Tim Fang

2021-Nov-28

Foundations of Programming (Python)

Assignment 07

Text/Binary Files and Structured Error Handling

# Introduction

This document will describe what I learned from Module 07, outline the steps I took to complete the seventh assignment, and summarize some challenges. For this assignment, I will continue using Spyder to modify a CD inventory program. Using my previous assignment, I will modify its script to work with binary files and include error handling. I will execute the program in both Spyder and Terminal.

Thank you for your feedback with each assignment. In addition to this week’s assignment, I have modified the previous *IO.user\_input* function to separate the I/O from the DataProcessor codes. I understand that work will not affect this assignment’s grade but would still appreciate your feedback.

# Program Planning

I worked through this week’s module as per usual. When I reached Lab B, that’s when I hit what felt like my own version an error/exception. Like Python, I halted, threw all sorts of error messages, and crashed. At the time of this writing, I still don’t understand how to add “functionality to select the operation mode via program arguments.” I had to leave that part of the lab behind and move forward to binary files and error handling.

Work on this week’s program began with learning what I needed to improve from last week’s assignment. I updated the *read\_file* and *write\_file* functions to use local variables vs previous global variables. I also learned that processing user input is considered data processing and really belongs in the latter class. A solution was used, and the final script is what I used for this week’s program.

# Final Script

I tested the program again to find the exceptions. That helped me hone in on where the errors were occurring. I also tested the try/except construct within function(s) as well as outside the class/function(s) to figure out what worked best. The figures below show the program working with the focus on error handling as I cycled through each option. Error handling was added to choices ‘a’, ‘d’, and ‘i’ outside of their related functions. The last error handling was added to the *read\_file* function. Pickling and unpickling were also added to the *read\_file* and *write\_file* functions to work with binary files.

Figure 1 shows option ‘L’ at the start of the program with error handling to account for missing .dat file.

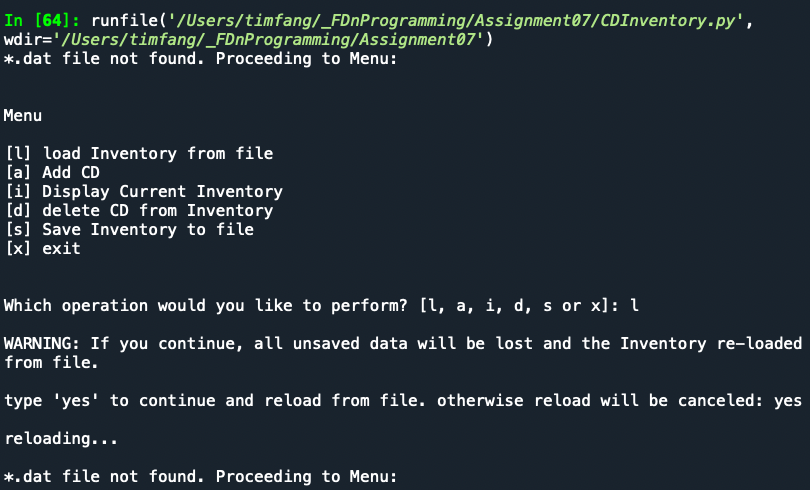


Figure 1 - CDInventory Program, option 'L'

Figure 2 shows option ‘A’ with error handling to account for non-integer values for ID. I tried to account for blank CD titles and Artist names but couldn’t master those exceptions in time.

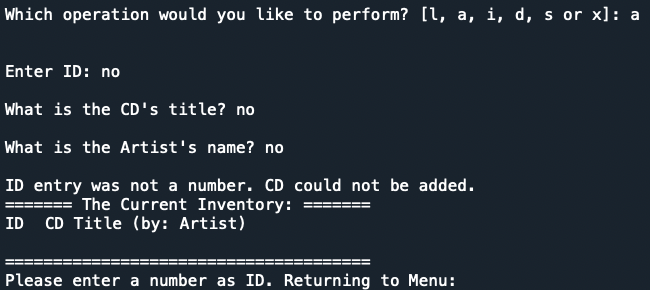


Figure 2 - CDInventory Program, option 'A'

Figure 3 shows option ‘D’ with error handling to account for non-integer values for ID.

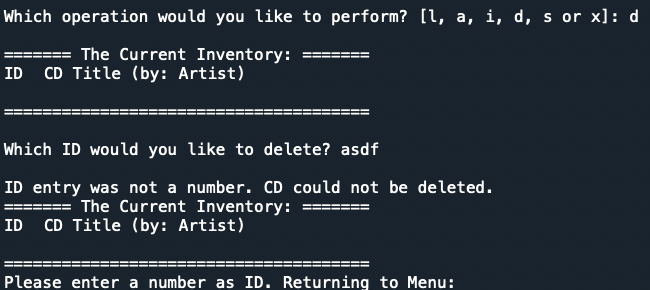


Figure 3 - CDInventory Program, option 'D'

Figure 4 shows a modified program behavior to display the inventory after unpickling.

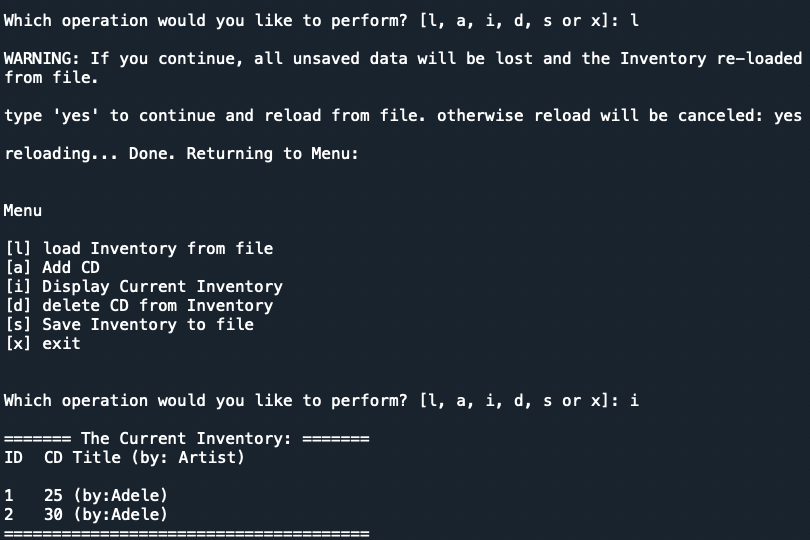


Figure 4 - CDInventory Program, option 'L'

Figure 5 shows the code modification I made to the option ‘L’ after unpickling. Figure 6 shows the code modification I made to option ‘I’ to display *lstTbl* if the variable *data* is empty (ie. showing inventory before loading from .dat file).

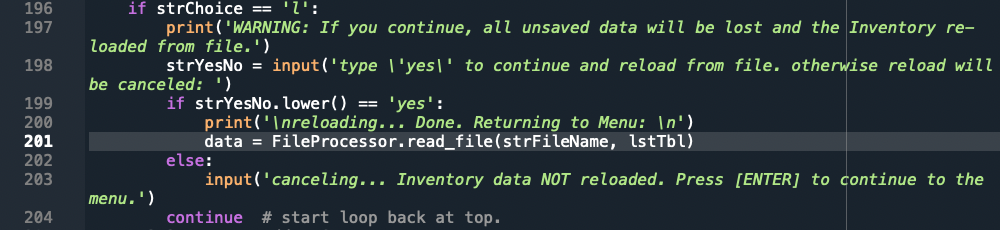


Figure 5 - CDInventory Program, loading after unpickling

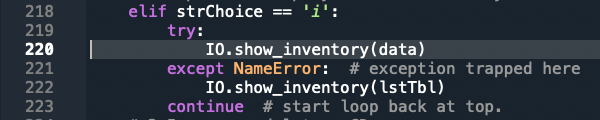


Figure 6 - CDInventory Program, error handling to display in-memory data if there was no unpickling

The following figures show the same error handling in Terminal.

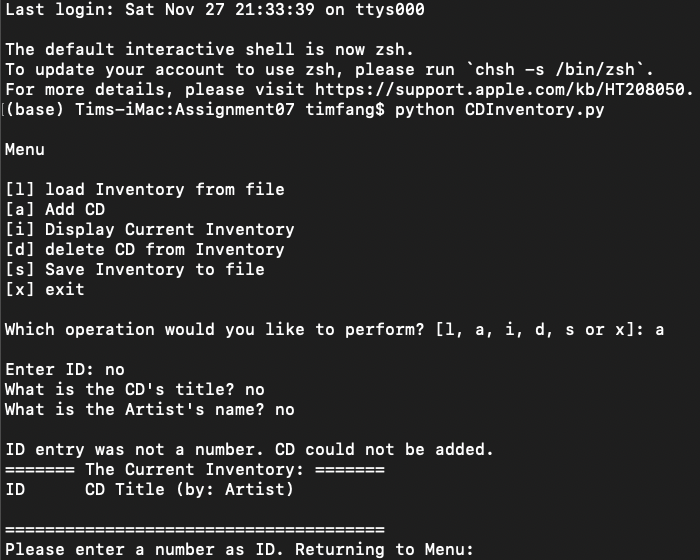


Figure 7 - CDInventory Program, option 'A' with error handling

# 

Figure 8 - CDInventory Program, option 'D' with error handling

Figure 9 shows pickling data to binary .dat file, unpickling, and the inventory view after unpickling.

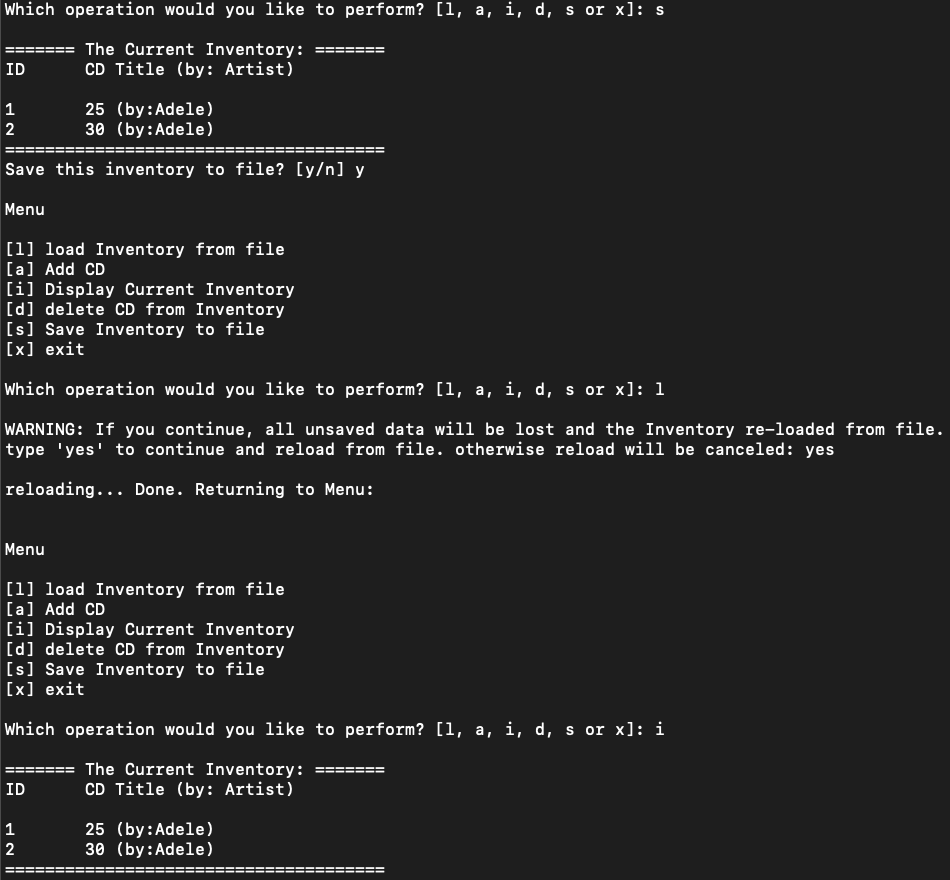


Figure 9 - CDInventory Program, options ‘S’, ‘L’, and ‘I’

# Summary

The online document I found that was somewhat helpful in explaining error handling can be found [here](https://www.tutorialsteacher.com/python/exception-handling-in-python). The online document that helped explained pickling/unpickling can be found [here](https://realpython.com/python-pickle-module/).

Please forgive me for a lighter summary this week compared to previous weeks. My main frustration and stumbling block are still with program arguments. The video from [Module 03](https://www.youtube.com/watch?v=rywVADu3dp4) that might have helped was cut mid-topic. I’ve spent most if not all my free time during this long weekend trying to reach the assignment deadline without dropping out in frustration. I hope you had a better holiday week.

# Appendix

<https://github.com/timsfang/Assignment_07>