Exercise 1.4: File Handling in Python
Learning Goals

- Use files to store and retrieve data in Python

Reflection Questions

•Why is file storage important when you're using Python? What would happen if you didn't store local files?

- File storage is pretty important in Python because it's how programs remember stuff between times they run. Think of it like notes you jot down to remember things later. Without storing files locally, Python programs would forget everything once they're done running, making it hard to keep track of settings, data, and all the important stuff they do. It's like trying to cook without a recipe book—you'd have to start from scratch every time!
-

•In this Exercise you learned about the pickling process with the pickle.dump() method. What are pickles? In which situations would you choose to use pickles and why?

- It's handy for things like caching data, sharing information between programs, or even just saving your progress in a game. It's like putting your data in a Tupperware container—keeps it fresh and ready for whenever you need it!

•In Python, what function do you use to find out which directory you're currently in? What if you wanted to change your current working directory?

- In Python, you use the os.getcwd() function to find out which directory you're currently i, If you wanted to change your current working directory, you'd use the `os.chdir()` function.
-

•Imagine you're working on a Python script and are worried there may be an error in a block of code. How would you approach the situation to prevent the entire script from terminating due to an error?

- To prevent the entire script from terminating due to an error in a specific block of code, I would use a try-except block. Within the try block, I'd place the code I suspect might raise an error. If an error occurs within this block, the script will not terminate immediately. Instead, it will jump to the except block where I can handle the error gracefully, log it, or perform any necessary cleanup operations. This approach ensures that the script can continue running despite encountering errors in specific parts of the code.