Adam Hockemeyer

2020-Mar-09

Introduction to Programming (Python)

Assignment07

**Data Pickling and Error Handling**

**Introduction**

During this assignment, I took our CDInventory script and made two significant tweaks. I changed the data handling method to use pickling to store the data, and added some error handling checks against user inputs. I also changed some of the lines of code to use with... as functionality. Since the base details of the script have been covered for weeks, I'll go mostly into the changes I made instead of the base functionality of the program.

**Data Pickling Changes**

To handle data pickling, I had to change the read to file function and the write to file function, as these would be saving and reading the saved file in a different method than we had before. Since pickled data comes out looking the same as how our program interracted with it before, there wouldn't need to be changes to how the rest of the program interacts with the data while it's in memory.

In my read\_file function, I changed the code to read pickled data by using pickle.load into the the variable 'data', then returning that variable. This is one of the bigger changes, since this changes how the function is used. Now, each time the function is called, I had to change it to update the table we use instead of having it do so as a part of the function.

In the write\_file function, I used pickle.dump out of the table passed to the function as an argument, and into the file location used as the other argument. With that change, the script officially uses pickled data to read and write.

**Using With... As**

In both the read and write file functions, I changed the code to use with... as functionality instead of opening the file, iterating through it, and closing it at the end. This consolidated multiple lines into just two lines of code, and made the funciton easier to read and understand. Note that since the script now uses pickled data, I had to use 'rb' and 'wb' instead of 'r' and 'w' as the open types, due to it reading and writing binary data.

**Error Handling with Exceptions**

The third functionality change I incorporated this week was inserting some error handling, using try and except. In both the file reading and writing functions, I encompassed the code in a try statement, with the exception being a FileNotFoundError. I then put a custom statement at the end of each to tell the user what went wrong.



*Figure 1 - the script being run in Spyder IDE, showing the load function.*

**Testing the Changes**

After setting these changes in the code, I ran the script in both Spyder and in Anaconda Prompt, testing out each menu option and taking screenshots.



*Figure 2 - the script being run in Anaconda Prompt, showing off add and save functions.*

**Summary**

I was able to change the data functions to pickle the data and read pickled data. I was also able to use error handling to make the file functions not crash the program if the inventory file doesn't exist.

GitHub link: <https://github.com/heliotropite/Assignment_07>