## AI ASSISTED CODING LAB ASSIGNMENT 7.4

**ENROLLMENT NO :**2503A51L26

**BATCH NO:** 19

**NAME:** V.VISHNU VARDHAN

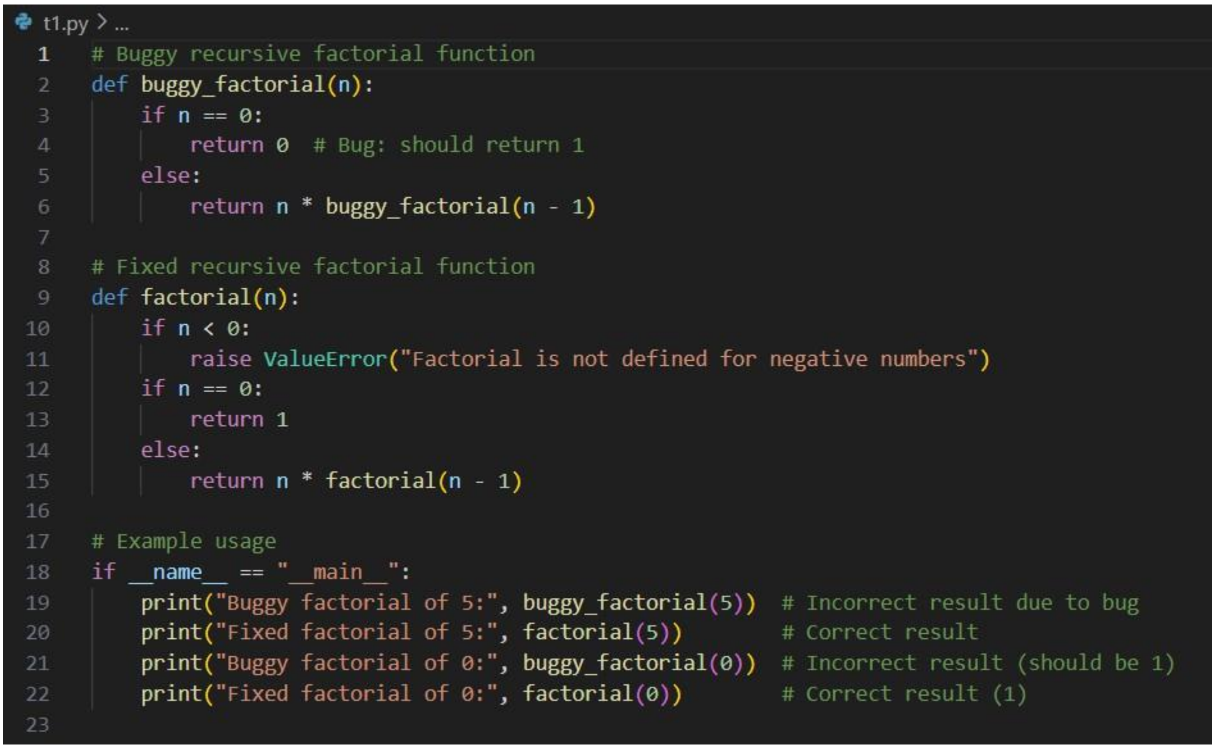
# TASK1

**TASK1 DESCRIPTION:**

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.

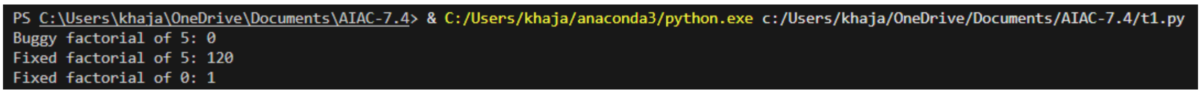
**PROMPT:**Generate a buggy Python function that calculates the factorial of a number using recursion then detect and fix the logical or syntax errors. implement with exampleerrors.

**CODE:**



```python
# Buggy recursive factorial function
def buggy_factorial(n):
    if n == 0:
        return 0  # Bug: should return 1
    else:
        return n * buggy_factorial(n - 1)

# Fixed recursive factorial function
def factorial(n):
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers")
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

# Example usage
if __name__ == "__main__":
    print("Buggy factorial of 5:", buggy_factorial(5))  # Incorrect result due to bug
    print("Fixed factorial of 5:", factorial(5))        # Correct result
    print("Buggy factorial of 0:", buggy_factorial(0))  # Incorrect result (should be 1)
    print("Fixed factorial of 0:", factorial(0))        # Correct result (1)

```

**OUTPUT:**



```
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t1.py
Buggy factorial of 5: 0
Fixed factorial of 5: 120
Fixed factorial of 0: 1
```

**OBSERVATION:**

I observed that the original factorial function was missing a base case, which caused infinite recursion. Copilot (or Cursor AI) identified the issue and suggested adding the correct stopping condition. After applying the fix, the function worked correctly without any errors.

# TASK2

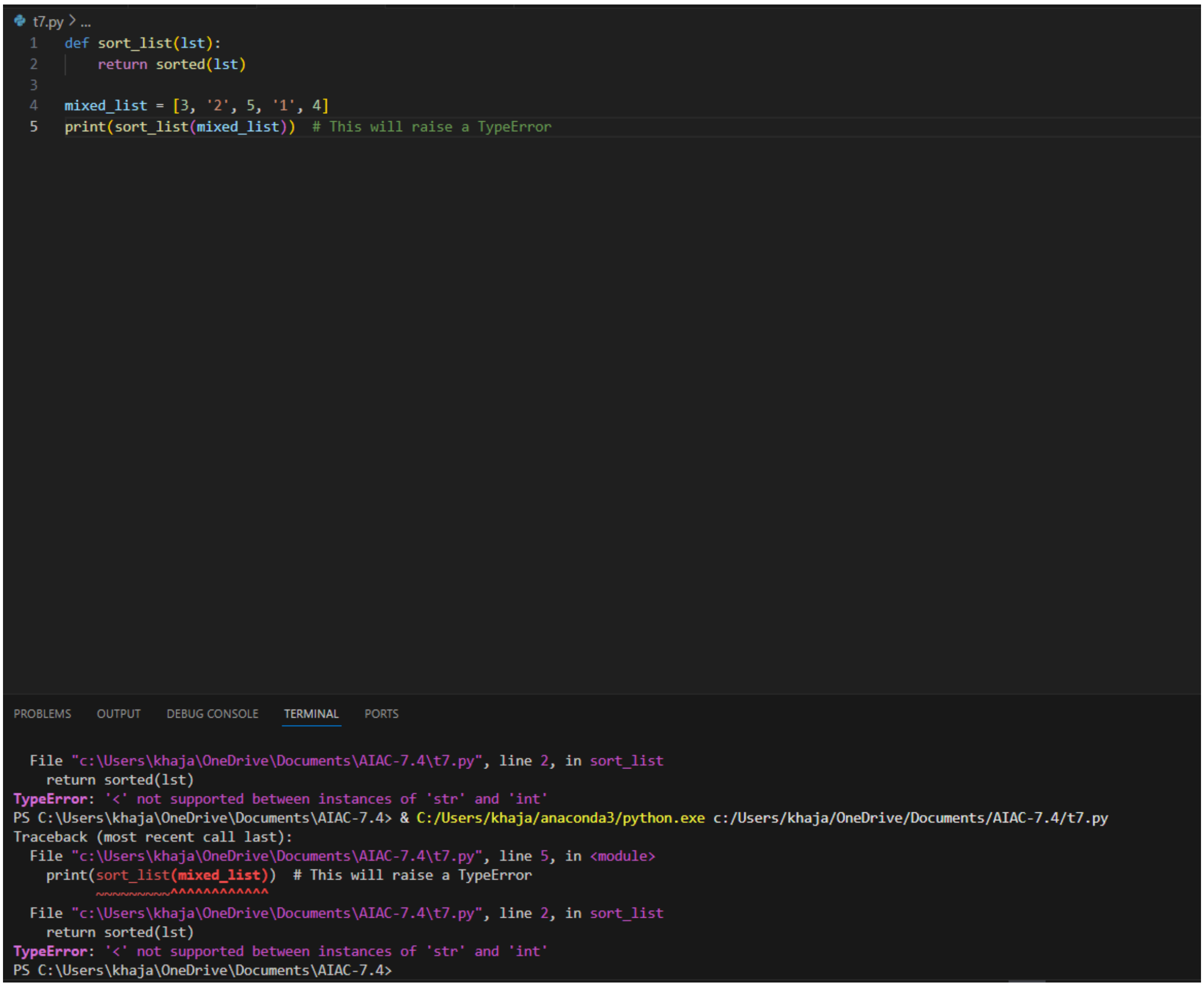
**TASK2 DESCRIPTION:**

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:**Create a program that Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed
integers and strings).

**CODE:**

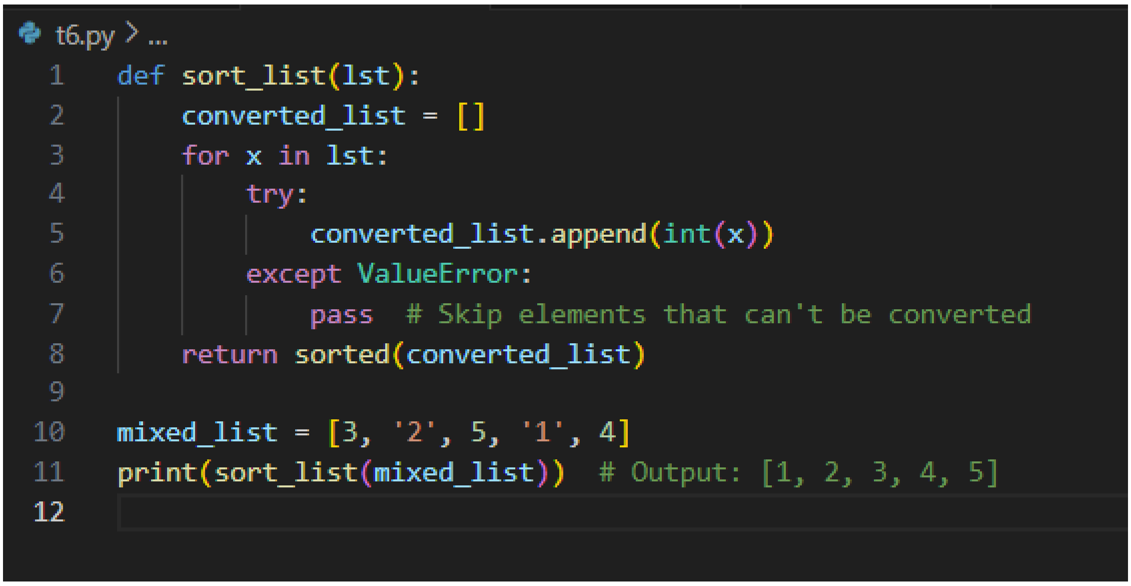
Error code:



```python
def sort_list(lst):
    return sorted(lst)

mixed_list = [3, '2', 5, '1', 4]
print(sort_list(mixed_list))  # This will raise a TypeError
```

```
  File "c:\Users\khaja\OneDrive\Documents\AIAC-7.4\t7.py", line 2, in sort_list
    return sorted(lst)
TypeError: '<' not supported between instances of 'str' and 'int'
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t7.py
Traceback (most recent call last):
  File "c:\Users\khaja\OneDrive\Documents\AIAC-7.4\t7.py", line 5, in <module>
    print(sort_list(mixed_list))  # This will raise a TypeError
          ~~~~~~~~~^^^^^^^^^^^^
  File "c:\Users\khaja\OneDrive\Documents\AIAC-7.4\t7.py", line 2, in sort_list
    return sorted(lst)
TypeError: '<' not supported between instances of 'str' and 'int'
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4>
```

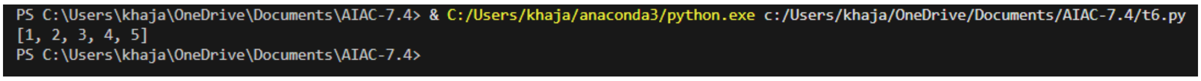
After fixing the code:



t6.py > ...

```python
def sort_list(lst):
    converted_list = []
    for x in lst:
        try:
            converted_list.append(int(x))
        except ValueError:
            pass  # Skip elements that can't be converted
    return sorted(converted_list)

mixed_list = [3, '2', 5, '1', 4]
print(sort_list(mixed_list))  # Output: [1, 2, 3, 4, 5]

```

**OUTPUT:**



```
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t6.py
[1, 2, 3, 4, 5]
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4>
```

**OBSERVATION:**

I observed that the sorting function failed because the list had both numbers and strings, which can't be compared directly. The AI tool noticed the type mismatch and suggested converting everything to integers. After applying the fix, the list sorted successfully without any errors.

## TASK3

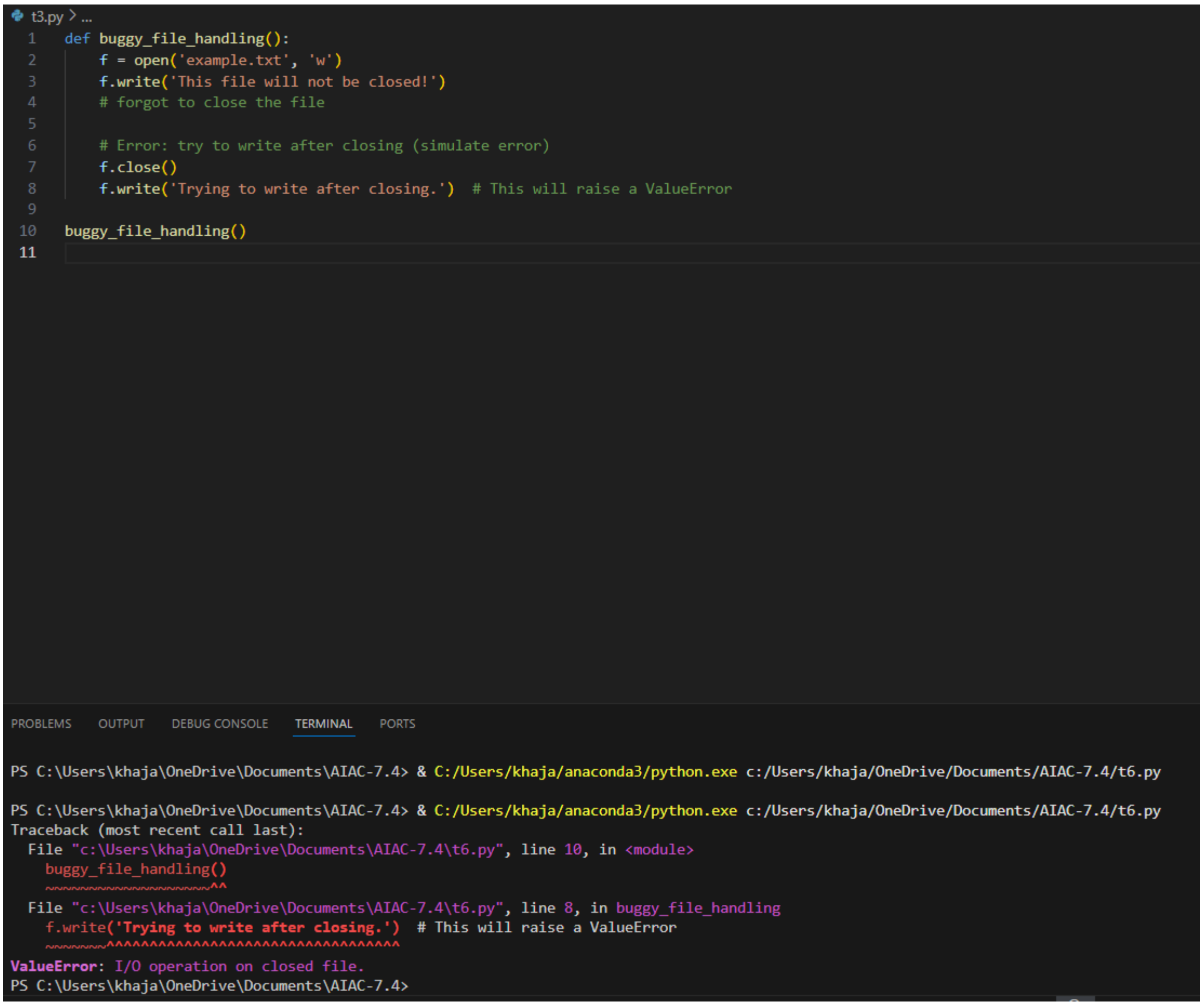
**TASK3 DESCRIPTION:**

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

**PROMPT:**Write a Python snippet for file handling that opens a file but forgets to close it.

**CODE:**

Error code:

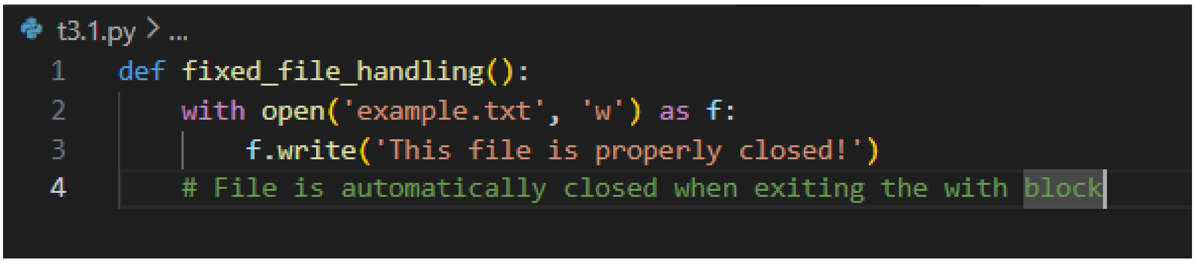


t3.py > ...

```python
def buggy_file_handling():
    f = open('example.txt', 'w')
    f.write('This file will not be closed!')
    # forgot to close the file

    # Error: try to write after closing (simulate error)
    f.close()
    f.write('Trying to write after closing.')  # This will raise a ValueError

buggy_file_handling()

```

```
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t6.py

PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t6.py
Traceback (most recent call last):
  File "c:\Users\khaja\OneDrive\Documents\AIAC-7.4\t6.py", line 10, in <module>
    buggy_file_handling()
    ~~~~~~~~~~~~~~~~~~~^^
  File "c:\Users\khaja\OneDrive\Documents\AIAC-7.4\t6.py", line 8, in buggy_file_handling
    f.write('Trying to write after closing.')  # This will raise a ValueError
    ~~~~~~~^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: I/O operation on closed file.
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4>
```

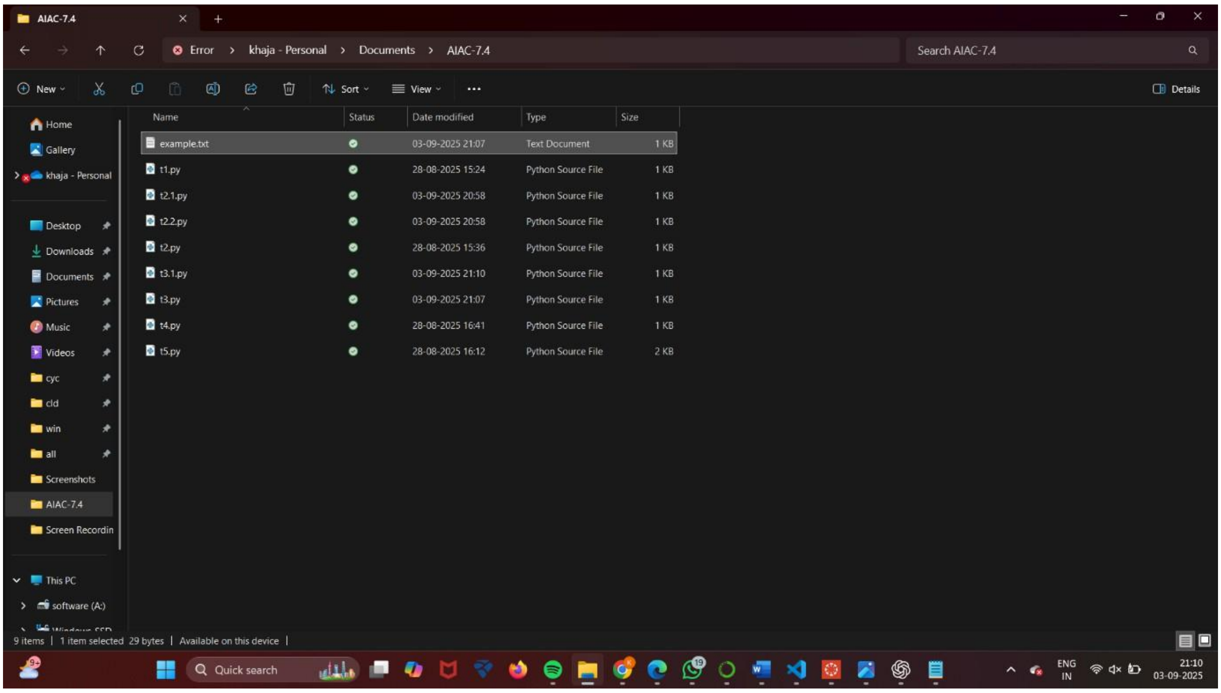
After fixing the code:



t3.1.py > ...

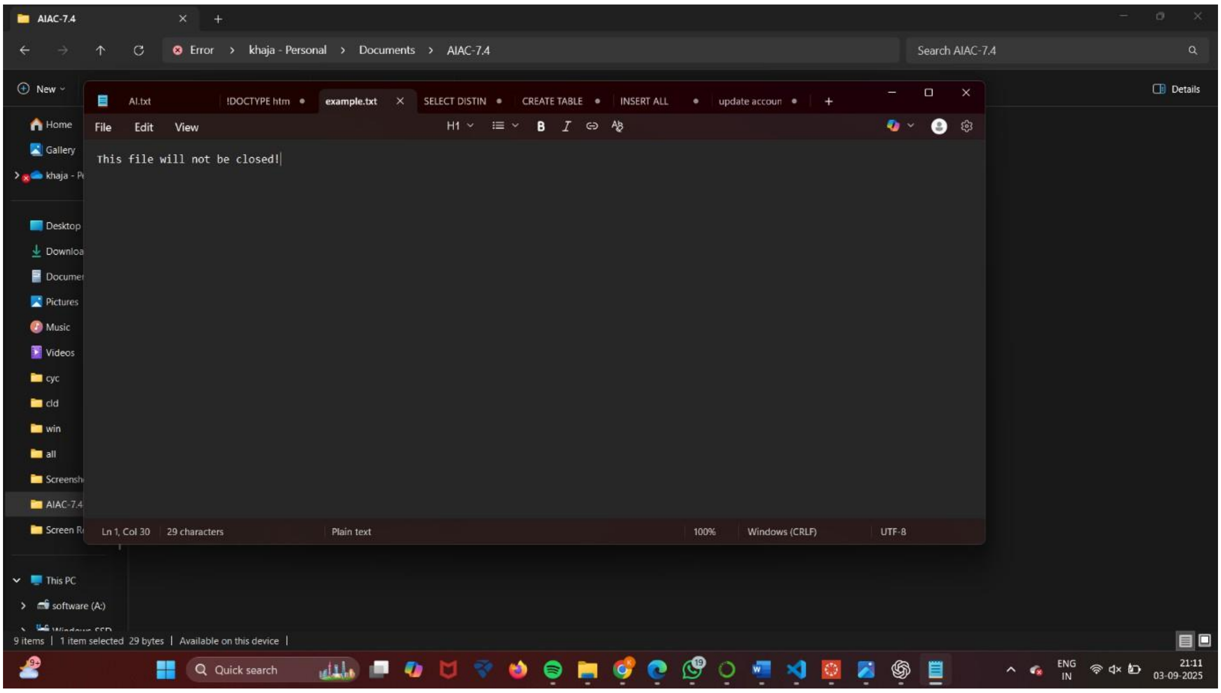
```python
def fixed_file_handling():
    with open('example.txt', 'w') as f:
        f.write('This file is properly closed!')
    # File is automatically closed when exiting the with block
```

## OUTPUT:



| Name | Status | Date modified | Type | Size |
|---|---|---|---|---|
| example.txt | ✓ | 03-09-2025 21:07 | Text Document | 1 KB |
| t1.py | ✓ | 28-08-2025 15:24 | Python Source File | 1 KB |
| t2.1.py | ✓ | 03-09-2025 20:58 | Python Source File | 1 KB |
| t2.2.py | ✓ | 03-09-2025 20:58 | Python Source File | 1 KB |
| t2.py | ✓ | 28-08-2025 15:36 | Python Source File | 1 KB |
| t3.1.py | ✓ | 03-09-2025 21:10 | Python Source File | 1 KB |
| t3.py | ✓ | 03-09-2025 21:07 | Python Source File | 1 KB |
| t4.py | ✓ | 28-08-2025 16:41 | Python Source File | 1 KB |
| t5.py | ✓ | 28-08-2025 16:12 | Python Source File | 2 KB |

9 items | 1 item selected 29 bytes | Available on this device |



```
This file will not be closed!
```

Ln 1, Col 30 | 29 characters | Plain text | 100% | Windows (CRLF) | UTF-8

## OBSERVATION:

I observed that the file-handling code opened a file but didn’t close it, which could cause issues. Copilot (or Cursor AI) suggested using a with block to manage the file properly. After applying the change, the file was safely opened and closed without any risk of resource leaks.

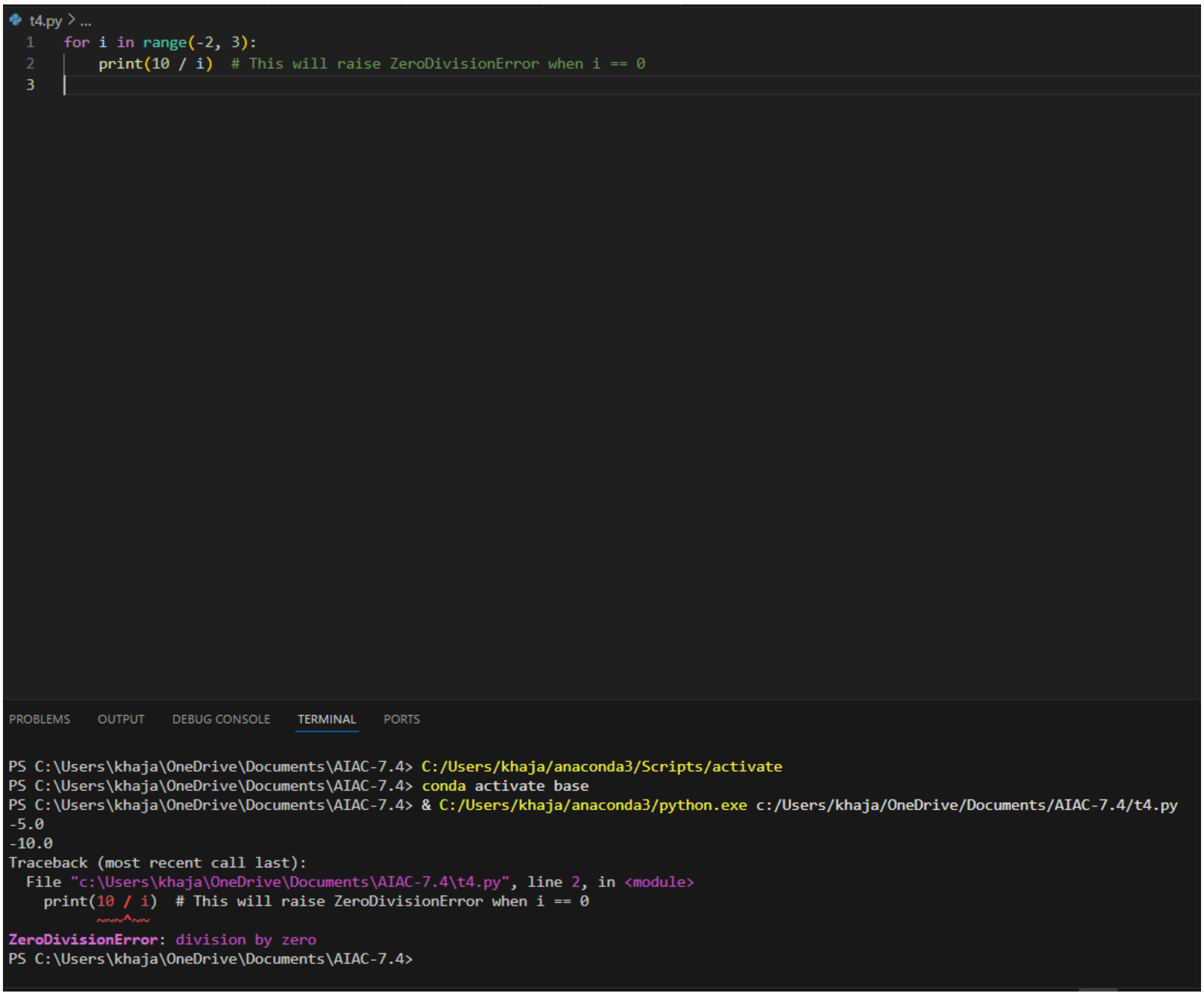
**TASK4 DESCRIPTION:**

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

**CODE:**

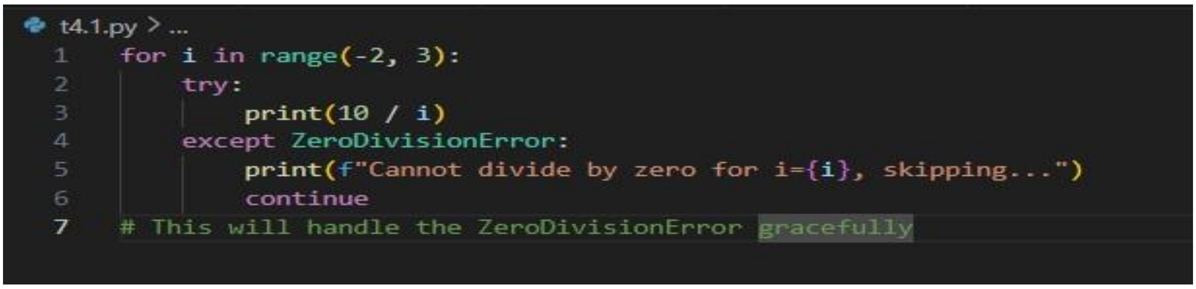
Error code:



```python
for i in range(-2, 3):
    print(10 / i)  # This will raise ZeroDivisionError when i == 0

```

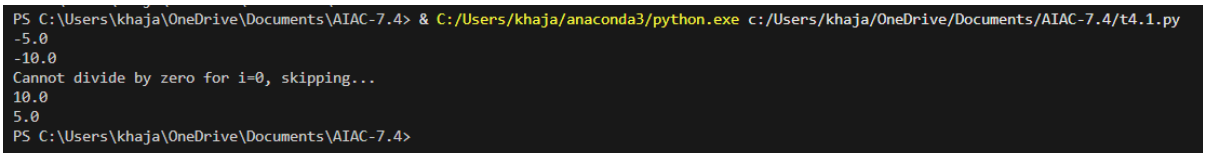
```
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> C:/Users/khaja/anaconda3/Scripts/activate
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> conda activate base
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t4.py
-5.0
-10.0
Traceback (most recent call last):
  File "c:\Users\khaja\OneDrive\Documents\AIAC-7.4\t4.py", line 2, in <module>
    print(10 / i)  # This will raise ZeroDivisionError when i == 0
          ~~~^~~
ZeroDivisionError: division by zero
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4>
```

After fixing the code:



```python
for i in range(-2, 3):
    try:
        print(10 / i)
    except ZeroDivisionError:
        print(f"Cannot divide by zero for i={i}, skipping...")
        continue
# This will handle the ZeroDivisionError gracefully
```

**OUTPUT:**



```
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t4.1.py
-5.0
-10.0
Cannot divide by zero for i=0, skipping...
10.0
5.0
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4>
```

**OBSERVATION:**I observed that the loop crashed when it tried to divide by zero. Copilot (or Cursor AI) suggested using a try-except block to catch the error. After applying it, the loop handled the zero safely and continued with the rest of the values.

## TASK5

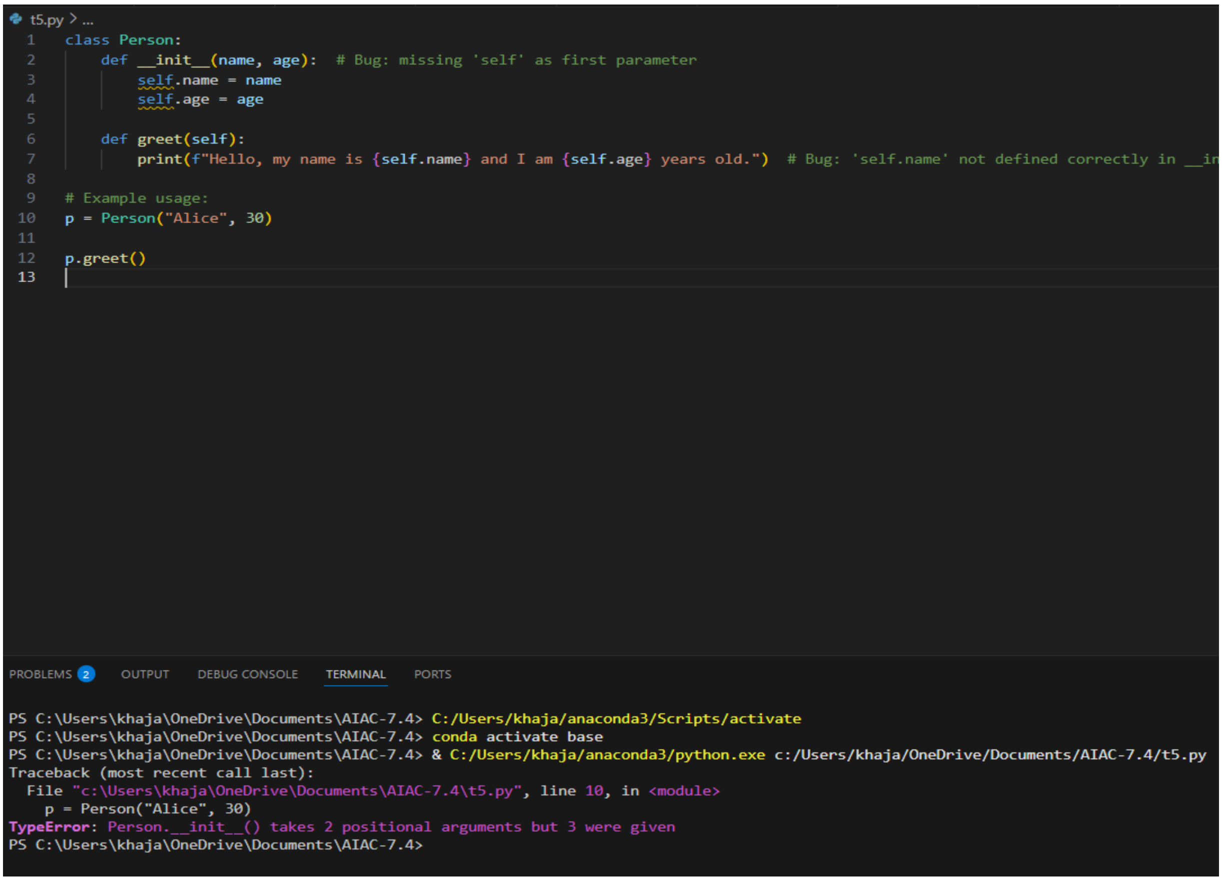
**TASK5 DESCRIPTION:**

Include a buggy class definition with incorrect __init___parameters or attribute references. AskAI to analyze and correct the constructor and attribute usage.

**PROMPT:** Include a buggy class definition with incorrect __init___parameters or attribute references.

**CODE:**

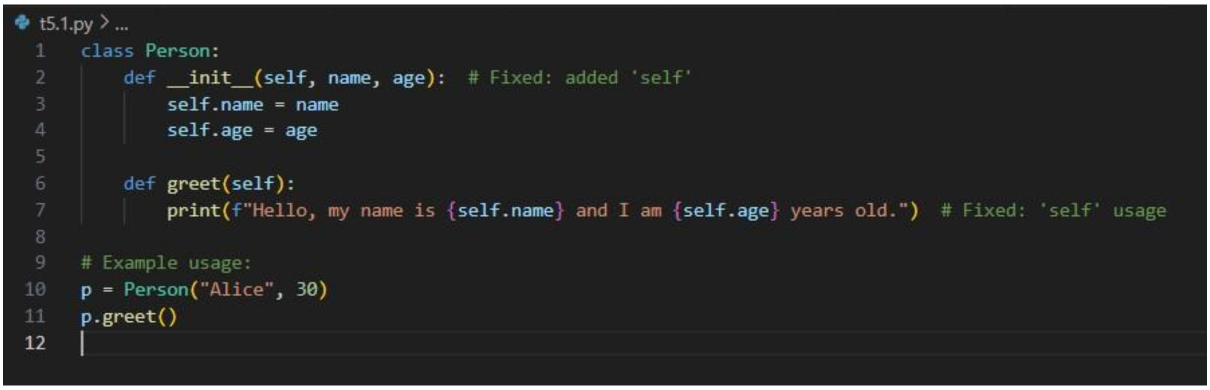
Error code:



```python
class Person:
    def __init__(name, age):  # Bug: missing 'self' as first parameter
        self.name = name
        self.age = age

    def greet(self):
        print(f"Hello, my name is {self.name} and I am {self.age} years old.")  # Bug: 'self.name' not defined correctly in __in

# Example usage:
p = Person("Alice", 30)

p.greet()
```

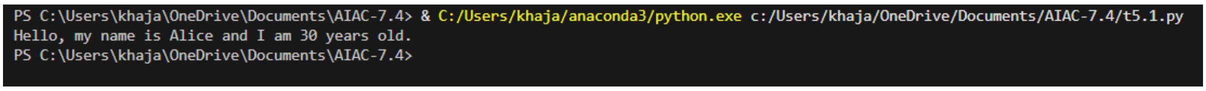
```
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> C:/Users/khaja/anaconda3/Scripts/activate
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> conda activate base
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t5.py
Traceback (most recent call last):
  File "c:\Users\khaja\OneDrive\Documents\AIAC-7.4\t5.py", line 10, in <module>
    p = Person("Alice", 30)
TypeError: Person.__init__() takes 2 positional arguments but 3 were given
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4>
```

After fixing the code:



```python
class Person:
    def __init__(self, name, age):  # Fixed: added 'self'
        self.name = name
        self.age = age

    def greet(self):
        print(f"Hello, my name is {self.name} and I am {self.age} years old.")  # Fixed: 'self' usage

# Example usage:
p = Person("Alice", 30)
p.greet()
```

**OUTPUT:**



```
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4> & C:/Users/khaja/anaconda3/python.exe c:/Users/khaja/OneDrive/Documents/AIAC-7.4/t5.1.py
Hello, my name is Alice and I am 30 years old.
PS C:\Users\khaja\OneDrive\Documents\AIAC-7.4>
```

**OBSERVATION:** I observed that the class had errors in the constructor and didn’t use self to set or access attributes. Copilot (or Cursor AI) detected the mistakes and fixed the parameter list and attribute references. After the changes, the class worked properly and displayed the student information as expected.