Modular-mlproject-1

Exception handling

**Line 1:** import sys

* Imagine you have a toolbox filled with helpful tools. This line is like grabbing the "sys" toolbox, which contains useful tools for programmers.

**Line 2:** def error\_message\_detail(error, error\_details=sys.exc\_info()):

* This line defines a special helper named error\_message\_detail. Think of it as a detective you can call whenever there's an error in your program.
* It takes two pieces of information (like clues for the detective):
  + error: This is the actual problem that happened in your program.
  + error\_details (optional): By default, it uses the "sys" toolbox (from line 1) to gather more clues about the error. You can provide this information yourself if needed, but usually, the detective (the function) can find it on their own.

**Line 3:** \_, \_, exc\_tb = error\_details

* The detective (the function) uses the "sys" toolbox to gather clues, but right now, it only needs one specific clue: the exact location of the crime scene (the error).
* This line unpacks those clues, but throws away the first two (like discarding irrelevant details) and keeps only the third one, which is stored in exc\_tb. This third clue is like a fingerprint pointing to the location.

**Lines 4-5:** file\_name = exc\_tb.tb\_frame.f\_code.co\_filenameline\_number = exc\_tb.tb\_lineno

* Now that the detective has the fingerprint (the exc\_tb clue), they use it to find the exact spot in your program where the error happened.
* Line 4 extracts the name of the script (file) where the error occurred, similar to finding the room where the crime took place.
* Line 5 extracts the line number within that script, like pinpointing the specific spot within the room.

**Line 6:** error\_message = f"Error occured in python{file\_name} line number {line\_number} error message{str(error)}"

* The detective puts together a report (the error message) based on the clues they found.
* The f"..." part is like a fancy way of writing the message with placeholders for the information.
* It fills in the blanks with the script name (file\_name), line number (line\_number), and the actual error message (str(error)) to create a clear report.

**Line 7:** return error\_message

* The detective has finished their report and hands it back to you (the programmer). This report is the error\_message that the function returns.