first line's underline ending at the closing parenthesis ')' and the second line's ending at the '+' sign. The code editor interface includes standard icons for file operations and a status bar at the bottom.

Error Identified by AI

Type of Error: SyntaxError

Reason: Python requires a colon (:) at the end of a function definition line

OUTPUT:



The screenshot shows the VS Code terminal window. At the top, there are tabs for PROBLEMS (with 4 notifications), OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The terminal output shows a PowerShell prompt: PS C:\Users\aksha\Downloads\assistent\_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistent\_ai/ass7.py. It then displays an error message: File "c:/Users/aksha/Downloads/assistent\_ai/ass7.py", line 1 def add(a,b) ^ SyntaxError: expected ':' . Below this, it shows another PowerShell prompt: PS C:\Users\aksha\Downloads\assistent\_ai>. The terminal interface includes standard PowerShell icons and a status bar at the bottom.

EXPLANATION:

The AI detected a syntax error because the function declaration line did not end with a colon. In Python, the colon indicates the start of an indented code block. Without it, the interpreter cannot parse the function body.

## Task 2: Debugging Logic Errors in Loops

CODE:

The screenshot shows a code editor interface with multiple tabs at the top: as1.py, ii.py, u.py, 3\_3.py, ass7.py (which is the active tab), and customer\_data. The code in the ass7.py tab is as follows:

```
ass7.py > count_down
1 def count_down(n):
2     while n>=0:
3         print(n)
4     n+=1
5
```

The line `n+=1` is highlighted in red, indicating a syntax error. A blue button labeled "4 →" is visible on the left, and a small yellow box with a minus sign is on the right.

OUTPUT:

The screenshot shows a terminal window with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is the active tab), and PORTS. The terminal output shows the execution of the Python script ass7.py:

```
PS C:\Users\aksha\Downloads\assistant_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistant_ai/ass7.py
● 5
4
3
2
1
0
○ PS C:\Users\aksha\Downloads\assistant_ai>
```

EXPLANATION:

The loop condition depends on i increasing toward a stopping point. Decrementing i causes it to move away from the termination condition, resulting in an infinite loop.

Type of Error: Logical Error

Reason: The loop variable is decremented instead of incremented, so the condition `i <= 5` is always true.

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

CODE:

```
as1.py ii.py u.py 3_3.py ass7.py customer_data
ass7.py > ...
1 #Debug the following code
2 def divide(a,b):
3 → 3 → return a/b
4 print(divide(10,0))
try:
    return a/b
except ZeroDivisionError:
    return "Error: Division by zero is not allowed."
```

Error Identified by AI

Type of Error: Runtime Error (ZeroDivisionError)

Reason: Division by zero is mathematically undefined

OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\aksha\Downloads\assistent_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistent_ai/ass7.py
Error: Division by zero is not allowed.
○ PS C:\Users\aksha\Downloads\assistent_ai>
```

EXPLANATION:

The AI added exception handling using try-except to prevent program termination. This ensures graceful handling of invalid input instead of crashing the program.

#### Task 4: Debugging Class Definition Errors

CODE:

The screenshot shows a code editor with several tabs at the top: as1.py, ii.py, u.py, 3\_3.py, ass7.py, and customer\_d. The current file is ass7.py, which contains the following code:

```
8
9
10 class rectangle:
11     def __init__(self,length,width):
12         self.length=length
13         self.width=width
14 rect=rectangle(10,5)
15 print(rect.length)
16 print(rect.width)
```

Line 12 is highlighted with a blue background, and the assignment statement `self.length=length` is highlighted with a green background.

OUTPUT:

The terminal window shows the output of the Python code execution:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\aksha\Downloads\assistent_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistent_ai/ass7.py
10
5
○ PS C:\Users\aksha\Downloads\assistent_ai>
```

EXPLANATION:

The `self` parameter represents the current object instance. Without it, instance variables cannot be properly assigned or accessed.

### Task 5: Resolving Index Errors in Lists

CODE:

The code editor shows the following code:

```
18
19 numbers=[1,2,3]
20 print(numbers[5]) # This will raise an IndexError
```

A tooltip is visible over the line `print(numbers[5])`, showing options: < 1/2 > Accept Tab Accept Word Ctrl + RightArrow ...

```
8 #Index error handling
9 numbers=[1,2,3]
0 try:
1     print(numbers[5])
2 except IndexError:
3     print("Error: Index out of range.")
```

## OUTPUT:

```
▶ PS C:\Users\aksha\Downloads\assistent_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistent_ai/ass7.py
  Error: Index out of range.
▶ PS C:\Users\aksha\Downloads\assistent_ai> □
```

## EXPLANATION:

The AI suggested two safe approaches: checking the index against list length or handling the exception using try-except. Both prevent program crashes.