Austin Sivret

March 3 2020

IT FDN 100

Assignment 7

Assignment 7

# Introduction

In this assignment I got some exposure to more read/write functions using files, and I was given my first taste of binary. I also learned how to use Structured Error Handling in my code, and how to create my own exceptions. In the reading, I used these concepts to walk through a trivia game. Finally, to document my knowledge, I modified the CD script from previous assignments using Structured Error Handling, as well as pickling to binary for file saving.

# Module Videos

## Lab 07-A



Figure 1 - Script for Lab A

I wrote two functions. The first opens a file and reads the first line into a list called data. The first and third values are then saved as numA and numB (the second value is a comma). The second creates an empty string and for each result in the results argument, adds to the empty string with a comma. Then I get rid of the last comma and write the output to the file.

## Lab 07-B

## 

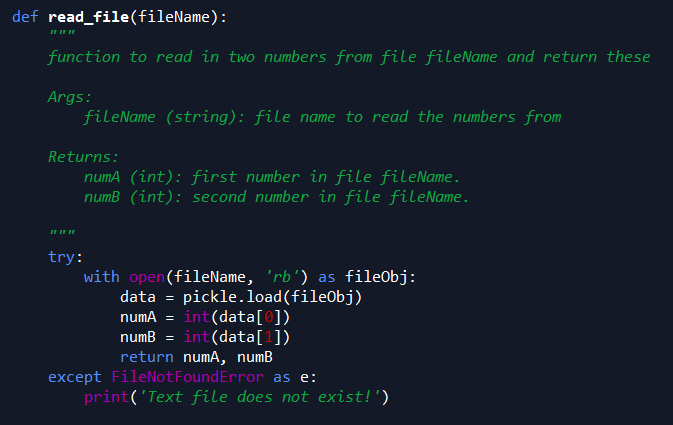
## 

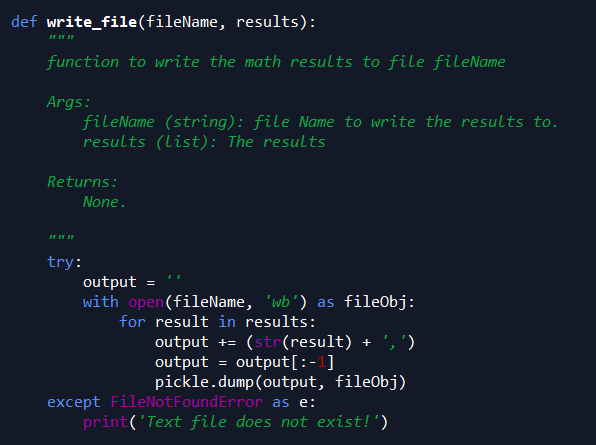
Figure - Lab B

For this lab, I took the code in Lab A, and modified the functions to use pickling. As a result, it is pretty much the same except it works with pickled numbers instead of normal numbers. I also switched to *with/as* notation. One downside of this code is that it requires pickled code to begin with. I’ll have to ask about this in class.

## Lab 07-C

For lab C, I added some error handling.





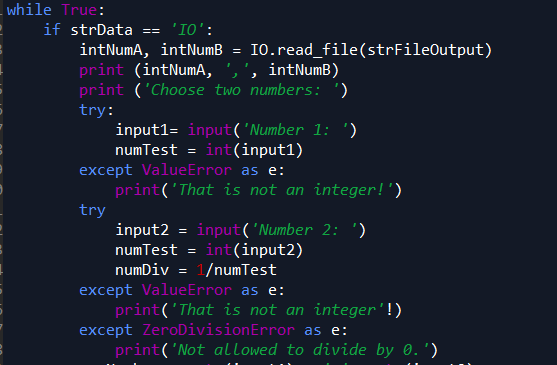


Figure - Lab C

In this lab, I added some exceptions to catch if the user tries to use a non-existent text file (top two images) or for I/O errors. For the I/O errors (bottom image), an exception will be raised for an input that cannot be converted to integer for either number, or if the second number is zero.

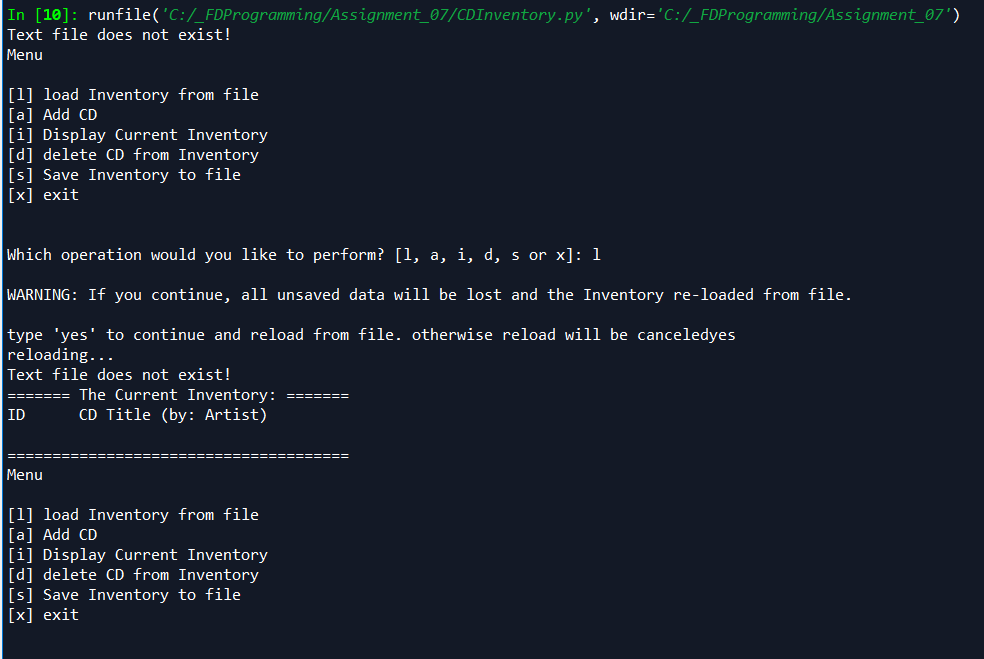
# Book Chapter

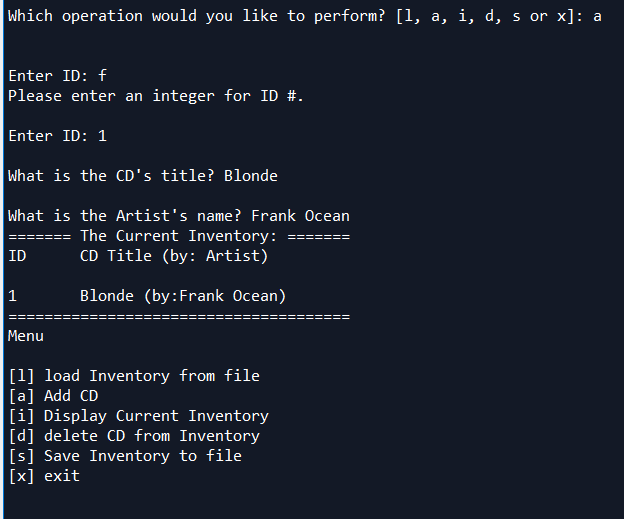
In the book chapter I learned about files and exceptions. More specifically, I learned about different ways to read and write with files. I learned how to “pickle” files in binary and about other binary functions. Finally I learned about exceptions and how to handle errors.

# Web Pages

I read a webpage[[1]](#footnote-1) on error handling that went over some new functions, *finally* and *raise*.

# Apply Your Knowledge





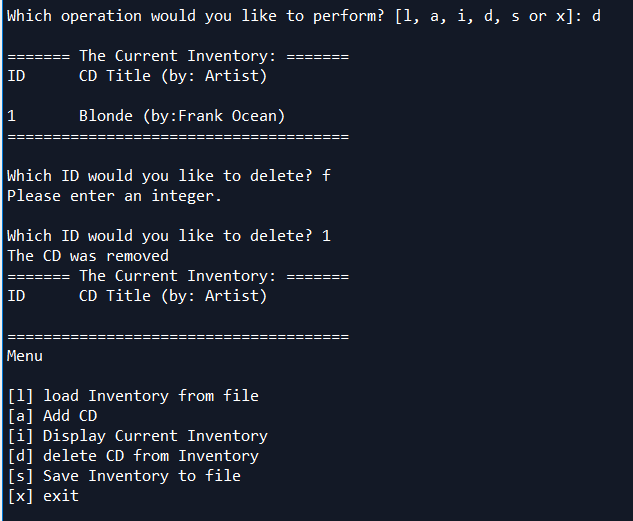
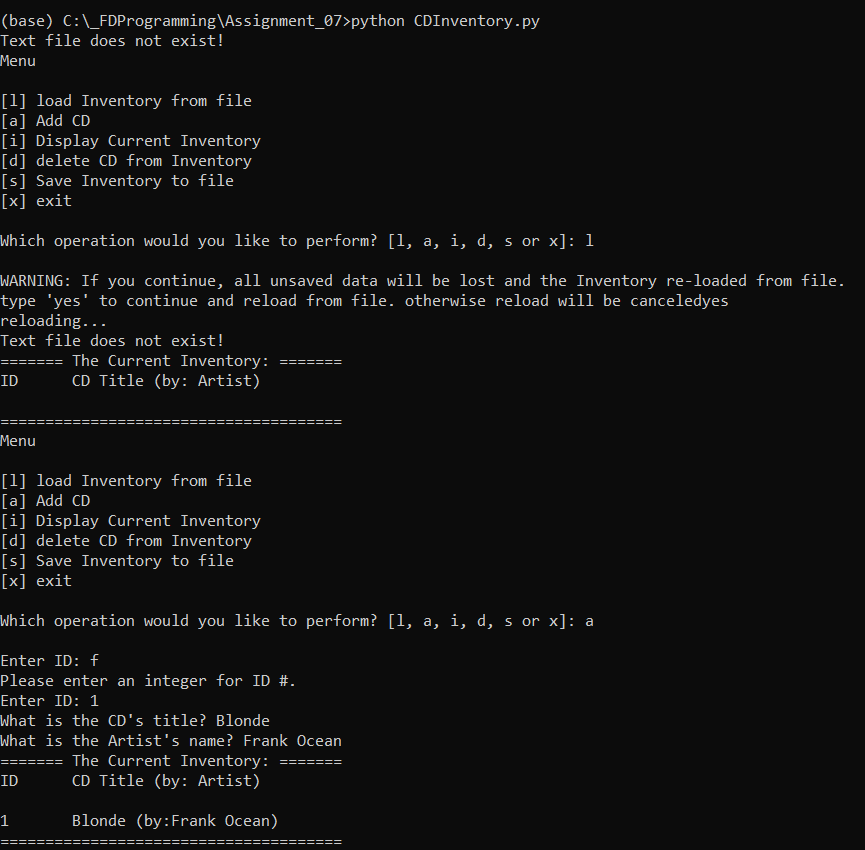


Figure - Output in Spyder showing new features.



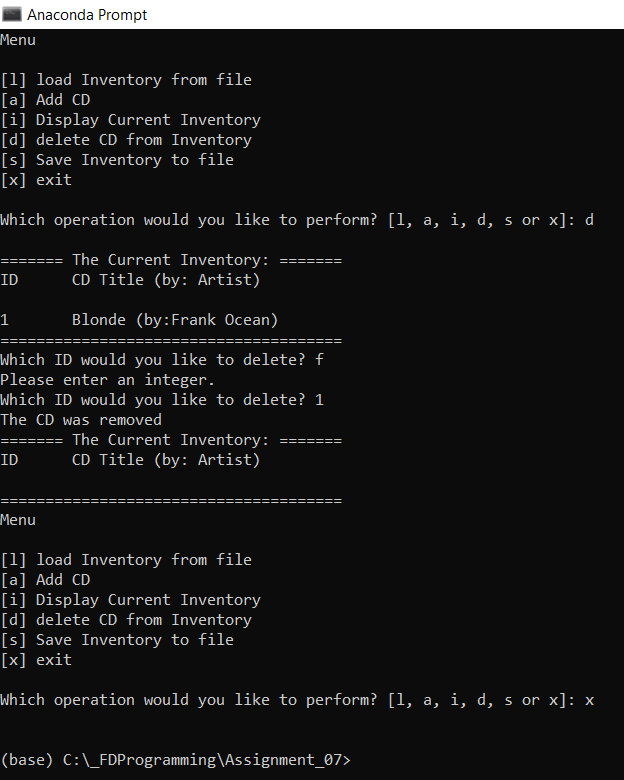


Figure - Command prompt running.

For this code, I added some error exceptions to catch some common errors that may occur. The two main areas errors could occur are with file handling as well as user input. I added an exception if there is no file and the user tries to load it. I also added exceptions if the user tries to enter an ID number that cannot be converted to *int* (like a string). Finally, I changed it to work with binary files by adding some pickle functions. Although it was a little annoying that I couldn’t really test the “load” feature very well by adding a line in the text file because I’m not well versed in the binary language.

# Summary

In this module, I focused on new ways to read and write to and from text files. I also learned about pickling and using binary files. Finally, I learned how to handle errors in my programs using *try/except* to raise exceptions, rather than letting the program die when there is a file issue or when the user enters something unexpected.

1. <https://www.programiz.com/python-programming/exception-handling> accessed March 9 2020 [↑](#footnote-ref-1)