

ASSIGNMENT-7

NAME: Nallala siri

HT.NO: 2403A52037

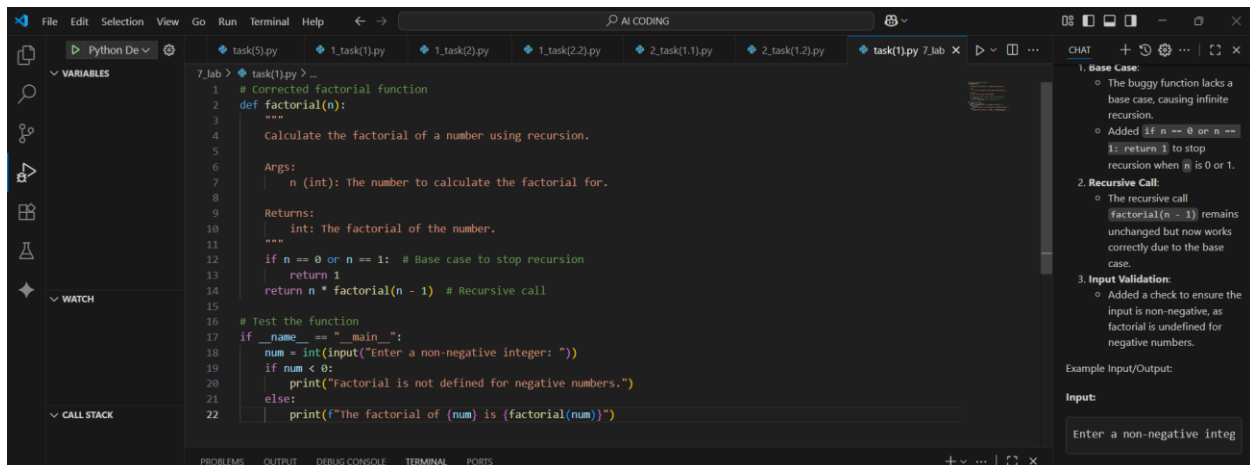
BATCH: AIB03

Task 1:

Introduce a buggy Python function that calculates the factorial of a number using recursion. Use Copilot or Cursor AI to detect and fix the logical or syntax errors.

Prompt: correct my error.

Code:



```
7_lab > task(1).py > ...
1 # corrected factorial function
2 def factorial(n):
3     """
4     Calculate the factorial of a number using recursion.
5
6     Args:
7         n (int): The number to calculate the factorial for.
8
9     Returns:
10        int: The factorial of the number.
11    """
12    if n == 0 or n == 1: # Base case to stop recursion
13        return 1
14    return n * factorial(n - 1) # Recursive call
15
16 # Test the function
17 if __name__ == "__main__":
18     num = int(input("Enter a non-negative integer: "))
19     if num < 0:
20         print("Factorial is not defined for negative numbers.")
21     else:
22         print(f"The factorial of {num} is {factorial(num)}")
```

1. Base Case:

- The buggy function lacks a base case, causing infinite recursion.
- Added `if n == 0 or n == 1: return 1` to stop recursion when `n` is 0 or 1.

2. Recursive Call:

- The recursive call `factorial(n - 1)` remains unchanged but now works correctly due to the base case.

3. Input Validation:

- Added a check to ensure the input is non-negative, as factorial is undefined for negative numbers.

Example Input/Output:

Input:

Enter a non-negative integer

OP:



```
PS C:\Users\tharu\OneDrive\Desktop\VAI CODING>
PS C:\Users\tharu\OneDrive\Desktop\VAI CODING> c:: cd 'c:\Users\tharu\OneDrive\Desktop\VAI CODING'; & 'c:\Program Files\Python11\python.exe' 'c:\Users\tharu\OneDrive\Desktop\VAI CODING\7_lab\task(1).py'
Enter a non-negative integer: 5
The factorial of 5 is 120
PS C:\Users\tharu\OneDrive\Desktop\VAI CODING>
```

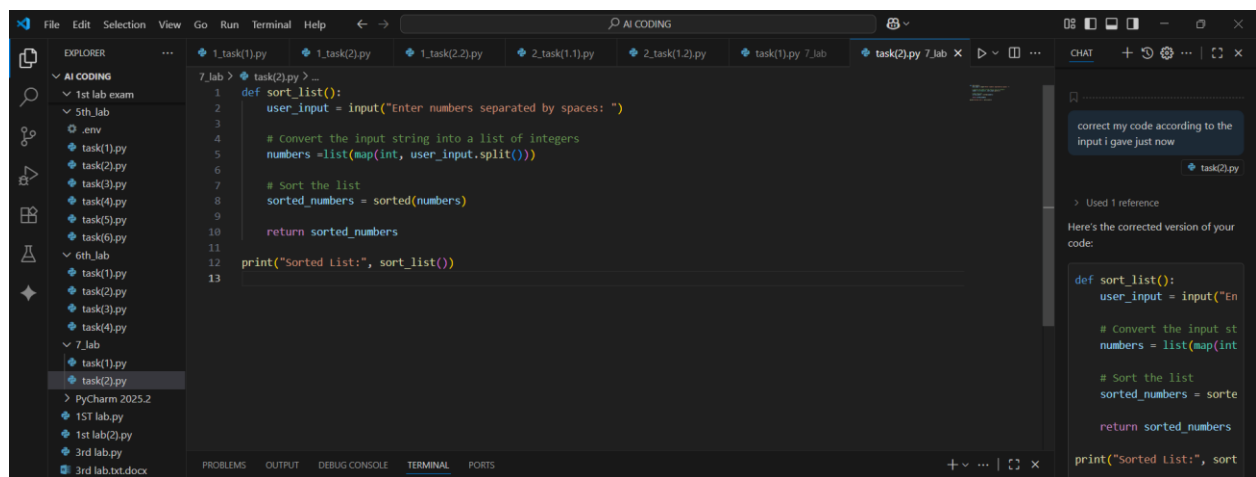
OBSERVATION: The buggy function lacks a base case, causing infinite recursion. Added `if n == 0 or n == 1: return 1` to stop recursion when `n` is 0 or 1. The recursive call `factorial(n - 1)` remains unchanged but now works correctly due to the base case. Added a check to ensure the input is non-negative, as factorial is undefined for negative numbers.

Task 2:

Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed integers and strings). Prompt AI to detect the issue and fix the code for consistent sorting

Prompt: correct my code according to the input i gave just now

Code:



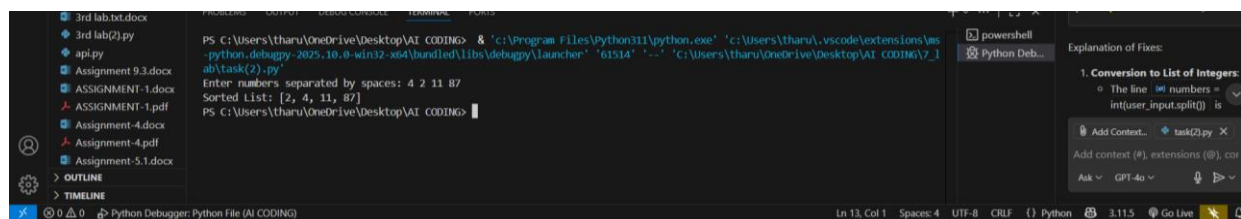
The screenshot shows the VS Code interface. The Explorer panel on the left shows a project structure with files like `task(1).py`, `task(2).py`, etc. The main editor shows a Python file `task(2).py` with the following code:

```
1 def sort_list():
2     user_input = input("Enter numbers separated by spaces: ")
3
4     # Convert the input string into a list of integers
5     numbers = list(map(int, user_input.split()))
6
7     # Sort the list
8     sorted_numbers = sorted(numbers)
9
10    return sorted_numbers
11
12 print("Sorted List:", sort_list())
13
```

The Chat panel on the right shows a conversation with an AI assistant. The user's prompt is "correct my code according to the input i gave just now". The assistant's response is:

```
def sort_list():
    user_input = input("En
    # Convert the input st
    numbers = list(map(int
    # Sort the list
    sorted_numbers = sorte
    return sorted_numbers
    print("Sorted List:", sort
```

OP:



The screenshot shows the VS Code interface with a terminal window open. The terminal output shows the execution of the sorting function:

```
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> & 'c:\Program Files\Python311\python.exe' 'c:\Users\tharu\.vscode\extensions\ms-python.debugpy-2025.10.0-winn32-x64\bundle\libs\debugpy\launcher' '61514' '--' 'c:\Users\tharu\OneDrive\Desktop\AI CODING\7_1'
Enter numbers separated by spaces: 4 2 11 87
Sorted List: [2, 4, 11, 87]
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

The Chat panel on the right shows a conversation with an AI assistant. The user's prompt is "correct my code according to the input i gave just now". The assistant's response is:

```
def sort_list():
    user_input = input("En
    # Convert the input st
    numbers = list(map(int
    # Sort the list
    sorted_numbers = sorte
    return sorted_numbers
    print("Sorted List:", sort
```

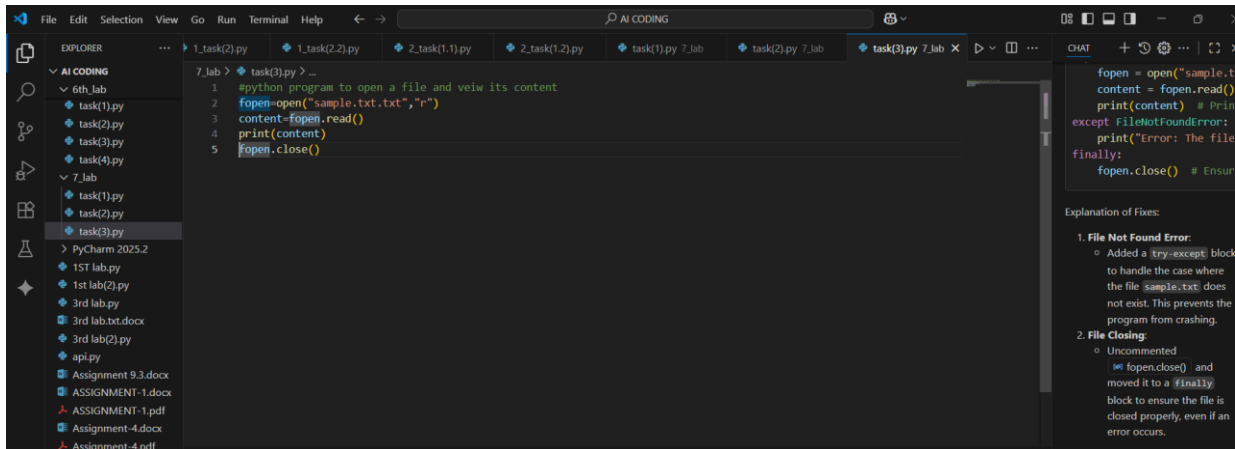
OBSERVATION: The line `number=int(user_input.split())` is incorrect because `split()` returns a list of strings, and `int()` cannot directly convert a list. Fixed by using `list(map(int, user_input.split()))`, which converts each string in the list to an integer. The `sorted()` function works correctly once the input is properly converted to a list of integers.

Task 3:

Write a Python snippet for file handling that opens a file but forgets to close it.
Ask Copilot or Cursor AI to improve it using the best practice (e.g., with open() block)

Prompt: correct my error

Code:



OP:



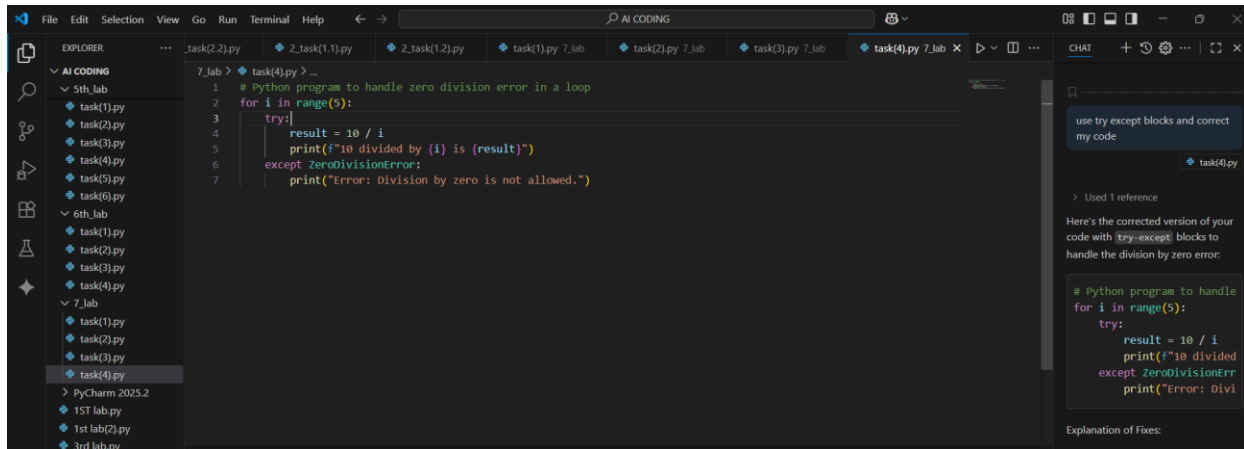
OBSERVATION: file sample.txt does not exist. This prevents the program from crashing. Uncommented `fopen.close()` and moved it to a finally block to ensure the file is closed properly, even if an error occurs.

Task 4:

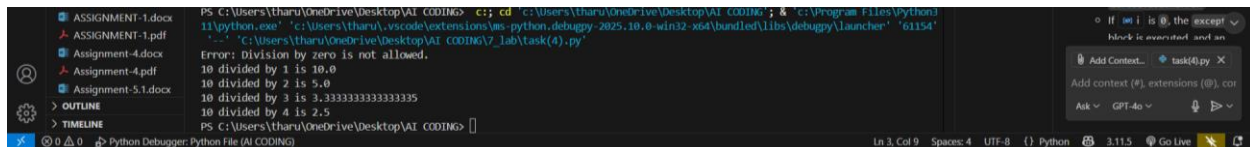
Provide a piece of code with a ZeroDivisionError inside a loop. Ask AI to add error handling using try-except and continue execution safely

Prompt: use try except blocks and correct my code

Code:



OP:



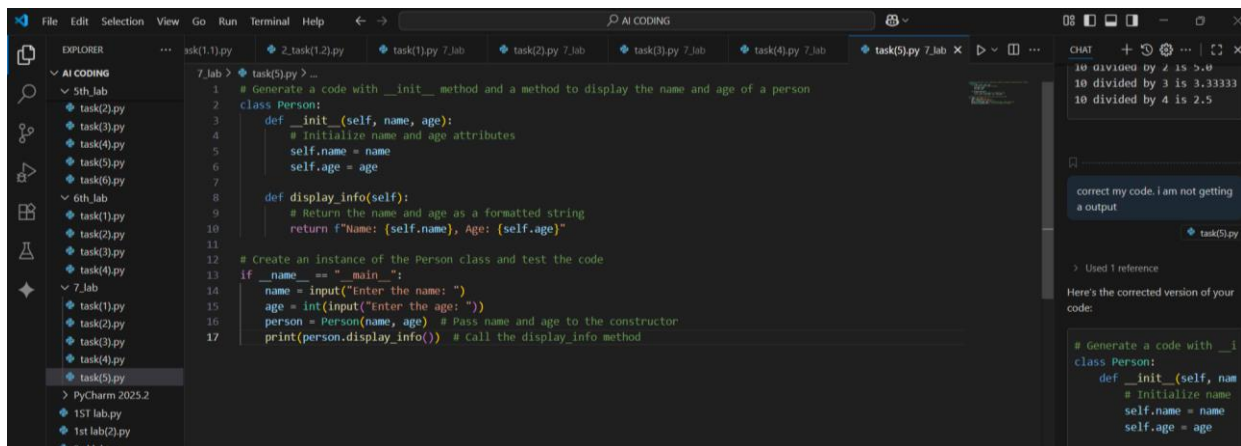
OBSERVATION: Wrapped the division operation `10 / i` in a try block to catch any `ZeroDivisionError`. If `i` is 0, the except block is executed, and an error message is printed. The program continues execution even if a division by zero occurs, thanks to the try-except block.

Task 5:

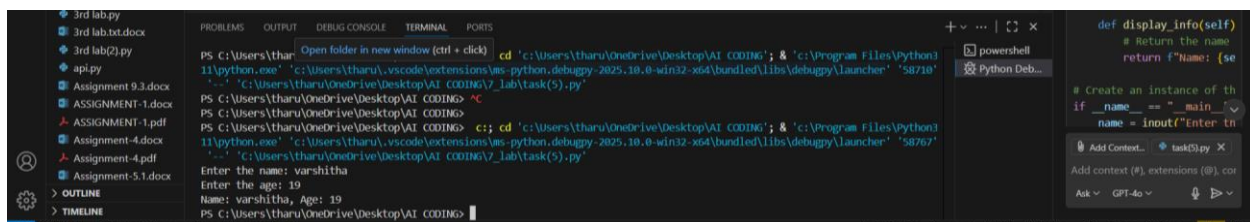
Include a buggy class definition with incorrect `__init__` parameters or attribute references. Ask AI to analyze and correct the constructor and attribute usage

Prompt: correct my code. i am not getting a output

Code:



OP:



OBSERVATION: Added name and age as parameters to the `__init__` method so that they can be passed when creating an instance of the person class. Added input prompts to allow the user to enter the name and age dynamically. Created an instance of the person class using the user-provided name and age. Called the `display_info` method to display the name and age.