Kimberly Brock

February 21, 2021

Foundations of Programming (Python) – IT FDN 110 A

Assignment07

GitHub: [kb1981/Assignment\_07 (github.com)](https://github.com/kb1981/Assignment_07)

Introduction

This week we learned more about writing and reading to text file and data files. We discussed and practiced using the pickle function for reading and writing to data files. We also discussed exception handling using python’s built-in functionality. Exception handling allows a program to not exit unintentionally because of a ‘bad’ user input or coding error.

Module 07 – Text and Binary Files

We begin Module 07 with a discussion on reading and writing to a text file. We reviewed the read, write, and append functions and how to use loops to read through a file. We learned using the with … as option is efficient at opening files and eliminates the need to remember a close function. The next part of Module 07 discussed binary file format and introduced the ‘pickling’ concept. The python pickle function takes python information and preserves it in a binary code. This can save time and space, depending on its usages. The last parts of Module 07 deals with structured file handling. Python has an internal feature of handling errors that happen in scripts. Because of this feature, we can use those codes to help our scripts not crash and burn when bad information is input into the program. The best option is to utilize the error codes that are already built into Python, but there is an option to create your own unique error codes as well. TO wrap up the module, we learned a little about markdown language in GitHub, which is something I utilize regularly using Azure DevOps.

*Module 07 can be found at the link:* [*https://saravji.github.io/saravjis\_hut/FDN\_Prog/Modules.html*](https://saravji.github.io/saravjis_hut/FDN_Prog/Modules.html) *and the Lab07\_A, B, C are shown in Appendix A.*

Python Programming for the Absolute Beginner, Third Edition

This week in the book *Python Programming for the Absolute Beginner, Third Edition*, By Michael Dawson I read through. “Chapter 7–Files and Exceptions: The Triva Challenge Game.” The chapter went through some basics of opening and closing a text file. The chapter discussed opening a text file and reading it line by line using a couple of different options. The book discussed how once you start reading from a file, in order to reset the curser, you must close the file and open it again. We went through the difference of readline() and readlines(). Readline reads characters from the current line only, whereas readlines reads all the lines into a list. This works the same with writeline() and writelines(). Next topic was using the pickle module. Pickling an object stores it to a .dat binary file, it cannot be stored in a text file. Pickling stores the objects, and unpickling unpacks the objects so they can be used. Another function to use with the pickle module is the shelf module, that acts like a dictionary. Exception Handling was also discussed in this chapter. Using the Try/Except feature, you can catch errors and handle them before they crash your script and shut it down. Python is very good at giving feedback on what went wrong in a code, now we can use that feedback to predict if something may happen in our scripts and control that reaction in the program instead of getting an error.

Research Exception Handling in Python

1. [Python Exceptions: An Introduction – Real Python](https://realpython.com/python-exceptions/)

I thought this link was thorough at explaining exception handling. The link provided examples on the multiple ways to handle errors, like extending on the try/except option.

1. [HandlingExceptions - Python Wiki](https://wiki.python.org/moin/HandlingExceptions)

The link above is a decent example. The one think I like about this link is a link inside this link (that I now see was shown in the Module07 video). [Built-in Exceptions — Python 3.9.2 documentation](https://docs.python.org/3/library/exceptions.html) << this website likes a hierarchy of exceptions in Python that I find very useful.

Research pickling in Python

1. [pickle — Python object serialization — Python 3.9.2 documentation](https://docs.python.org/3/library/pickle.html)

The above link is a good, detailed link for describing the pickle function in python

1. [How to use Pickle to save and load Variables in Python? - GeeksforGeeks](https://www.geeksforgeeks.org/how-to-use-pickle-to-save-and-load-variables-in-python/)

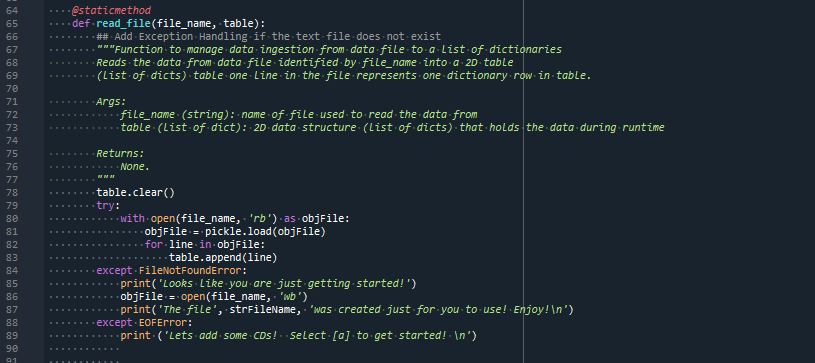
This link is a nice to-the-point description of the pickle function in python

Applying Knowledge – CDInventory.py, **more** *enhanced*

This week, we enhanced our CDInventory.py file to include writing to a date file and to catch errors using Exception Handling.

**Add Structured Error Handling**

The script below is handling (1) if there is no file created yet (FileNotFoundError) and (2) there is a file, but it is empty (EOFError)



Listing 1. CDInventoy.py File Error Handling

*Scenario 1: No file found*

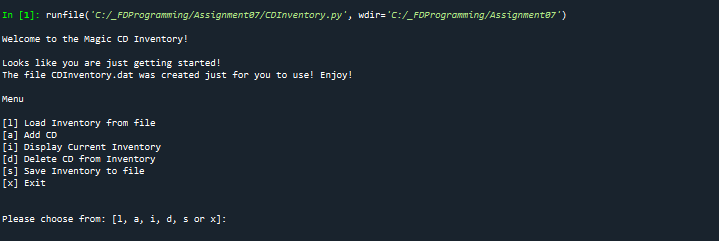


Figure 1. CDInventory.py in Spyder (Scenario 1), FileNotFoundError

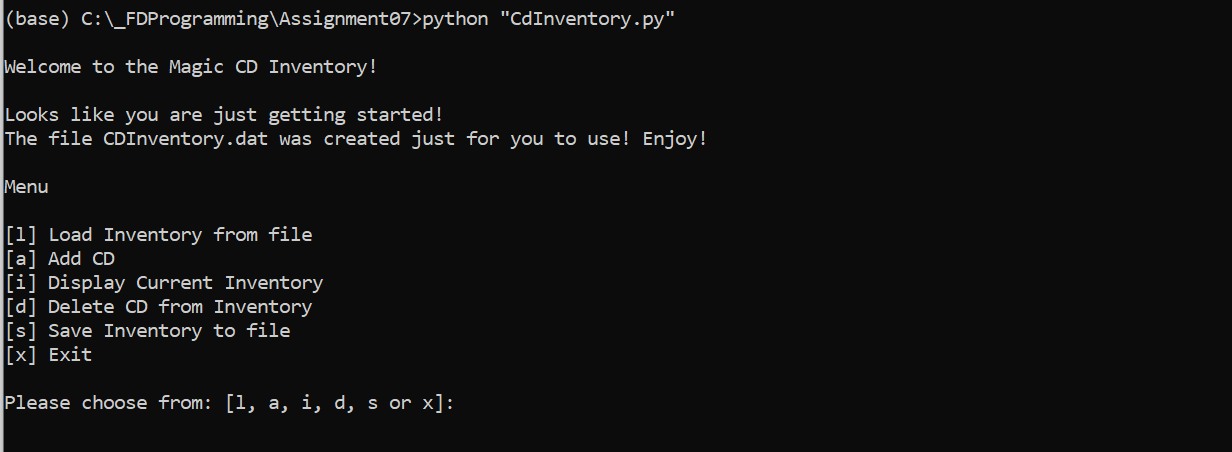


Figure 2. CDInventory.py in Terminal (Scenario 1), FileNotFoundError

*Scenario 2: Empty File*

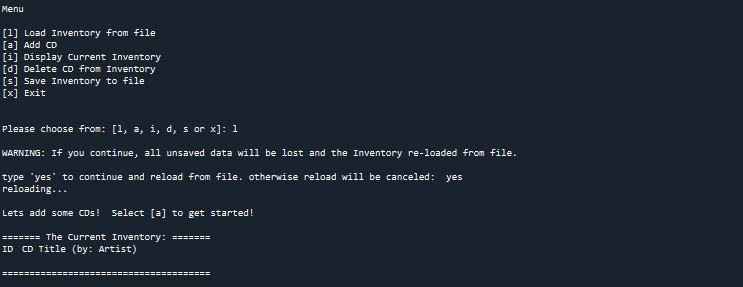


Figure 3. CDInventory.py in Spyder (Scenario 2), EOFError

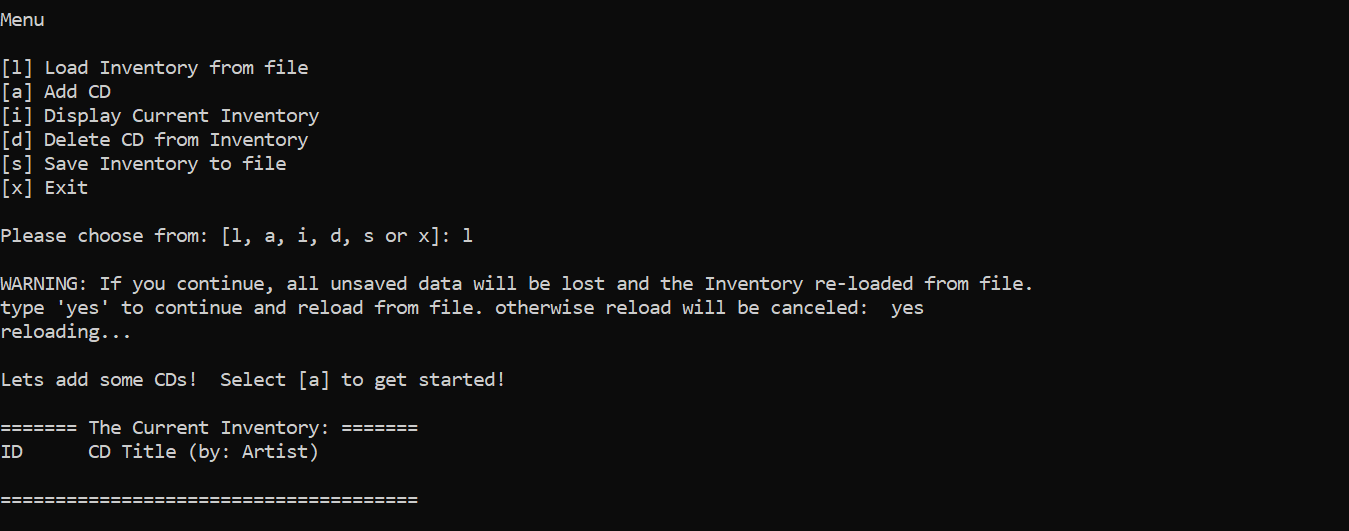
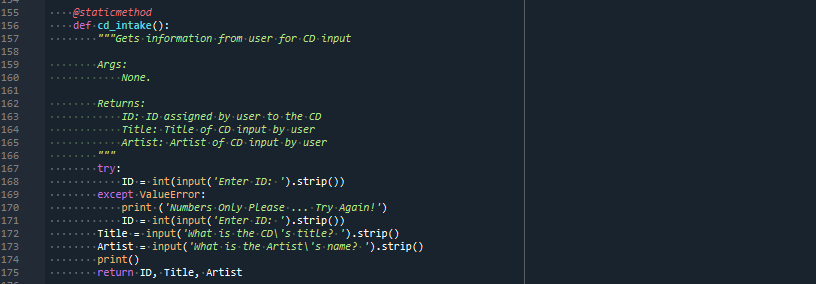


Figure 4. CDInventory.py in Terminal (Scenario 2), EOFError

The script below is handling the ValueError to make sure the CD ID is a number.



Listing 2. CDInventory.py ValueError Handling

*Scenario 3: Number not entered*

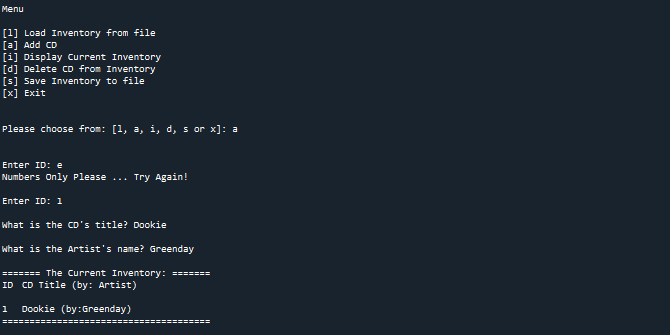


Figure 5. CDInventory.py in Spyder (Scenario 3), ValueError

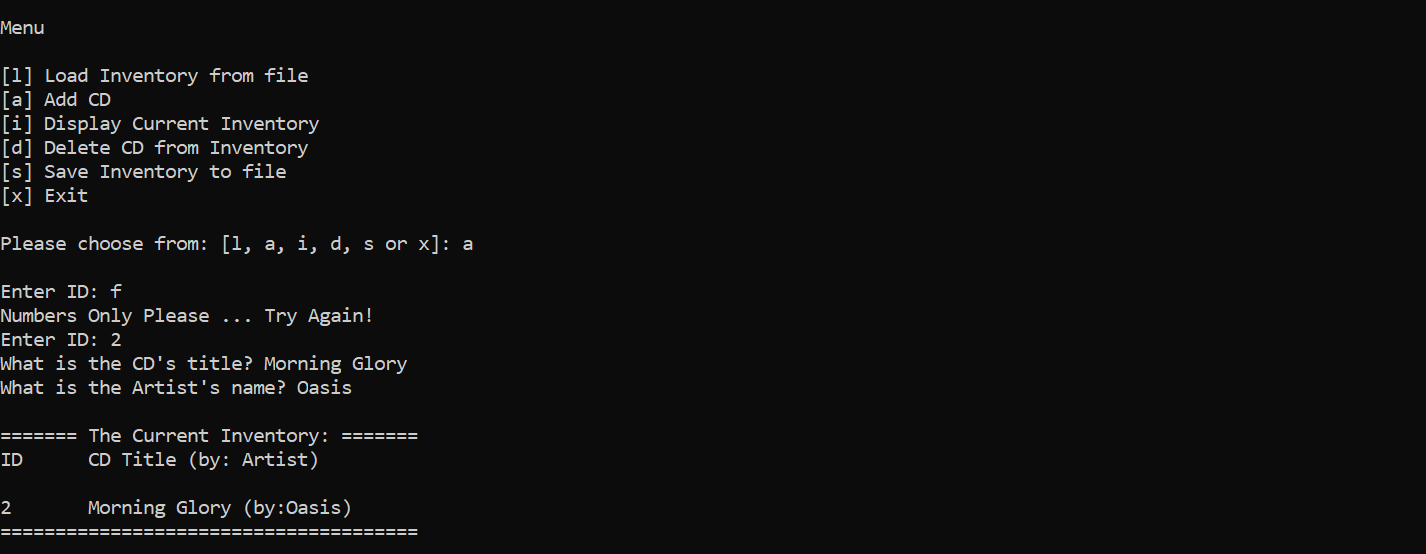
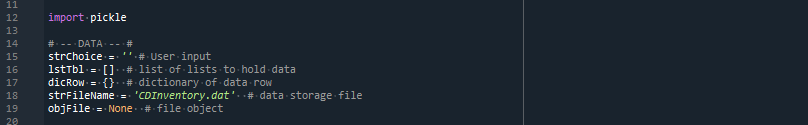


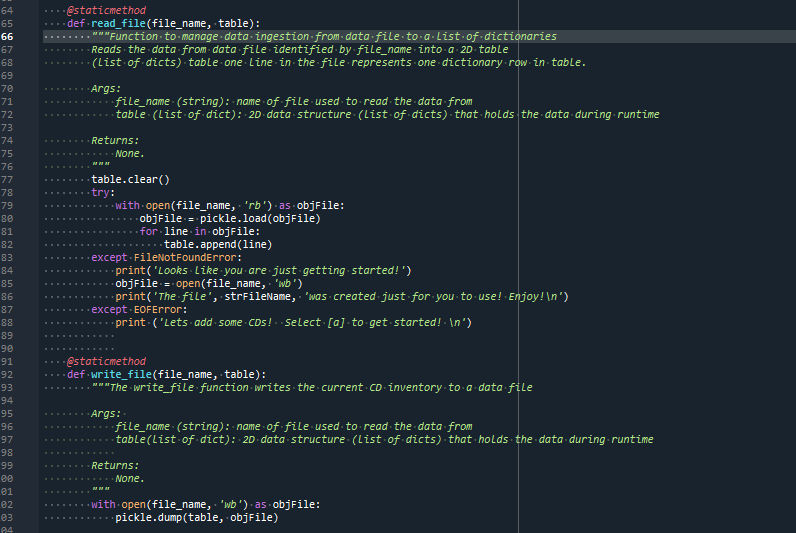
Figure 6. CDInventory.py in Termina (Scenario 3)l, ValueError

**Store data in a Binary Data File.**

The script below is showing the changes made to store information in a binary data file.



Listing 3. CDInventory.py Updates to call pickle function and change default file to a .dat



Listing 4. CDInventory.py using the pickle function in the functions read\_file and write\_file to write binary data to the .dat file.

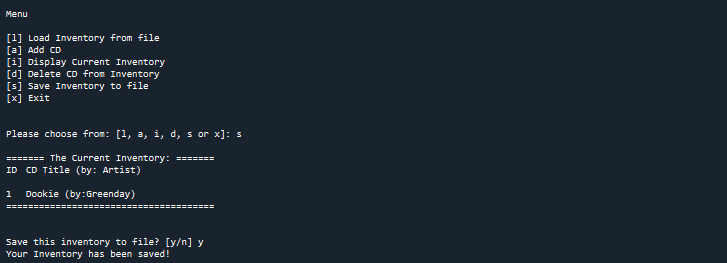


Figure 7. CDInventory.py saving data to CDInventory.dat in Spyder

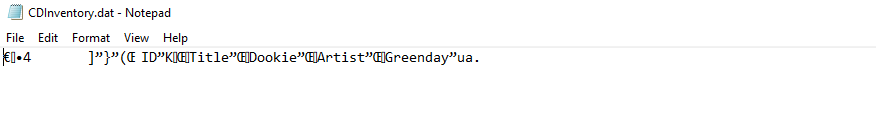


Figure 8. CDInventory.dat file after input (Spyder)

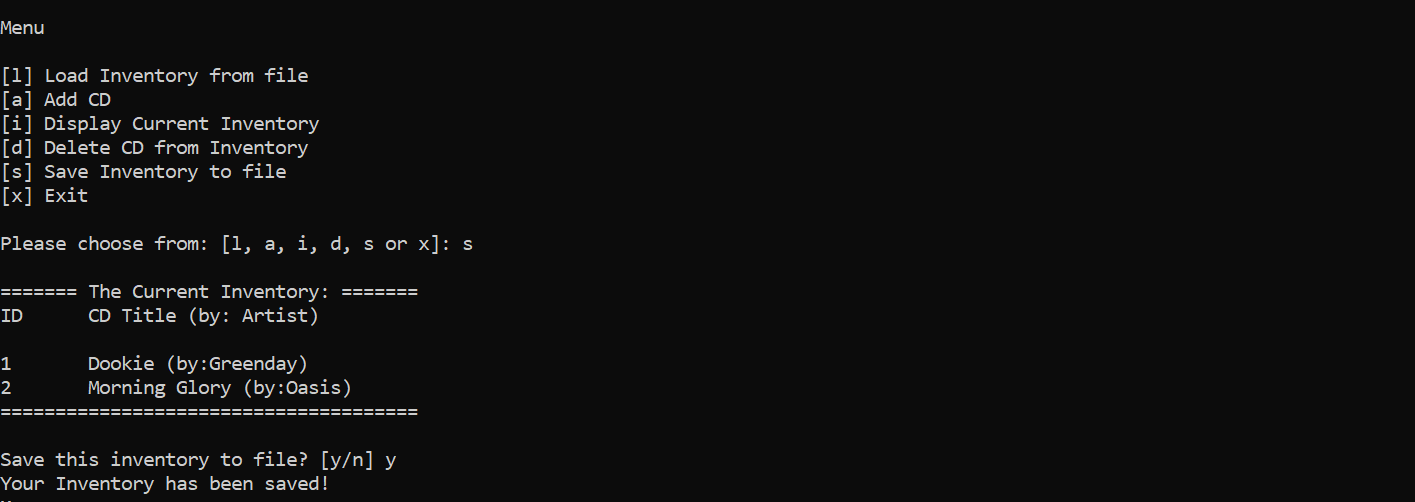


Figure 9. CDInventory.py saving data to CDInventory.dat in Terminal

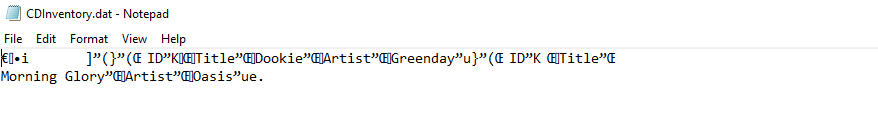


Figure 10. CDInventory.dat file after input (Terminal)

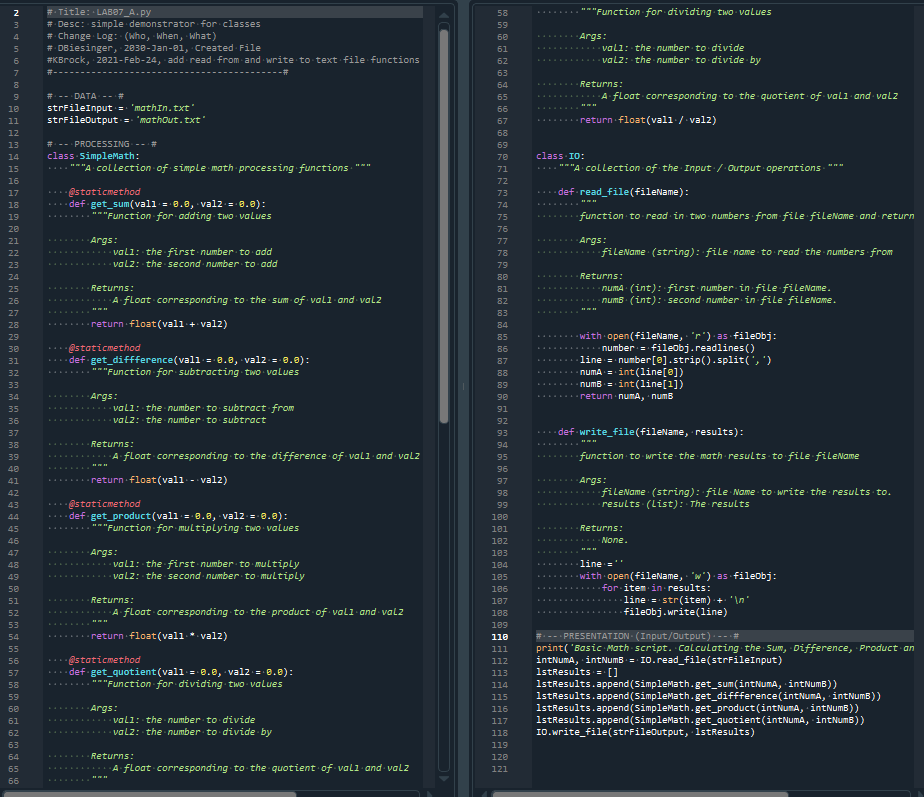
Summary

We went through Module 07, that discussed reading and writing to files that are both text and data and exception handling. We read chapter 07 in the book, that took us through reading and writing to a file to create a Trivia Game. We did some research on Exception Handling and the Pickle function. Finally, we updated our CDInventory file to be a data file and included Exception Handling. I thought this module was very interesting, and kind of blew my mind. It took me a little bit to grasp is concept of pickling, but I do understand now, and looking forward to utilizing this in the future.

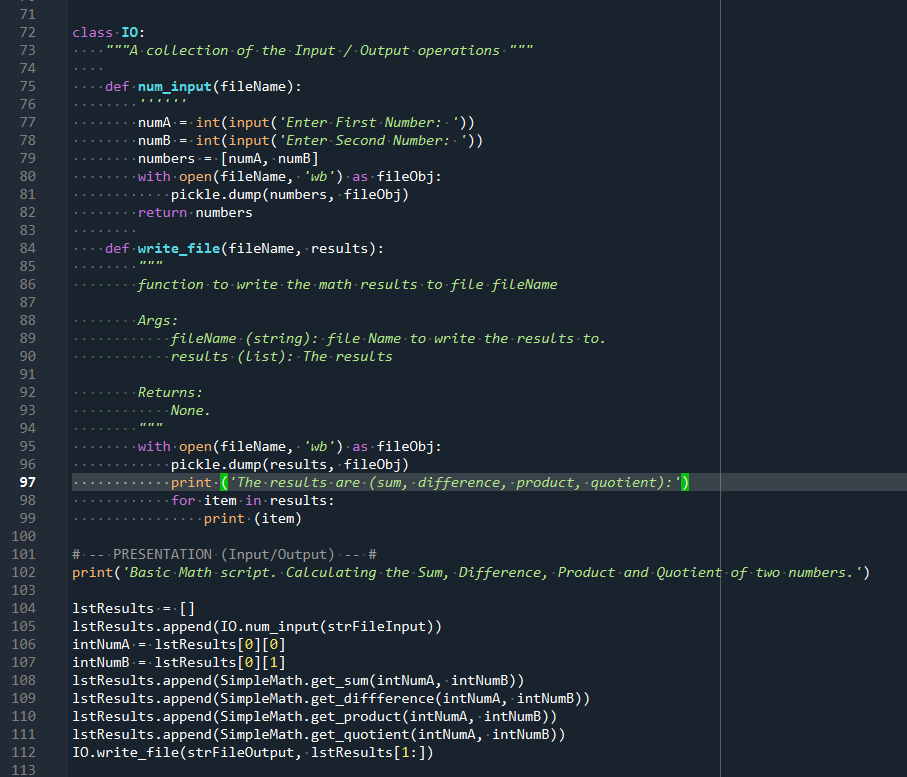
Appendix

*Appendix A. Module 07 Lab Listings*

Lab07\_A.py calls a function named *read\_file* to open a file named “mathIn.txt”. When the file is open, the function reads in two numbers from the file, and passes them back into the loop. The loop then used those new values and calls a series of functions to perform basic math and saves that output to a list. The loop then calls the function named *write\_file* to open a file named “mathOut.txt” and saves the outputs in a formatted manner.

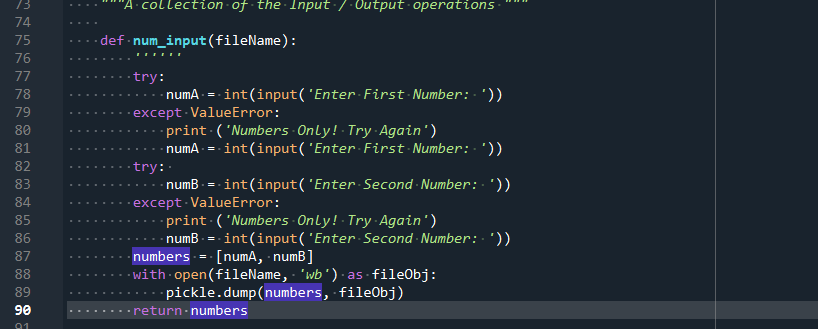


Appendix A Listing 1. Lab07\_A.py

For Lab07\_B.py, not going to lie, I did not do this right. I could not figure out the Sys Arguments. I do not know why my brain is not accepting sys arguments information. I did however update the code to use the pickled data file to write the results to, but pretty sure I am not utilizing the reading. I am pretty sure I messed this lab up completely! Doh! 

Appendix A Listing . Lab\_07.py

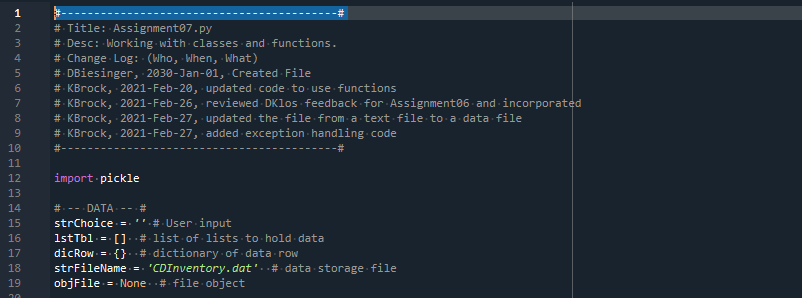
Lab07\_C.py. I added exception handling to a messed-up script! ☹



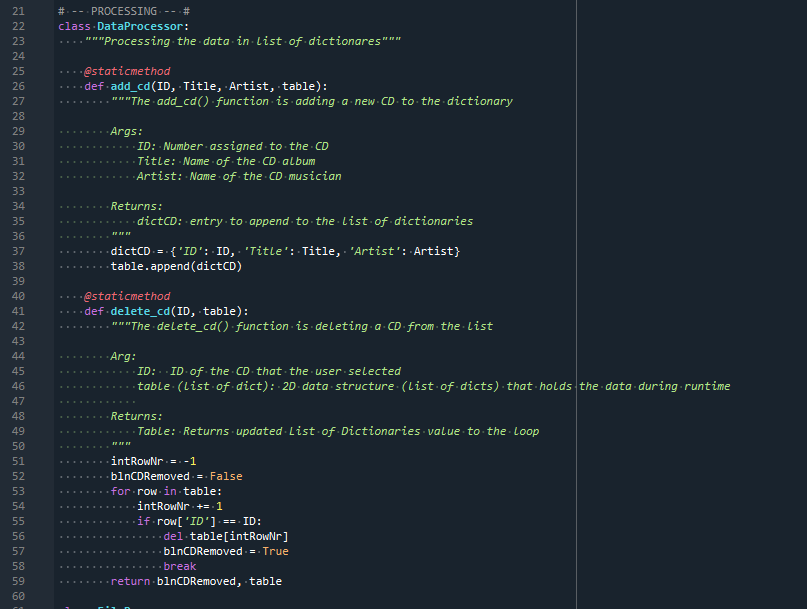
Appendix A Listing . Lab07\_C.py

*Appendix B. Assignment07 – CDInventory.py*

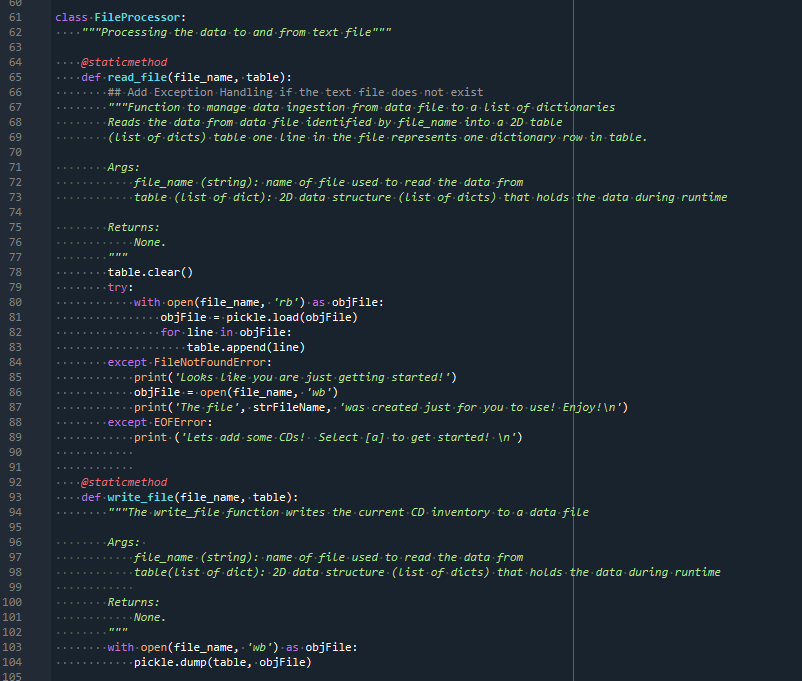
Assignment uploaded to GitHub: [kb1981/Assignment\_07 (github.com)](https://github.com/kb1981/Assignment_07)



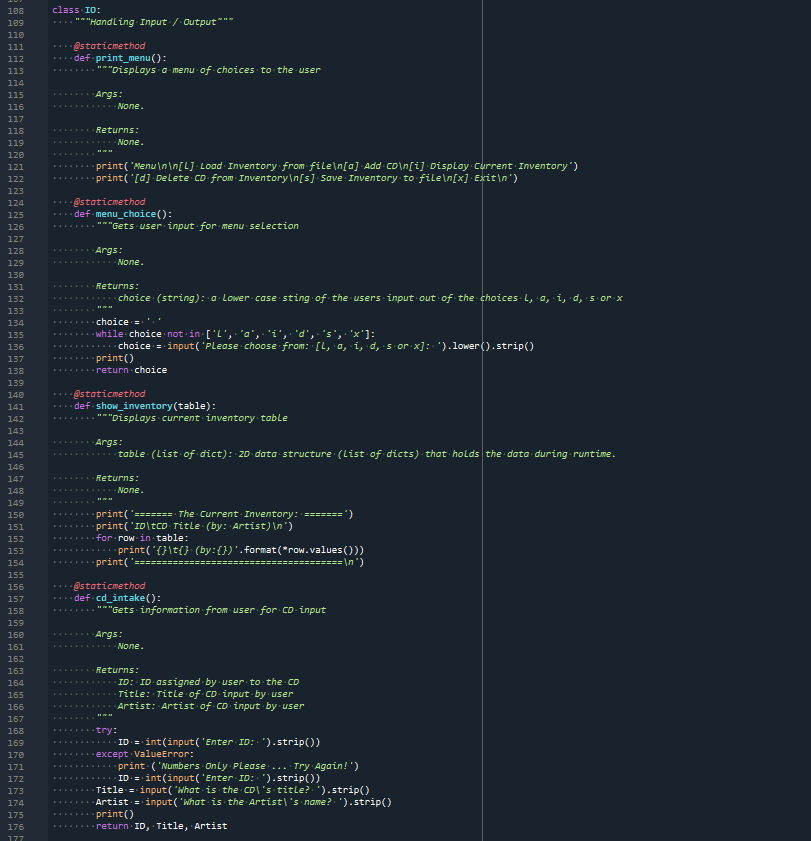
Appendix B Listing . CDInventory.py data



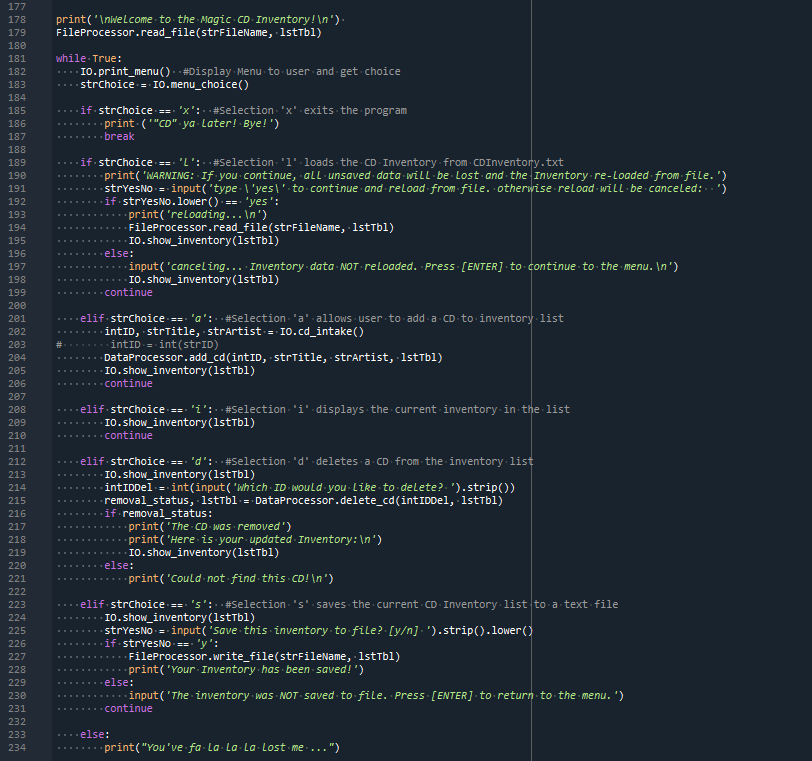
Appendix B Listing . CDInventory.py class DataProcessor functions



Appendix B Listing 3. CDInventory.py class FileProcessor functions



Appendix B Listing 4. CDInventory.py class IO functions



Appendix B Listing 5. CDInventory.py main loop