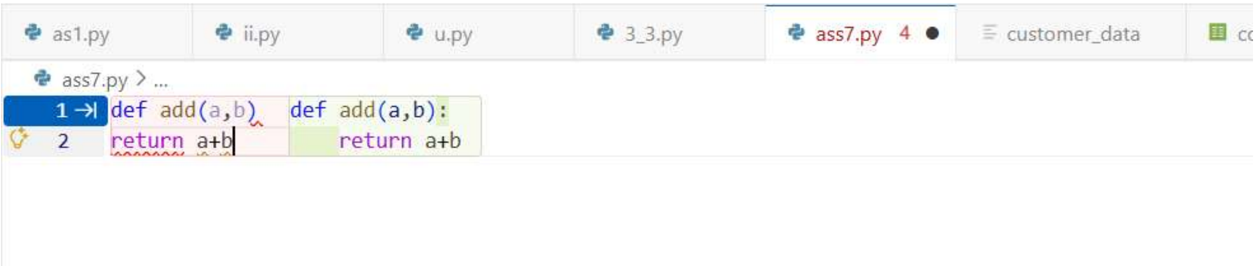
ASSIGNMENT – 7

Name: Abhinav

Hall.no: 2303A52174

Task 1: Fixing Syntax Errors

CODE:



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

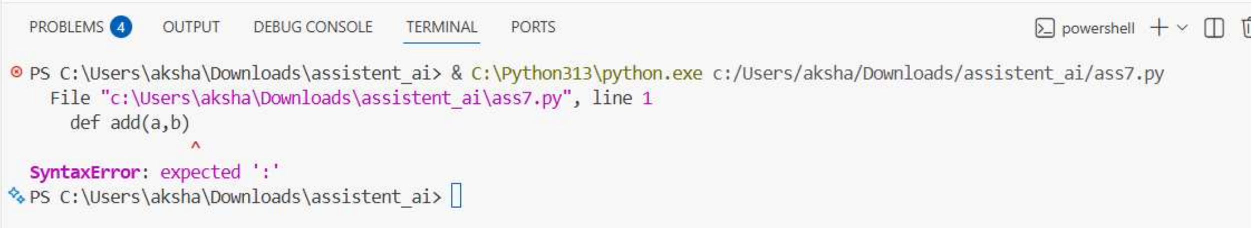
```
def add(a,b):
    return a+b
```

Error Identified by AI

Type of Error: SyntaxError

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

OUTPUT:



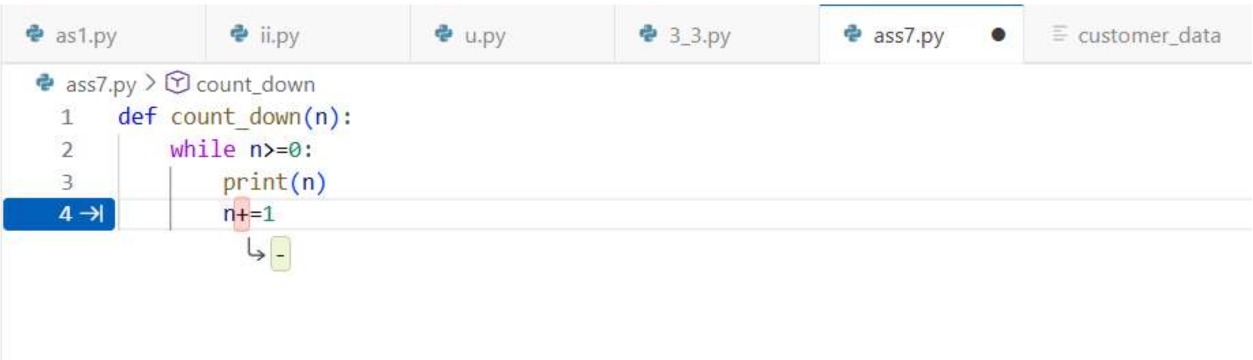
```
PS C:\Users\aksha\Downloads\assistent_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistent_ai/ass7.py
  File "c:\Users\aksha\Downloads\assistent_ai\ass7.py", line 1
    def add(a,b)
                ^
SyntaxError: expected ':'
PS C:\Users\aksha\Downloads\assistent_ai>
```

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.

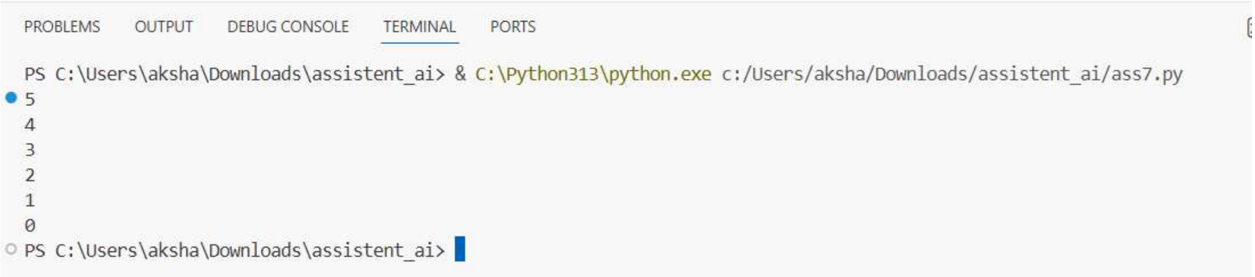
Task 2: Debugging Logic Errors in Loops

CODE:



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

OUTPUT:



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

EXPLANATION:

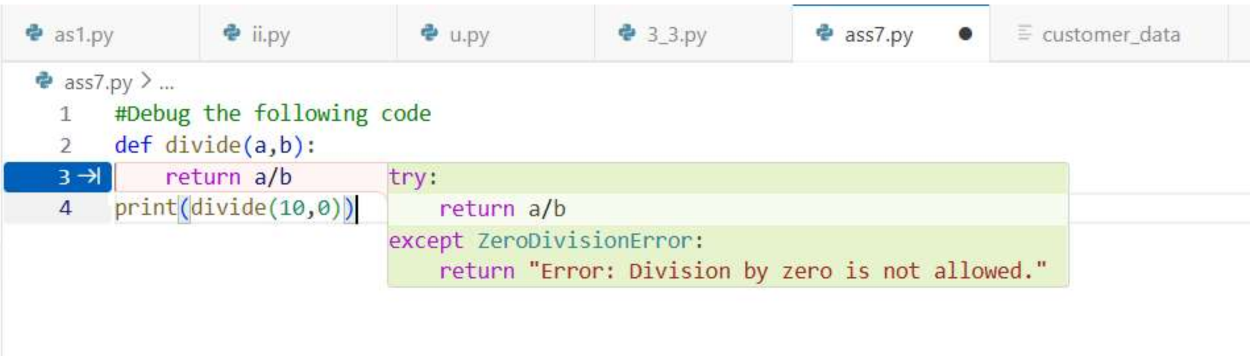
The loop condition depends on i increasing toward a stopping point. Decrementing i causes it to move away from the termination condition, resulting in an infinite loop.

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:



```
#Debug the following code
def divide(a,b):
    return a/b
print(divide(10,0))
```

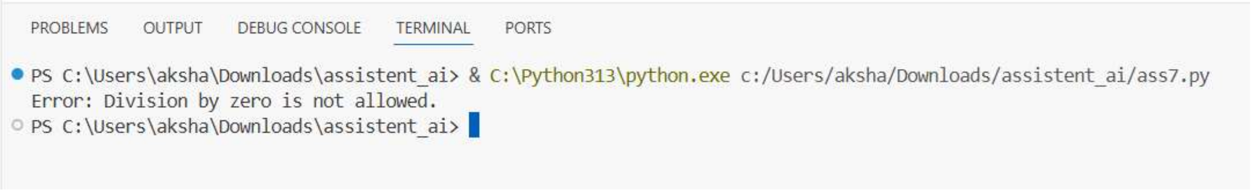
```
try:
    return a/b
except ZeroDivisionError:
    return "Error: Division by zero is not allowed."
```

Error Identified by AI

Type of Error: Runtime Error (ZeroDivisionError)

Reason: Division by zero is mathematically undefined

OUTPUT:



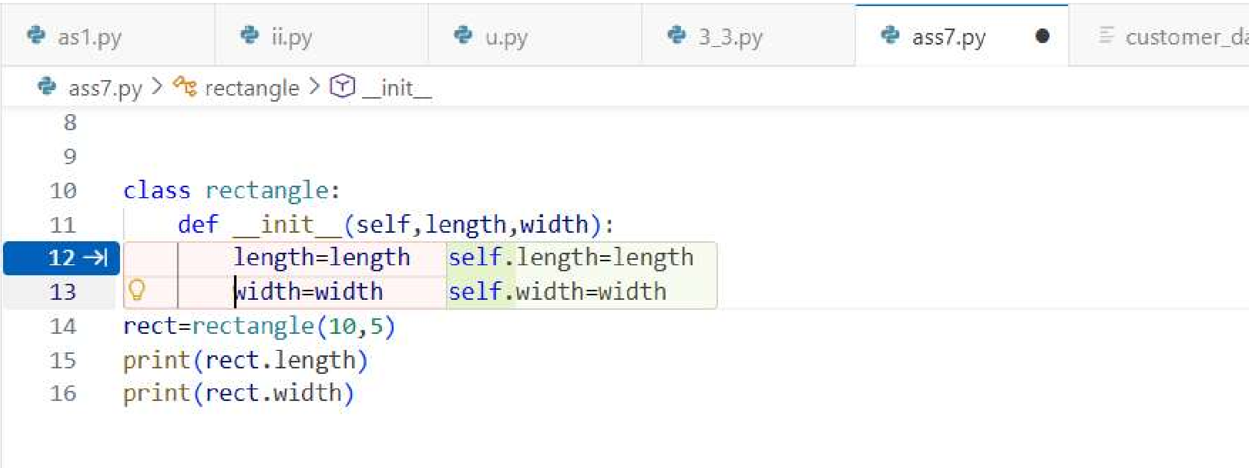
```
PS C:\Users\aksha\Downloads\assistent_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistent_ai/ass7.py
Error: Division by zero is not allowed.
PS C:\Users\aksha\Downloads\assistent_ai>
```

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.

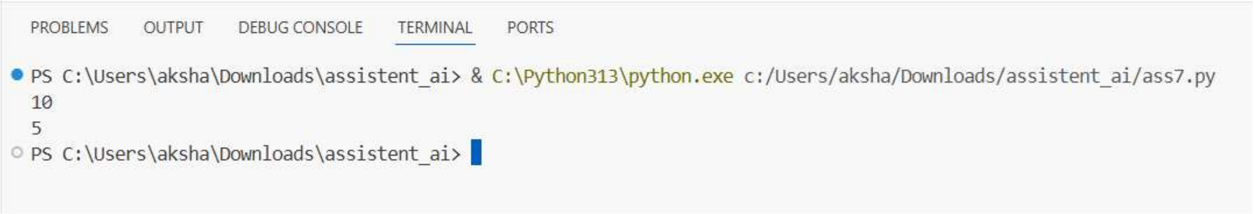
Task 4: Debugging Class Definition Errors

CODE:



```
class rectangle:
    def __init__(self,length,width):
        length=length    self.length=length
        width=width      self.width=width
rect=rectangle(10,5)
print(rect.length)
print(rect.width)
```

OUTPUT:



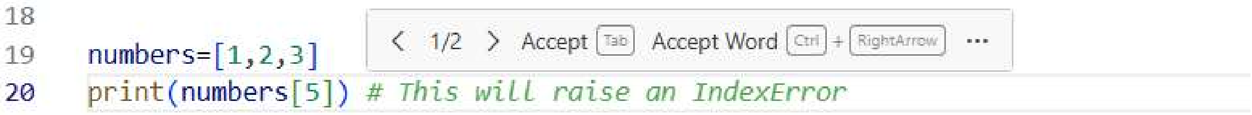
```
PS C:\Users\aksha\Downloads\assistent_ai> & C:\Python313\python.exe c:/Users/aksha/Downloads/assistent_ai/ass7.py
10
5
PS C:\Users\aksha\Downloads\assistent_ai>
```

EXPLANATION:

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

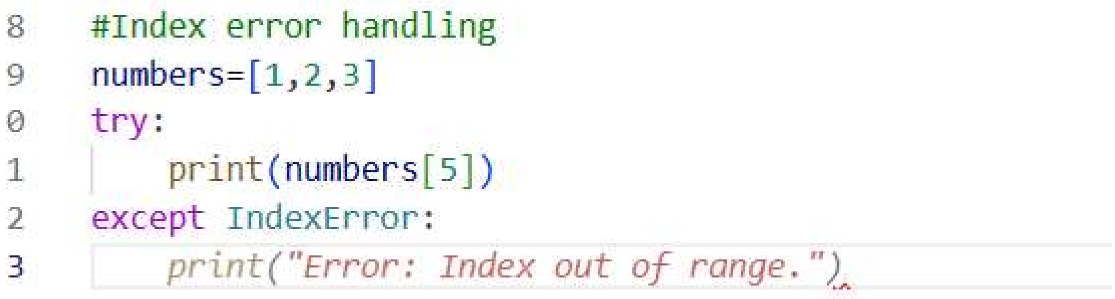
Task 5: Resolving Index Errors in Lists

CODE:



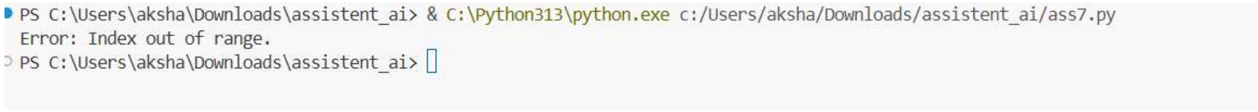
```

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



```python
#Index error handling
numbers=[1,2,3]
try:
    print(numbers[5])
except IndexError:
    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.