Rudolph Carvajal

August 28, 2020

Foundations of Programming: Python

Assignment\_07

CD Inventory: Pickling and Exception Handling

# Introduction

While we have learned how to read and write to text files in prior modules, we spent time in this week’s module learning how to store more complex data. This week, we learned about the pickle and shelve modules in python, making it much easier and efficient to store data to DAT files. We also went deeper into exception handling, learning how to trap specific exceptions. In this document, I will discuss the principles I learned around pickling data and handling exceptions. I will then re-visit the CD Inventory program and discuss what changes I made.

# Pickling Data

Pickling is used for storing more complex data. The disadvantage of it is that you cannot simply open up a text editor and manipulate the data/code like you can in a text editor. To save data in the pickle module, you can use the keyword *dump* to “dump” binary data and to read pickled data you use the keyword *load*. I found it much easier to use the pickle module to store data as I didn’t have to worry about formatting in and out of storage.

While researching pickling via Google, I found the most helpful references to be on YouTube where it was much easier to gain context on how the module was being used. I also found Stack Overflow to be helpful when putting together my program. And I actually didn’t come across any references that were poor – some don’t exactly answer my question but that is more due to specifics than quality of reference. Overall, I have been impressed with the abundance of information on python on the web.

YouTube: [Using Pickle to write to and read from a file in Python [Part 3]](https://www.youtube.com/watch?v=Bzt6vlf865k) -- This was helpful to walkthrough a very simple example of pickling a dictionary.

Stackoverflow: [EOFError: Ran out of input](https://stackoverflow.com/questions/24791987/why-do-i-get-pickle-eoferror-ran-out-of-input-reading-an-empty-file) -- I simply google’d this error message and found a stack overflow answer that I was able to use within minutes. This helped me handle the error instance when you try to pickle an empty file.

# Exception Handling

One thing I have learned throughout this course is that programs will crash. To help negate the frequency in which they crash, it is helpful to trap for specific exceptions. Any time you have user input or you’re turning over some aspect of control of your program (e.g. another programmer is going to be using it) it is likely that someone will use it differently than what it was built for. Thus, it is helpful to think through ways it can crash and then trap for specific exception types to let the user know what went wrong.

I found [this documentation](https://docs.python.org/3/tutorial/errors.html) at python.org to be very helpful in understanding how to handle exceptions. It also walks you through how to write programs that handle custom exceptions. Additionally, it has a link to all of the built-in exceptions within python.

# The CD Inventory Program: with pickled data and added exception handling

I revised the CD Inventory Program to store binary data as well as added exception handling in areas where I was able to crash my program. Below I discuss my main revisions and then show snapshots of my program running. Note that there were some instances where I intentionally allowed my while loop to handle input errors and thus I did not revise the code for those -- for example, if you select anything but the listed menu options, the while loop continues to run until you make a valid selection. Thus, I did not build error handling for this. That said, I made the following revisions:

* Used the pickle module and revised my read\_file and write\_file functions to work with files that stored binary data
* Error handling to make sure the user inputted an integer when selecting a CD to delete
* A check to see if the inputted CD id already exists in the file. This also ensures it is not confusing when you go to delete a CD id listed twice
* Error handling if the user attempts to load a file that is not there
* Error handling if the user attempts to load a blank file

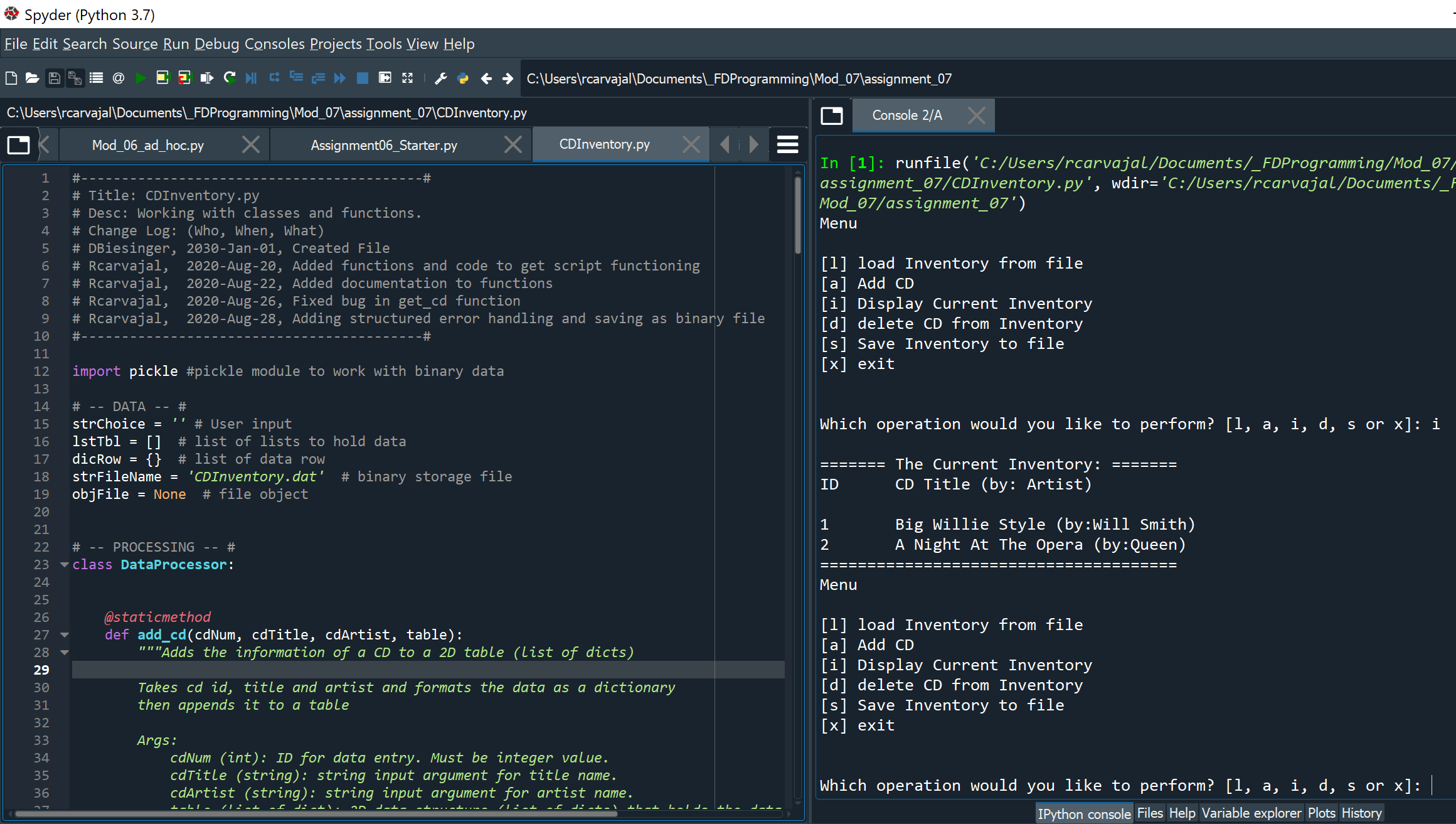
Below are snapshots of the program running – the first two show different parts of the program running in Spyder and the last one shows the program running in the terminal window. 

Figure 1 - Displaying contents from my CD Inventory DAT file

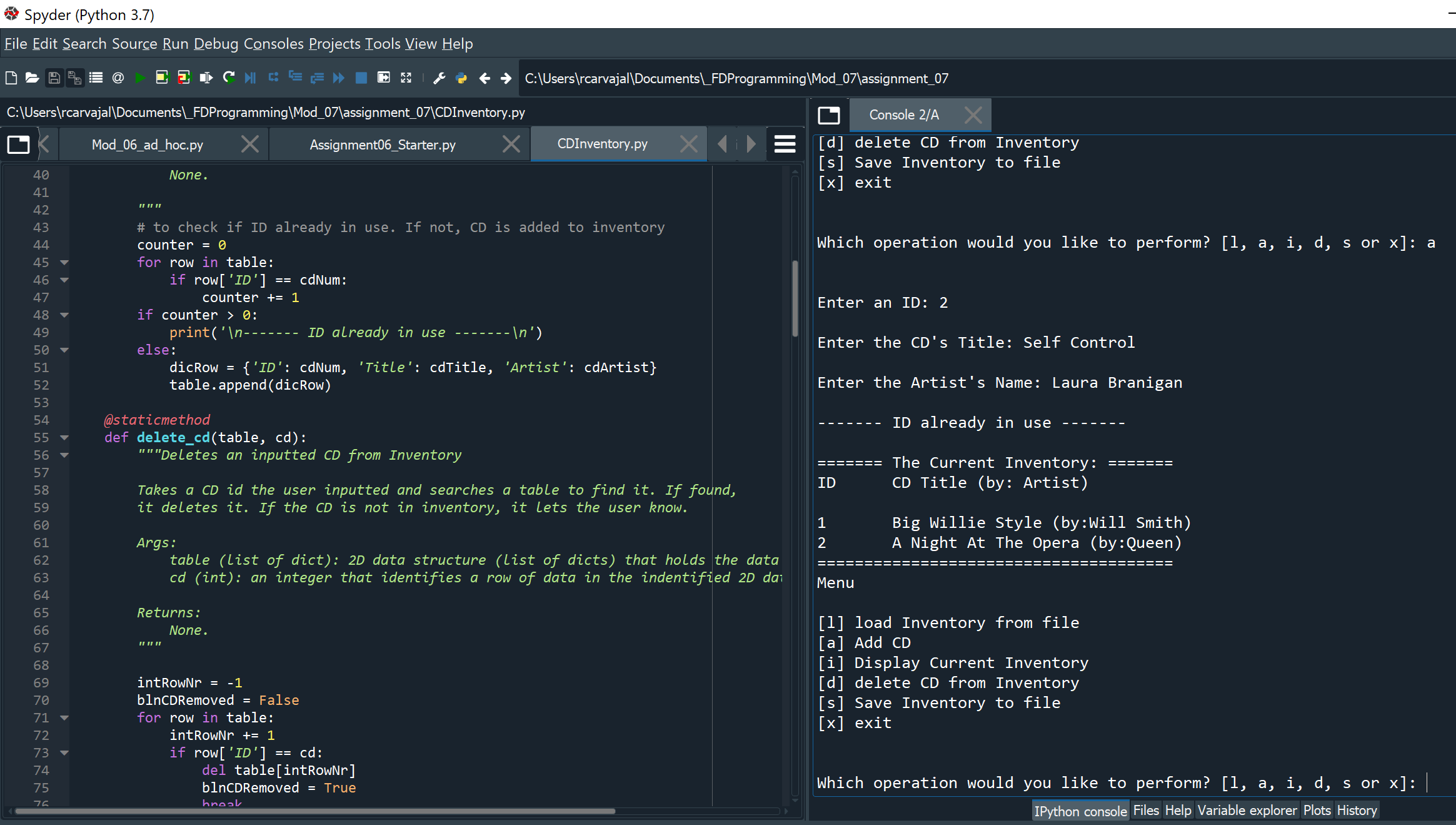
The snapshot above shows the program loading the DAT file and the one below shoes the program letting the user know the ID is already in use and the user can see the CD was not added to inventory. 

Figure 2 - Trying but failing to add a CD with an already listed CD id

The snapshot below from the terminal window shows the program not being able to find the file because there was no copy in the base directory. I then continue to add one CD

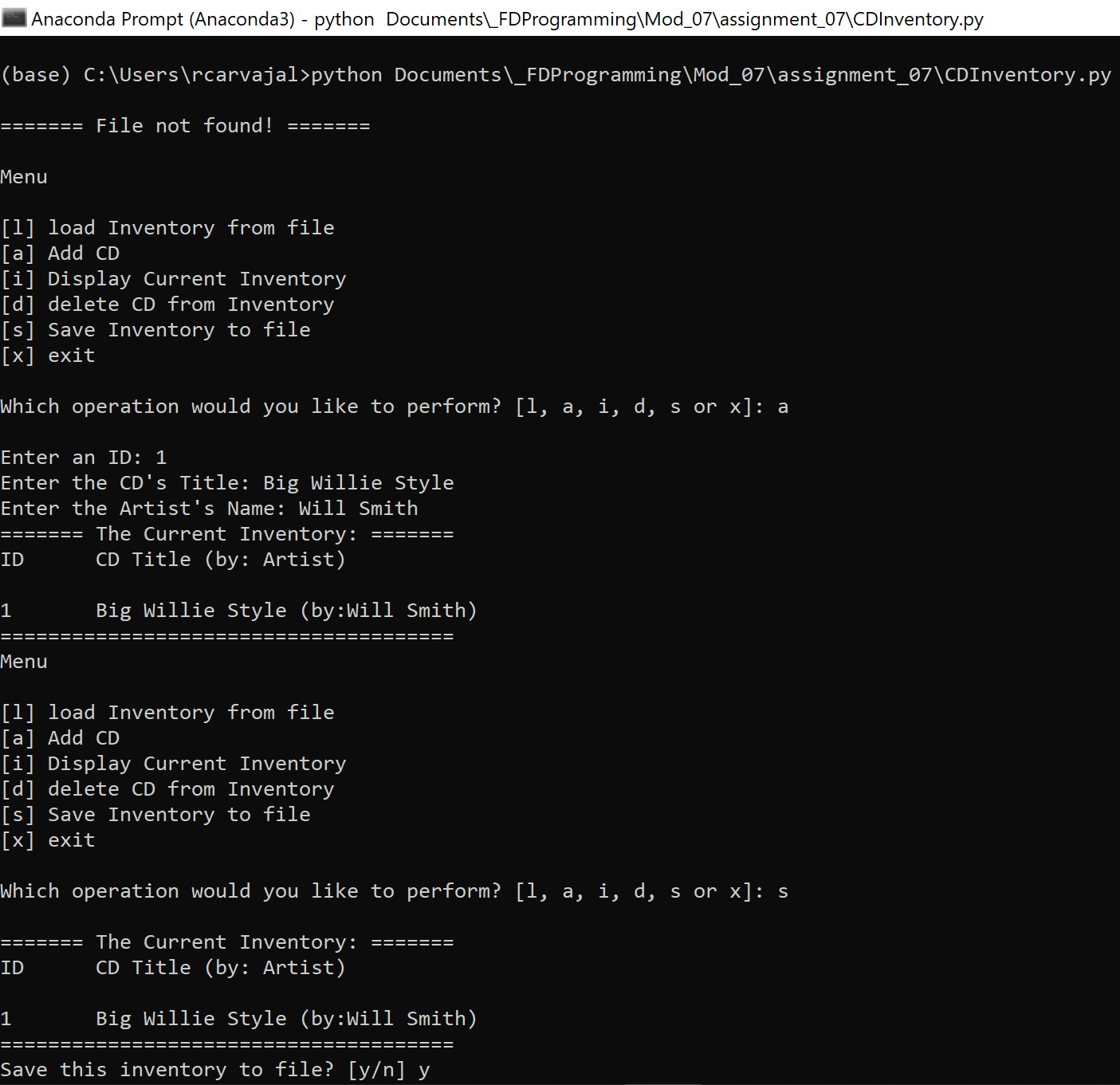


Figure 3 - File not found but program continues to run

# Summary

This week we focused on reading and writing text from files. We learned how to read lines and characters from text files and then also learned how to store more complex data. Pickling or shelving data can be an easier and more efficient way to store data though it doesn’t come with the advantage of modifying the data in a text file. We then went a bit deeper with handling errors, covering how to trap for the different errors that can arise in python. Finally, this doc ends with revising the CD Inventory program with these learnings.