John Czarnek
08/22/2021
IT FDN 110 B Foundations of Programming Python
Assignment 07

Assignment 7

Introduction

This assignment was about modifying an existing CD inventory program written by someone else and then previously modified by me. The modifications were meant to replace code reading and writing text files with code reading and writing binary files, and to add error handling to keep the code from abruptly ending when users input the wrong thing, like entering a string instead of an integer for instance, or when a file was not found.

Learning

For this assignment, I read a book chapter, read a document on my computer and researched binary file manipulation and error handling on the internet. The key thing I learned about was pickle, which is a python module which allows files to be permanently written and read as digital files.

The assignment also introduced error handling and how to incorporate try … except structure to keep code from abruptly ending. Markdown language for Github was briefly mentioned in the Module 7 material.

Some of the modifications to the code included

- Modifying write and write functions from text to binary – here I used pickle to read and write and applied pickle directly to the whole table, which greatly simplified the code.
- Adding error handling to the start of the program to avoid the FileNotFoundError that would occur when the program was run without a CDInventory.dat file in the directory. I tested this multiple ways (with a .dat file with data, with a blank .dat file, with the file missing.
- Adding error handling to trap ValueError errors which occurred when the user imput data with the wrong data type in the input_CD and delete_CD functions.
- Moving print statements from delete_CD function to a separate function in the IO class to keep processing and IO separate.

Running the Script in Spyder

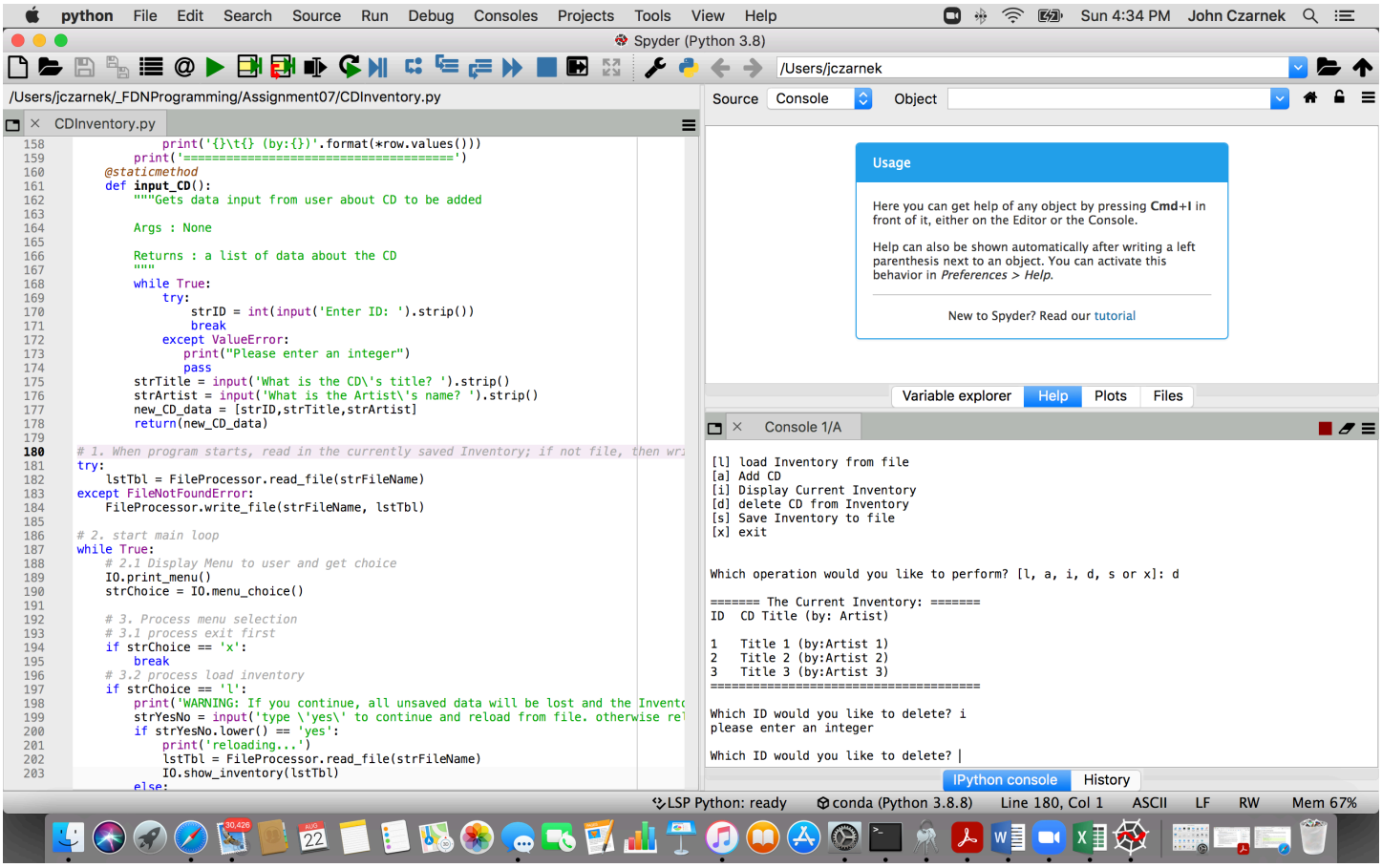

Figure 1- Script running in Spyder

Running the Script from a Terminal Window

```
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: i

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Title 1 (by:Artist 1)
2       Title 2 (by:Artist 2)
3       Title 3 (by:Artist 3)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: d

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Title 1 (by:Artist 1)
2       Title 2 (by:Artist 2)
3       Title 3 (by:Artist 3)
======================================
Which ID would you like to delete? i
please enter an integer
Which ID would you like to delete? 10
Could not find this CD!
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Title 1 (by:Artist 1)
2       Title 2 (by:Artist 2)
3       Title 3 (by:Artist 3)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]:
```

Figure 2 - Script in Terminal Window

Github Repository

https://github.com/Jczarnek/Assignment_07

Summary

I found writing code for the reading and writing of binary files using pickle to be difficult, mainly because I was overthinking and using methods applicable to text files. To help better understand what was going on, I created a test version of the program that was very simplified that did not have user interaction, and then proceeded to try different things in this simplified setup. I finally was able to see that I could read and write the whole 2D table without trying to do this row by row, which is what the textbook and internet sources seemed to suggest. My efforts to research this on the internet were fruitless, as I was searching for 'using pickle on lists of dictionaries', for instance, which was not helpful. One website that seemed like a good thing, but ended up leading me down a rabbit hole was: https://www.geeksforgeeks.org/read-list-of-dictionaries-from-file-in-python/

The error handling was easier to understand, and my favorite website for this was: https://docs.python.org/3/tutorial/errors.html