Cyber Security Bootcamp

Hyperiondev

Defensive Programming

Welcome

Your Lecturer for this session

Liano Naidoo

Objectives

- Discover the different types of errors that could occur in your programs and how to handle them.

Everyone Makes Mistakes

- ★ No programmer is perfect, and we're going to make a lot of mistakes in our journey – and that is perfectly okay!
- ★ What separates the good programmers from the average ones is the ability to find and debug errors they encounter.

Error Messages

- ★ The output window of your IDE will usually show any and all Error messages if an error or mistake is detected.
- ★ It should display the type of error found as well as the line number in your code where the error occurred.
- ★ Your program will stop running immediately when an error is found.

Error Message Example

```
Traceback (most recent call last):
  File "c:\Work\defender.py", line 2, in <module>
    num1 = int(input("Enter a number: "))
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'o'
```

- ★ Looking at the above example:
  - The message states that the error occurred around line 9 – a good starting point for debugging.
  - It also states the type of error, which appears to be a TypeError. Useful, since we already could have ideas on how to fix the error.

Syntax Errors

- ★ Some of the easiest errors to fix ...
  - ... Usually
- ★ Mainly caused by typos in code or Python specific keywords that were misspelled or rules that were not followed.
- ★ When incorrect syntax is detected, Python will stop running and display an error message.

Indentation Errors

- ★ Indentation is important in programming.
- ★ Python uses indentation to understand where blocks of code start and stop.
- ★ The presence of indentation errors means that there is something wrong with the structure of the code.
- ★ A good rule of thumb: if a line of code ends with a colon (:), the next line should be indented.

Type Errors

- A type error occurs when your code has misinterpreted one type of data for another, like integers for strings.

- Remember that for Python to actually work, your code needs to make logical sense so that Python can interpret it correctly and achieve the desired output.

Name Errors

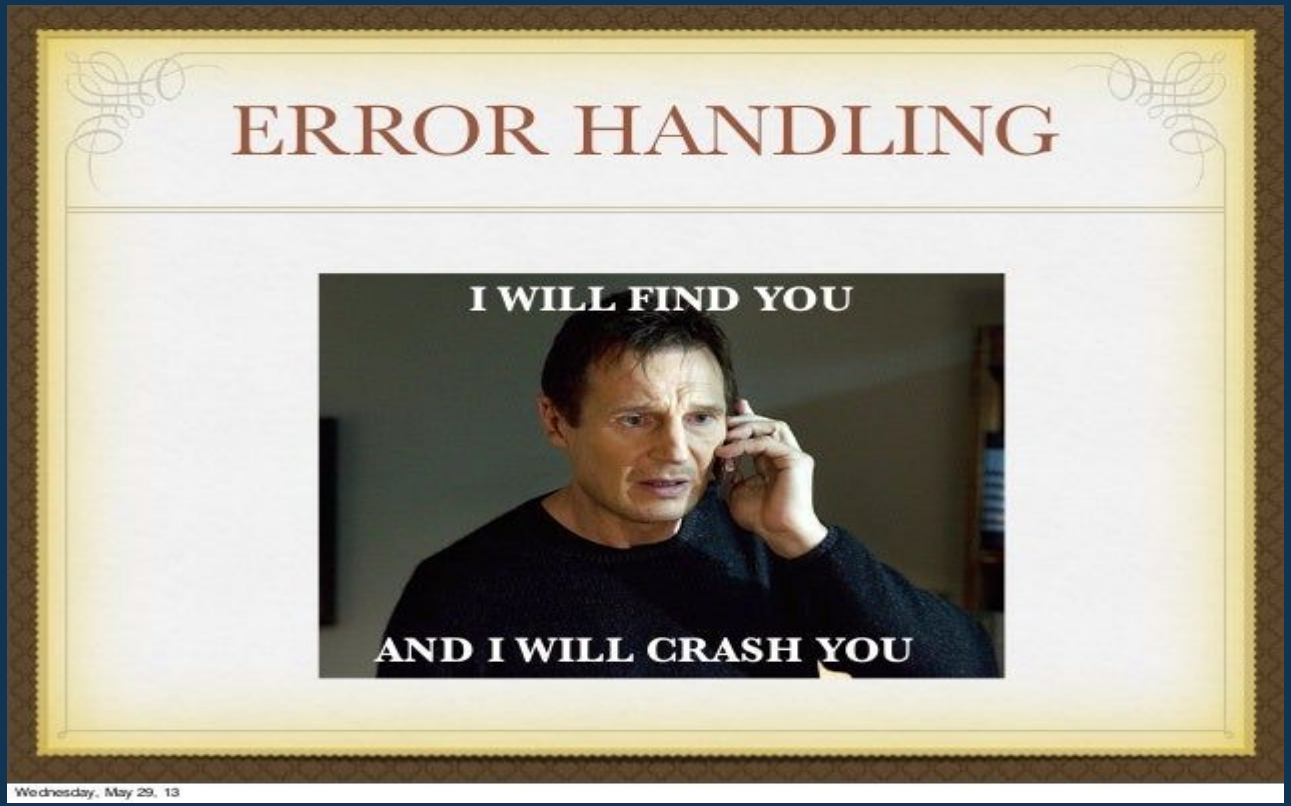
- ★ Naming errors occur when you try to reference or call a variable that has not been declared / created yet.
- ★ A good habit to get into when coding is to first define all variables, functions, etc. at the top of your program.

Logical Errors

- ★ Logical errors occur when your program is running, but the output you are receiving is not what you are expecting.

- ★ The code could be typed incorrectly, or perhaps an important line has been omitted, or the instructions given to the program have been coded in the wrong order.

Exception Handling



Exception Handling

- ★ Exception handling is a way to handle and manage errors that occur during the execution of a program

- ★ By using exception handling, we can catch and handle errors in a more graceful and controlled manner as opposed to allowing the program to crash as per usual.

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic, should you have any.

Hyperiondev

Thank You for Joining Us