Emily Martin

03/13/2022

IT FDN 110

Assignment 07

# Introduction

The coursework for Module 06 introduces the concept of structured error handling and pickling. Module 06 also iterates on previously introduced concepts of opening, closing, and writing to or appending files, and introduces the ‘with’ statement. In this assignment, I used the knowledge gained in Module 07 to iterate on a previously created CD Inventory program by swapping out text file operations for pickle file operations, and adding structured error handling where necessary.

# Approach

The following steps document the general approach taken to produce the source code referenced in the Appendix, as well as the reasoning behind a selection of notable syntactical decisions:

1. Based on the course recommendation for IDEs, I created a Python file in Spyder to write my code in. I added a header to capture administrative details.
2. First, I created a list of to-dos in the existing code, and created a separate list of items external to the IDE to track my progress. This allowed me to address each to-do more strategically, and enabled me to add notes to the list to capture additional follow up items.
3. I first swapped out the text file operations for pickle file operations in read and write file functions in the FileProcessor class. I added a return statement to the end of each previously listed function to capture the output of the function after running my code and noticing that the pickle file was not correctly persisting the content of the CD inventory.
4. After surveying the main loop, I decided to create a new function – delete\_cd\_prompt – to capture the prompt to delete a CD. This would allow me to introduce cleaner error handling around cases where the user did not enter an integer.
5. I then added error handling to the code. I split the error handling into major cases:
   1. User Interaction: when there’s logic dependent on input from the user.
      1. In class IO, there were two functions (menu\_choice and load\_inventory\_prompt) that when assessed, I felt had no need for error handling. In these cases, I left comments to explain my reasoning.
   2. Type Casting: when a string is re-cast as an integer.
      1. This was needed for two functions: delete\_cd\_prompt and add\_cd\_input. In both cases, the addition of error handling (to cover cases where a user did not input an integer) was necessary.
   3. File Access Operations: when file is access is attempted.
      1. For write\_file – which when called writes binary to the pickle file, I added general error handling that prints the error to the user.
      2. For read\_file – which unpickles and reads from the file into memory, I added both general error handling, and more specific error handling for the FileNotFound error. This is necessary for the program to run prior to pickle file creation, and additionally handles cases when the file may be deleted out from under the user during runtime.
6. As a last step, I surveyed all of the new and existing functions updated the docstrings where appropriate to accurately capture how each function is intended to perform.
7. Upon completion, I uploaded my assignment to GitHub for review.

# Summary

After iteratively testing the changes applied this code, I was able to verify that the CD Inventory program maintained the full scope of intended functionality. Additionally, I verified that I was able to both save binary data to a pickle file, and load that data in from the unpickled binary file. In summary, this assignment required knowledge from both current and previous modules to streamline the program.

# Outcome

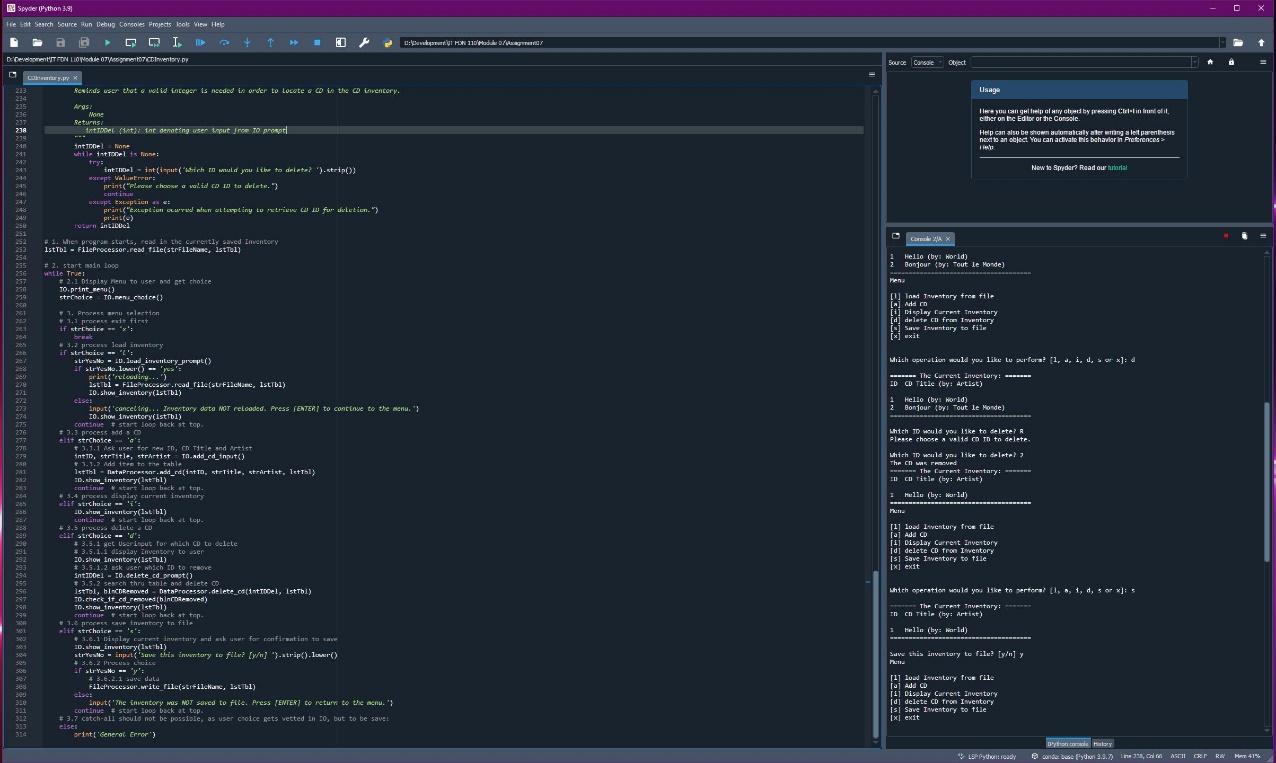


Figure : Capture of code running in Spyder

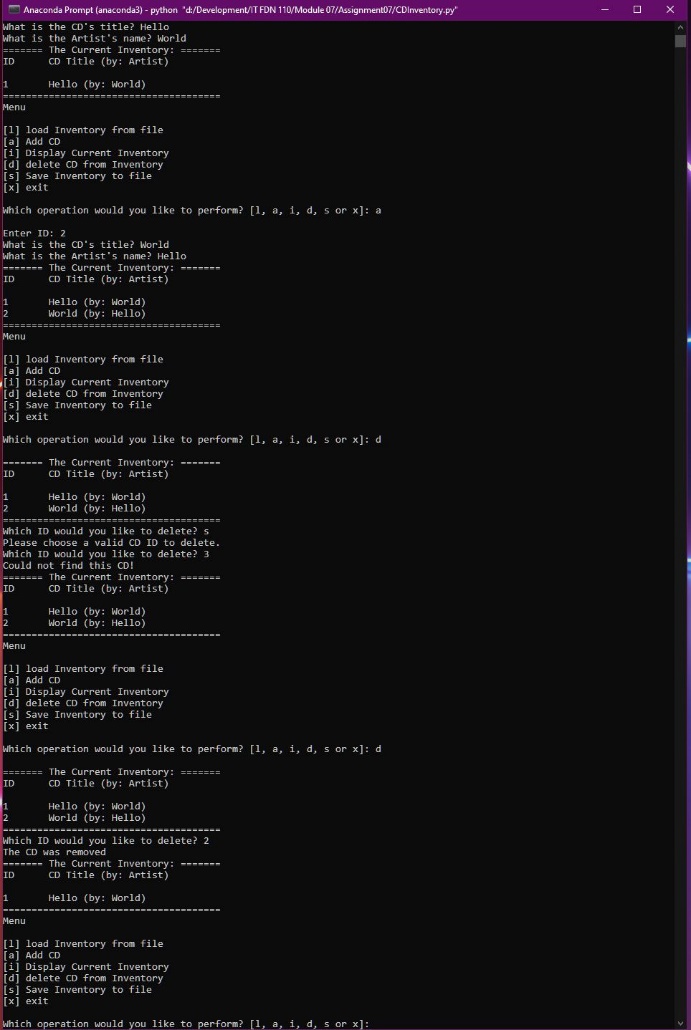


Figure : Capture of code running in a prompt

# Resources

As indicated in the Assignment 07 instructions, I searched the internet for external resources to further explain exception handling and pickling in Python. Below are examples of resources I learned from, and short explanations as to why these resources were helpful.

## Exception Handling in Python

1. [Tutorials with Nana: Python for Beginners.](https://www.youtube.com/watch?v=t8pPdKYpowI) The video format of this reference was easy to follow along with. Watching someone write error handling helped me grasp the syntax.
2. [Real Python: Python Exceptions.](https://realpython.com/python-exceptions/) The distinction between exceptions and syntax errors was a useful one. This article also goes into detail about how to write try and except blocks that I found helpful.

## Pickling in Python

1. [After Nerd: Pickling in Python](https://www.afternerd.com/blog/python-pickle/#:~:text=Pickling%20in%20Python%20means%20the%20process%20of%20serializing,First%2C%20let%E2%80%99s%20understand%20what%20serialization%20and%20deserialization%20mean%3F). This resource had easy-to-reference formatting for writing code to pickle objects in Python. I referenced it while coding my assignment.
2. [Geeks for Geeks: Understanding Python Pickling.](https://www.geeksforgeeks.org/understanding-python-pickling-example/) This resource had a section noting the advantages of using the Pickle Module, which helped me to understand the applications of pickling.