James Byrne

8/22/2021

Foundations of Programming (Python)

Assignment 07: https://github.com/Bizzyrne/Assignment\_07

Assignment 07

# Introduction

I will be modifying code submitted as Assignment 06 to add structured error handling and also utilize binary files with the pickle method.

# Error Handling

**To create this program, I will be using anaconda’s spyder. My first task for this assignment is to add structured error handling, which was added as follows:**

**FileProcessor.read\_file() – Line 61**

**Added a try/except statement to first try to open the file and return the contents as a list. If this was not successful, it prints “File does not exist”**

**IO.new\_CD\_data() – Line 157**

**Also added a try/except statement. The function first sets up a while loop and then uses a try/except statement to try to enter the CD information. If this operation is not successful due to the CD\_ID not being an integer, it prints “CD ID must be an interger” and then loops again. Once the correct information is input, the loop is broken and the function returns the CD data as a dictionary.**

**Start of the program – Line 184**

**I added a try/except statement at the start of the program to try and load file data. If the load is not successful, it prints “No Saved Data” and continues to execute without loading and existing list from file.**

**Delete portion of Menu – Line 226**

**In this portion of the code, I also included a while loop and a try/except statement. A while loop is set up and in a try statement, the program asks which CD\_ID to delete. The user inputs something and then the program attempts to convert that input to an integer. If its not successful, it prints “Please input an integer” and loops back into the loop statement. Once an integer is entered, the program continues.**

# **Converting Code to Binary Files**

# **To convert the code to use binary files, I first had to import the pickle method, which I did at the top of the code. The remaining portion of the work is in the DataProcessor class, specifically in the read\_file() and write\_file() functions.**

**Read\_file() – Line 61**

Converting the code to binary files simplified things quite a bit. I simply had to open the file using the ‘rb’ option, used the pickle.load() to assign the data to a list, close the file and then return the list data.

**Write\_file() – Line 85**

Binary files also simplified this portion of the code. I opened a file using the ‘wb’ option, used the pickle.dump() to write the current list in memory to file and then closed the file.

# **Using the Code**

# **Once I had added the error handling and use of binary files, I checked the code. The code in Spyder can be seen working in** Figure 1**.**

Text

Description automatically generated

Figure - Code working in Spyder

The code can also be seen working in a terminal window in Figure 2.

Text

Description automatically generated

Figure - Code working in terminal window

# Summary

This assignment focused on error handling and using binary files. Error handling wasn’t too hard to do, but for some reason it took me 6 hours to get the binary code to work. Not my favorite way to spend a Saturday ha ha.

# Appendix

--