

Assignment 7.1

M.Sai Teja

2303A510A3

Task 1: Debugging a Recursive Calculation Module

```
C: > Users > malya > Ai > Untitled-1.py > factorial
1  def factorial(n)
2  |  return n * factorial(n-1)
```

Output:

```
PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-1.py
SyntaxError: expected ':'
PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-1.py
File "c:\Users\malya\Ai\Untitled-1.py", line 1
    def factorial(n)
    ^
SyntaxError: expected ':'
PS C:\Users\malya>
```

After Debugging

I found the syntax error! You're missing a **colon (:)** at the end of the function definition line.

In Python, all function definitions must end with a colon to indicate the start of the function body. Here's the corrected code:

```
def factorial(n):
    return n * factorial(n-1)
```

Key points:

- The **colon (:)** signals to Python that an indented block follows
- The indentation of the [return](#) statement shows it's part of the function body

Additional observation: This recursive function has an issue—it will cause infinite recursion because it never has a **base case** (a condition to stop recursing). You'd want to add a base case:

```
def factorial(n):
```

```

if n <= 1:
    return 1

return n * factorial(n-1)

```

This prevents the recursion from continuing indefinitely by returning 1 when n reaches 1 or less.

```

C: > Users > malya > Ai > Untitled-1.py > factorial
1  def factorial(n):
2      if n <= 1:
3          return 1
4      return n * factorial(n-1)
5  number = int(input("Enter a number: "))
6  result = factorial(number)
7  print(f"Factorial of {number} is {result}")

```

Output:

```

PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-1.py
Enter a number: 7
Factorial of 7 is 5040
PS C:\Users\malya>

```

Task – 2:

Task 2: Fixing Data Type Errors in a Sorting Utility

```

C: > Users > malya > Ai > Untitled-2.py > ...
1  data = [10, "5", 3, "20", 1]
2  print(sorted(data))
3

```

Output:

```

PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-2.py
Traceback (most recent call last):
  File "c:\Users\malya\Ai\Untitled-2.py", line 2, in <module>
    print(sorted(data))
          ~~~~~~^~~~~~
TypeError: '<' not supported between instances of 'str' and 'int'
PS C:\Users\malya>

```

After Debugging:

The code has a **bug**. Here's what's happening:

```
data = [10, "5", 3, "20", 1]
```

```
print(sorted(data))
```

The Problem:

You're trying to sort a list that contains **both integers and strings**. In Python 3, you cannot compare integers with strings directly, so `sorted()` fails with a `TypeError`.

Why it fails:

- The list has: 10 (int), "5" (string), 3 (int), "20" (string), 1 (int)
- When sorting, Python tries to compare 10 with "5", but integers and strings aren't comparable

```
C: > Users > malya > Ai > Untitled-2.py > ...
1  data = [10, "5", 3, "20", 1]
2  # Convert all elements to integers before sorting
3  data = [int(x) for x in data]
4  print(sorted(data))
5  |
```

Output:

```
PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-2.py
[1, 3, 5, 10, 20]
PS C:\Users\malya> |
```

Task - 3:

Task 3: Improving File Handling Reliability

```
C: > Users > malya > Ai > Untitled-3.py > ...
1  file = open("data.txt", "r")
2  content = file.read()
3  print(content)
4  |
```

Output:

```
PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-3.py
Traceback (most recent call last):
  File "c:\Users\malya\Ai\Untitled-3.py", line 1, in <module>
    file = open("data.txt", "r")
FileNotFoundError: [Errno 2] No such file or directory: 'data.txt'
```

After Debugging:

What it does:

1. `open("data.txt", "r")` - Opens a file named data.txt in **read mode** ("r")
2. `file.read()` - Reads all the content from the file
3. `print(content)` - Displays the file content

Why

it's

failing:

The file data.txt doesn't exist in your working directory, so Python throws a `FileNotFoundError`.

```

# Create sample data file if it doesn't exist
try:
    with open("data.txt", "w") as file:
        file.write("Hello, World!\nThis is sample data.\nFile handling in Python!")
except Exception as e:
    print(f"Error creating file: {e}")

# Read and display the file
try:
    with open("data.txt", "r") as file:
        content = file.read()
        print("File Content:")
        print(content)
except FileNotFoundError:
    print("Error: data.txt file not found!")

```

Output:

```

PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-3.py
File Content:
Hello, World!
This is sample data.
File handling in Python!
PS C:\Users\malya>

```

Task – 4:

Task 4: Handling Runtime Errors Gracefully in Loops

```

C: > Users > malya > Ai > Untitled-14py.py > ...
1 values = [10, 5, 0, 2]
2 for v in values:
3     print(10 / v)
4

```

Output :

```

PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-14py.py
Traceback (most recent call last):
  File "c:/Users/malya/Ai/Untitled-14py.py", line 4, in <module>
    print(10 / v)
          ~~~~
ZeroDivisionError: division by zero

```

After Debugging:

What it does:

- Loops through each value in the list: 10, 5, 0, 2
- Divides 10 by each value and prints the result

Why

it's

failing:

When v = 0, Python tries to execute 10 / 0, which causes a **ZeroDivisionError** (you cannot divide by zero).

```

C: > Users > malya > Ai > Untitled-14py.py > ...
1  values = [10, 5, 0, 2]
2  for v in values:
3      try:
4          result = 10 / v
5          print(result)
6      except ZeroDivisionError:
7          print(f"Error: Cannot divide by {v}")
8

```

Output:

```

PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-14py.py
1.0
2.0
Error: Cannot divide by 0
5.0
PS C:\Users\malya>

```

Task -5:

Task 5: Debugging Class Initialization Errors

```

C: > Users > malya > Ai > Untitled-5.py > ...
1  class Student:
2      def __init__(name, age):
3          self.name = name
4          self.age = age
5
6      def display():
7          print(name, age)
8
9  s = Student("Teja", 20)
10 s.display()
11

```

Output :

```

PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-5.py
Traceback (most recent call last):
  File "c:\Users\malya\Ai\Untitled-5.py", line 9, in <module>
    s = Student("Teja", 20)
TypeError: Student.__init__() takes 2 positional arguments but 3 were given
PS C:\Users\malya>

```

After debugging:

What it's trying to do:

- Create a [Student](#) class with name and age attributes
- Initialize a student object and display their details

Why it's failing:

1. `__init__` **method**: Missing `self` parameter
 - Should be: `def __init__(self, name, age):`
2. `display()` **method**: Missing `self` parameter AND references to variables
 - Should be: `def display(self):`
 - Should reference: `self.name` and `self.age`

```
C: > Users > malya > Ai > Untitled-5.py > Student
1  class Student:
2      def __init__(self, name, age):
3          self.name = name
4          self.age = age
5
6      def display(self):
7          print(f"{self.name}, {self.age}")
8
9  s = Student("Teja", 20)
10 s.display()
11
```

Output:

```
PS C:\Users\malya> & C:/Users/malya/AppData/Local/Programs/Python/Python313/python.exe c:/Users/malya/Ai/Untitled-5.py
Teja, 20
```