Name: Salvador Gavonel

Date: December 04, 2021

Course: IT FDN 110 B

Assignment 08

CD Inventory – Class Objects Script

**Introduction:**

Module 08 introduces the concept of object-oriented programming and the class definition. We look into the structure of what a class is and how we can use it to model real world behaviors or to use as a replacement for a dictionary or a list.

Assignment 08 challenges us to create a CD:Inventory script using objects instead of dictionaries to store our CD entries. The following paragraphs summarize how I managed to introduce them into the script and any insights I had during the exercise.

**Opening the Assignment08\_Starter File:**

The starter file had a lot less code than I thought I would encounter, and more pseudocode to replace it. After reading through it, I realized I could just recycle most of the code from Assignemnt06 to perform the tasks for the TODO.

**Creating the CD Class and defining Properties:**

The first thing that I started with was with the creation of the attributes that the object would have; since it was fresh in my mind, I used the Lab exercises to create properties for ID, Title, and Artist. I did not add any specific format for the new attributes other than the .title() method, but for the ID I did make a check to make sure it was an integer, else it would raise an exception. After trying the code out, I realized that raising an exception breaks the code so I decided to just print some text stating that the ID needed to be an integer and then return to the main menu.

Later in the development of the code and after the QA session I decided to add methods for the presentation of the objects. The first ones I created was the \_\_str\_\_ method and a \_\_repr\_\_ method to make sure the objects would have a readable format if I wanted to print the list or an object at any point of the debugging process. The last method I created was writetofile() to format the object when writing it into a file.

**Recycling Past Code:**

**FileIO Class:**

For the file processing code I used the code I used in Assignment06. The first difference/hint I noticed was that the load\_inventory only had the file name as the parameter; the method would then have to return a list of objects. I then resorted to create a table variable in the method and appended each entry to it as an object, and set it to return it. I also added some exception handling in case there was no file or if the file was not read correctly.

For the save\_inventory method I was able to make the code much simpler by using the writetofile method I created earlier in the CD class.

**IO Class:**

I was able to copy most of the code form Assignment06 as well here. For the print\_menu and menu\_choice I could just use the same code, I just removed the option to delete an entry since the pseudocode did not ask for that option.

For the show inventory method, I used a simple for loop and for each object in the list I would print the object itself. Since I had defined the \_\_repr\_\_ method in the CD class it would print out the object in the correct format.

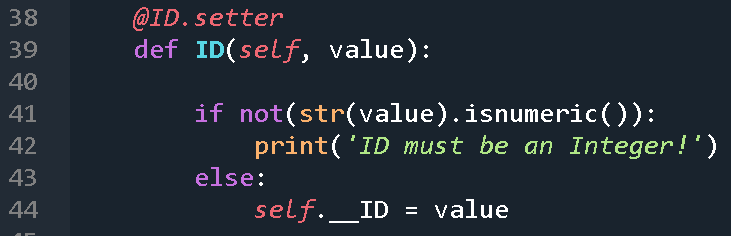
For the CD\_Choice\_Add I had to think a little harder here; in the previous assignments I had used all inputs to create a dictionary and return it but here I did not have a variable to hold the object. Also I had used an add\_item method in my previous code to add the new entry to the list, which looks like I did not neet after looking at the pseudocode. I had some difficulties trying to understand how I could accomplish this, but I was able to have the CD\_Choice\_Add method return a CD object, which I could then just append to the existing list.

**Exception Handling:**

For the exception handling I looked at the same error occurrences that could have happened in the last assignment: File processing and data entry.

For the first, I was able to have error handling added to the load\_inventory in case the file name did not exist in the directory, and if the format in the file was not correct (comma separated) then it would shoot another exception. The code would still continue and show the menu back to the user.

The second instance to add exception handling was in case the ID entered was not an integer. In the previous assignment I had added exception handling on both the main body and in the add\_CD function, but since the property setter checks that the input is valid I could now get rid of them.



Listing - ID.Setter Exception Handling

**Running Code:**

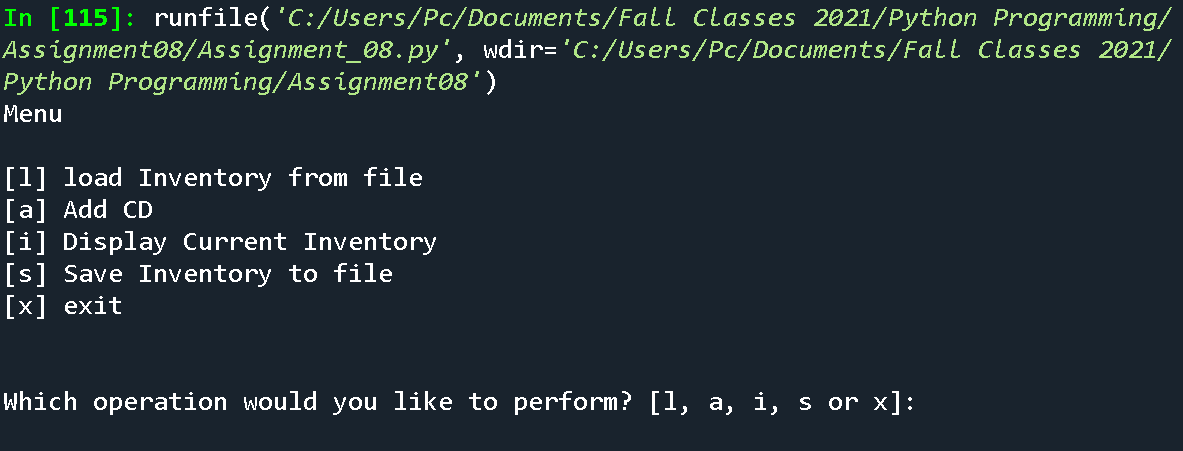


Figure - Running Menu and Loading File (Spyder)

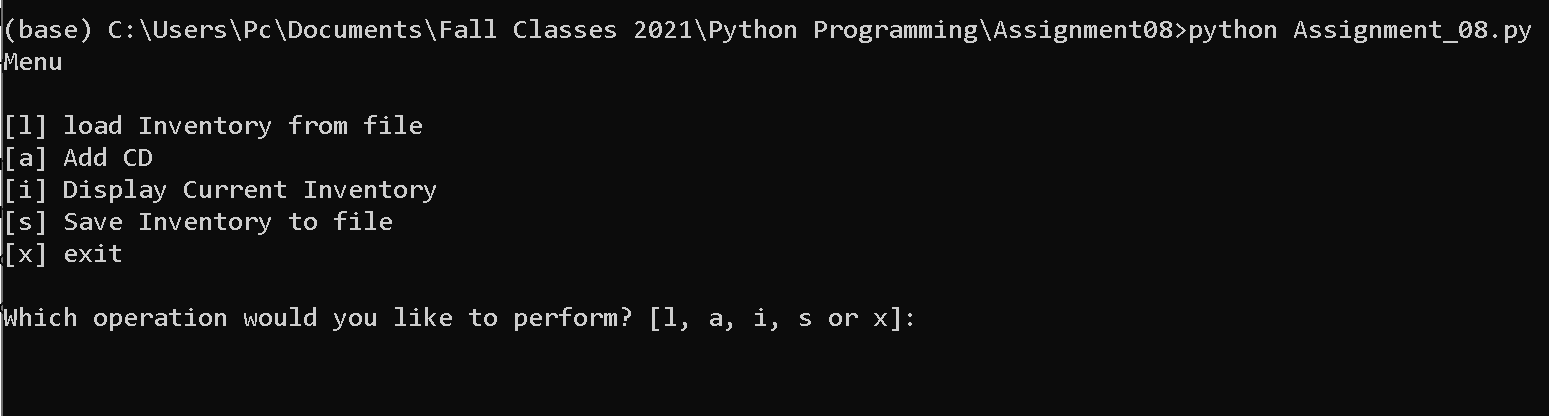


Figure 2 - Running Menu and Loading File(Terminal)

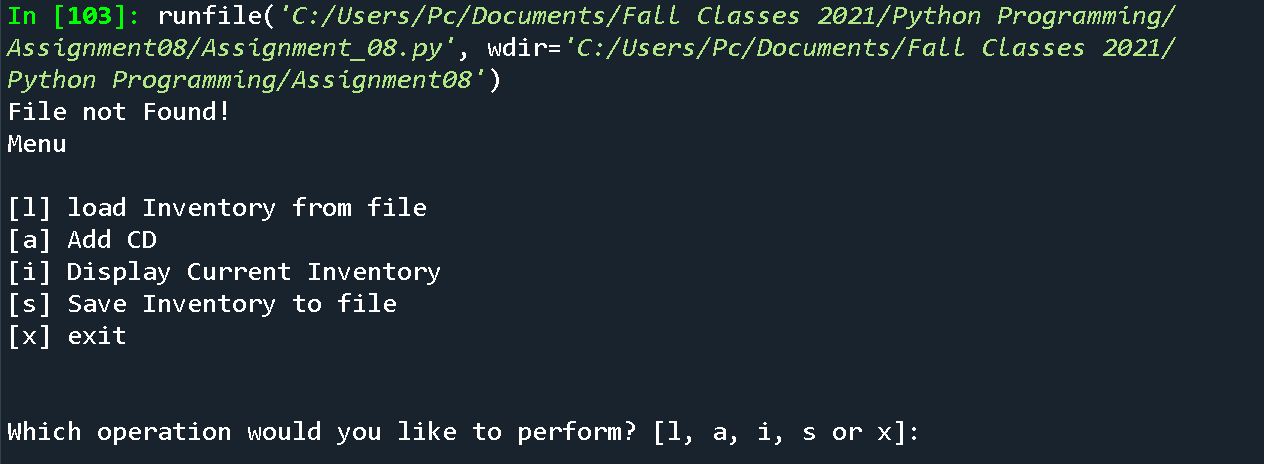


Figure – Error Handling 1 – File not found (Spyder)

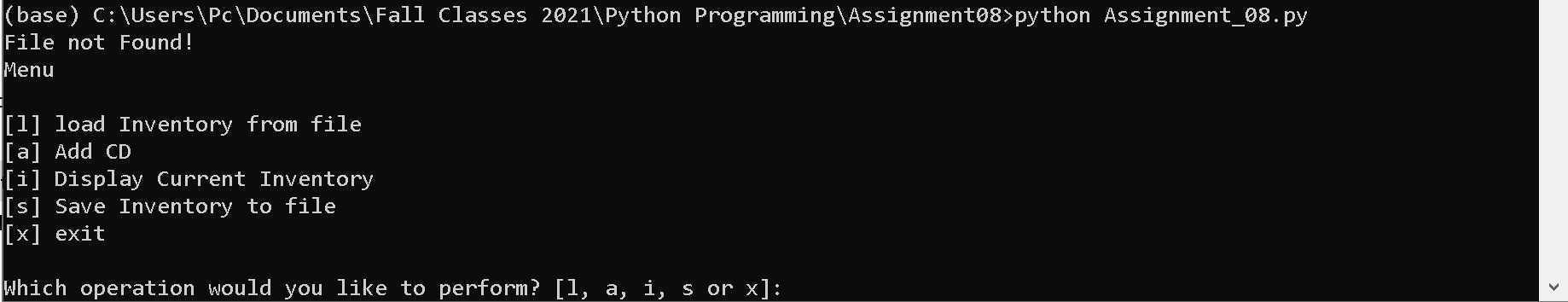


Figure 4 - Error Handling 1 – File not found (Terminal)

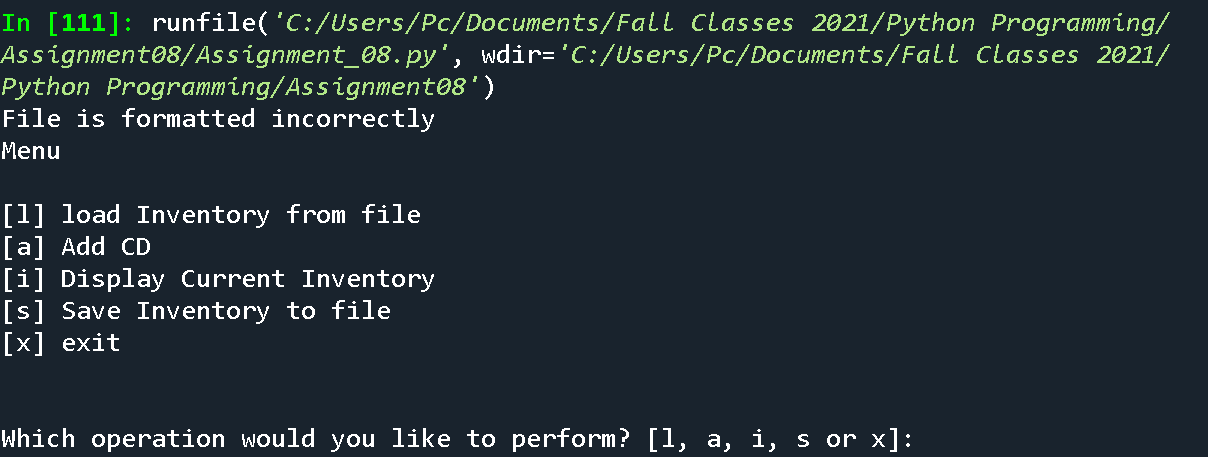


Figure – Error Handling 2 – Invalid File Format when loading file (Spyder)

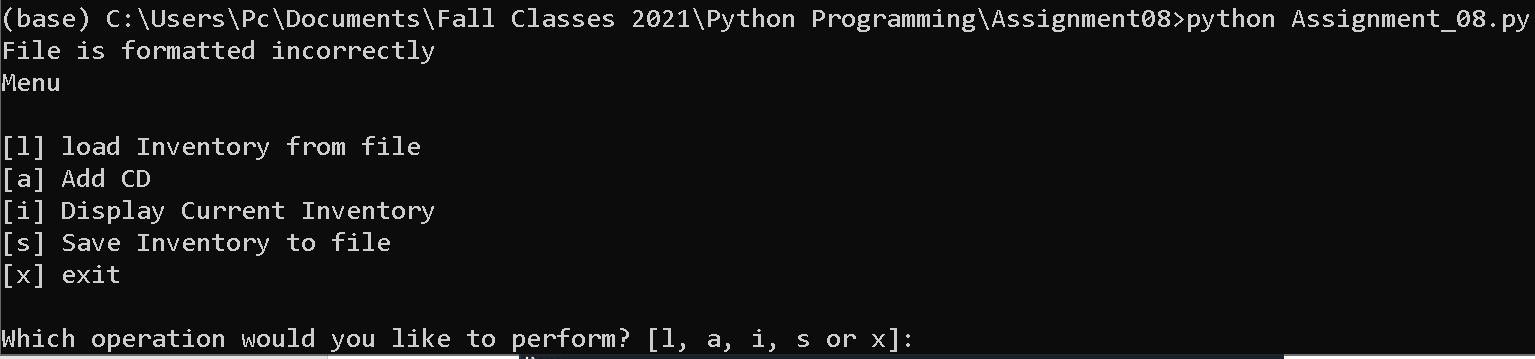


Figure - Error Handling 2 – Invalid File Format when loading file (Terminal)

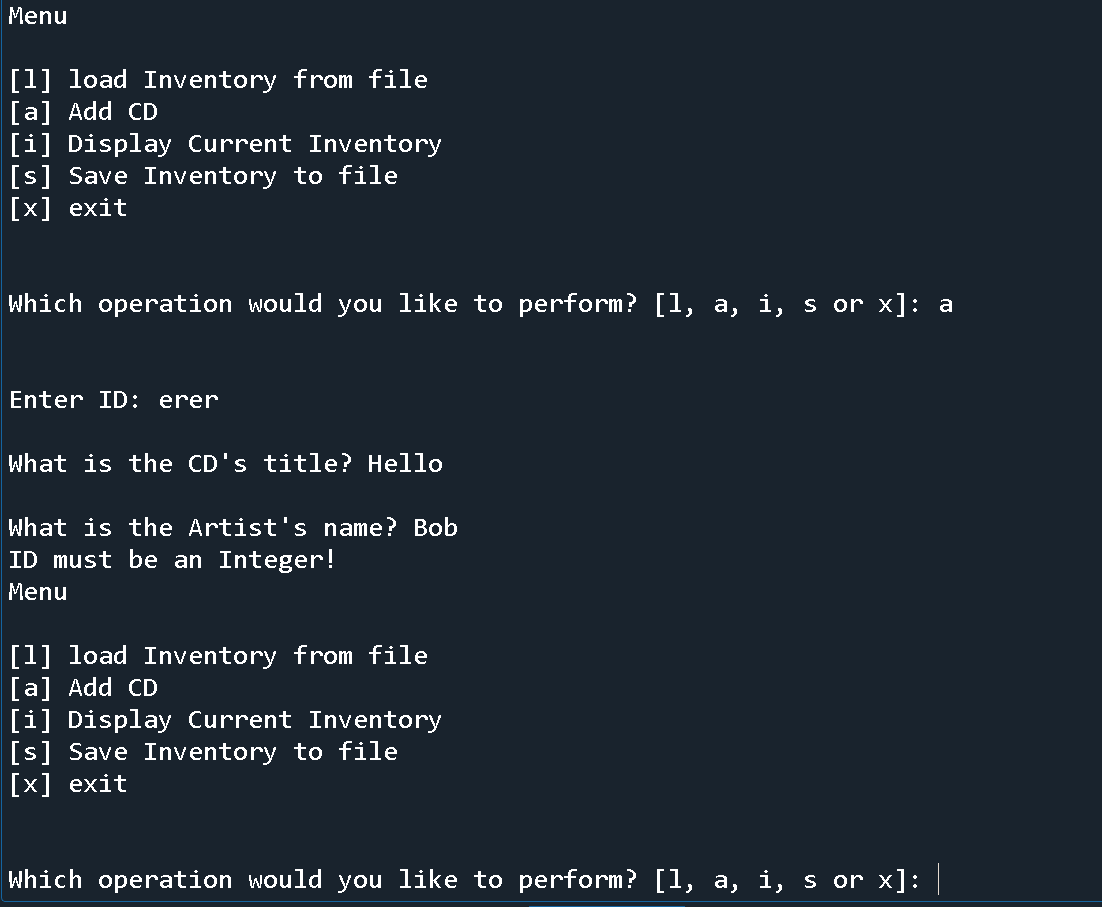


Figure - Error Handling 3 – Invalid ID when adding new entry (Spyder)

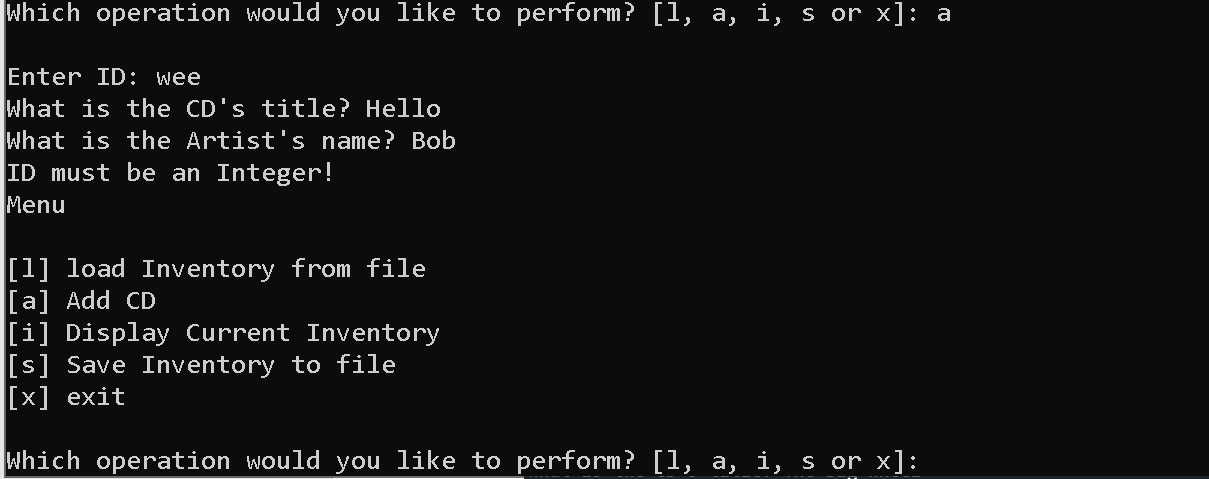


Figure - Error Handling 3 – Invalid ID when adding new entry (Terminal)

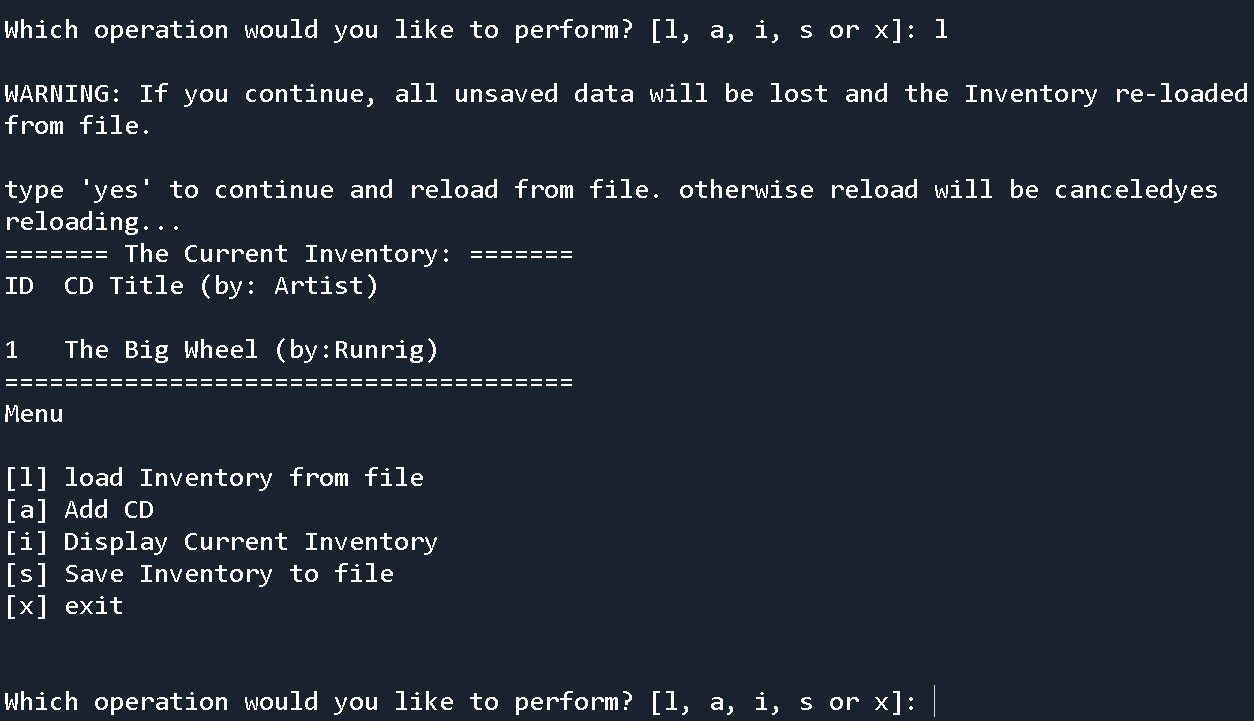


Figure - Loading File (Spyder)

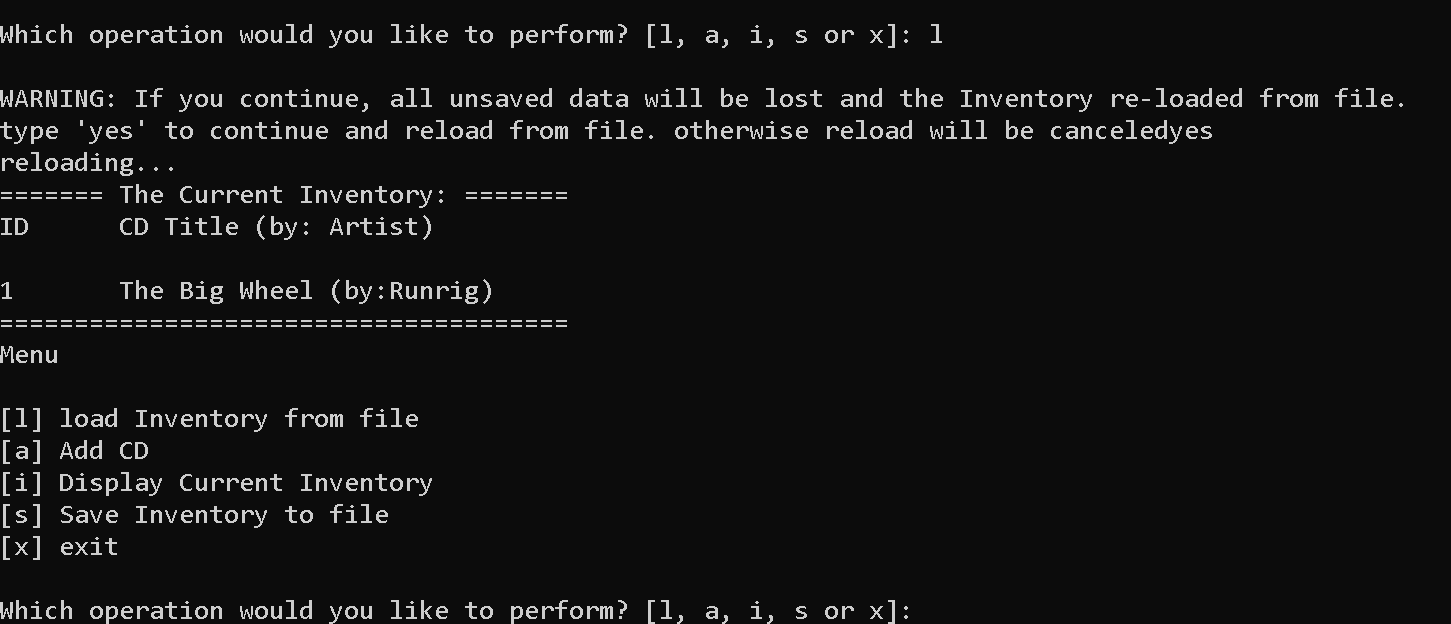


Figure - Loading File (Terminal)

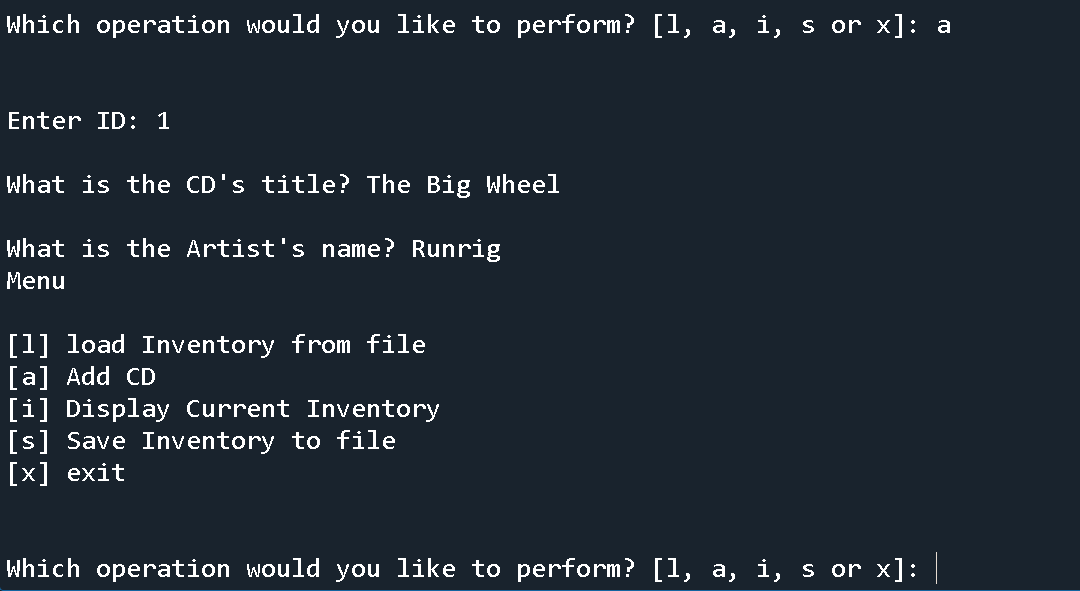


Figure - Adding new CD (Spyder)

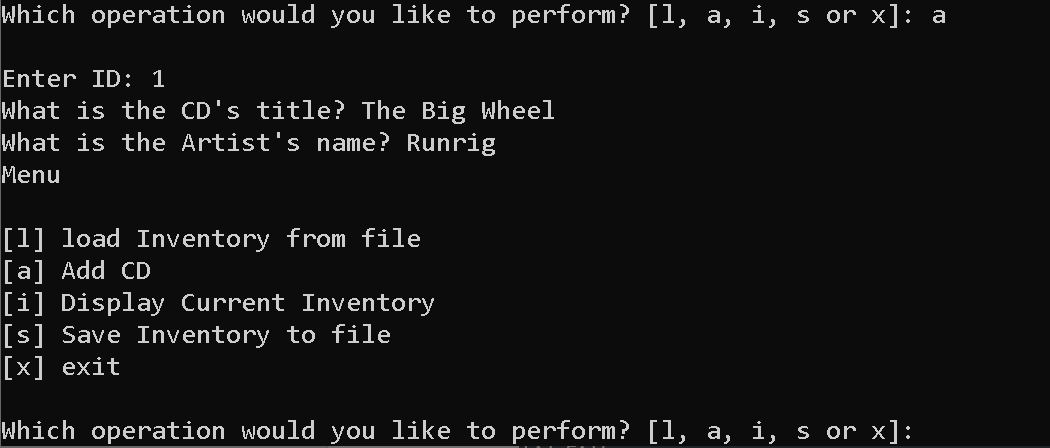


Figure - Adding new CD (Terminal)

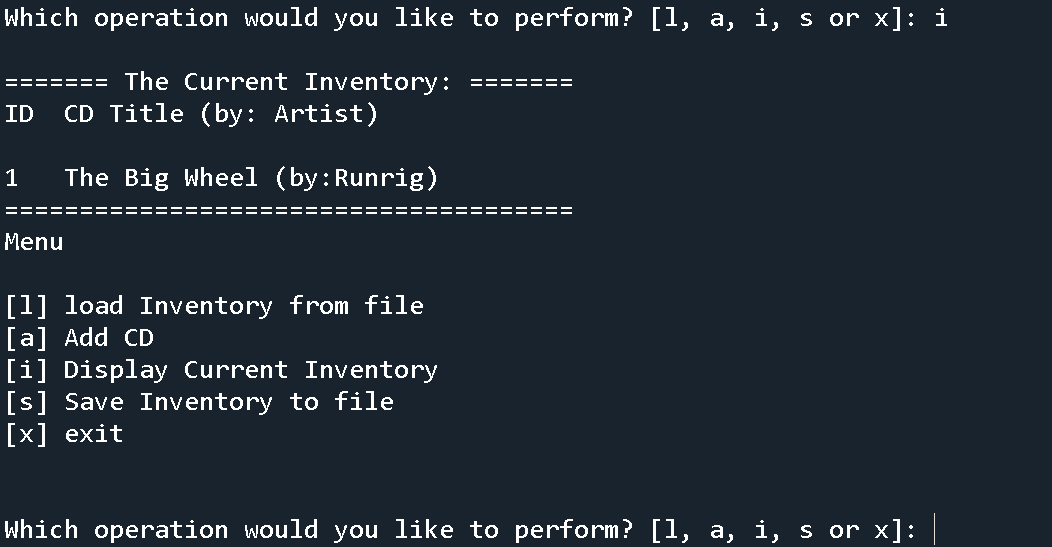


Figure - Display Inventory (Spyder)

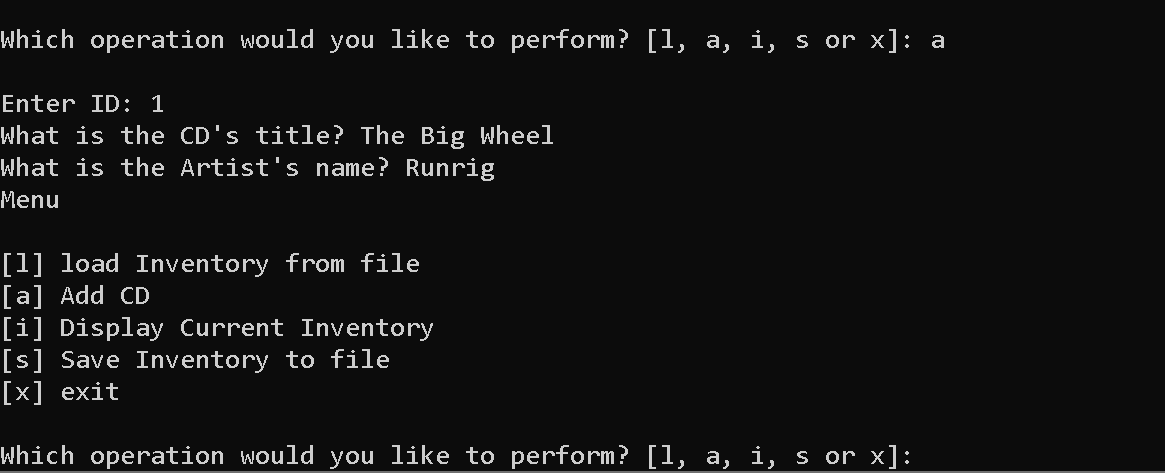


Figure - Display Inventory (Terminal)

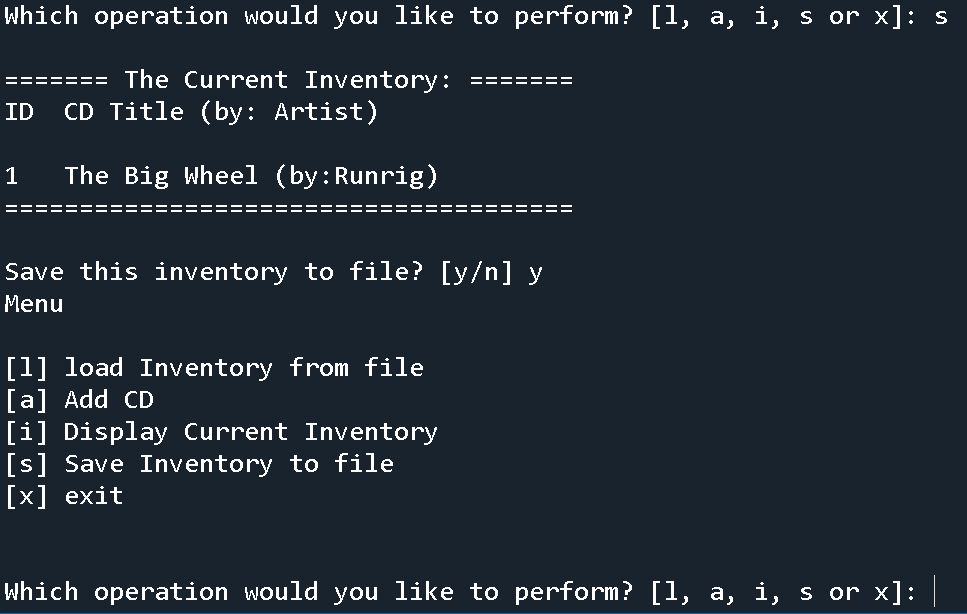


Figure - Save to File (Spyder)

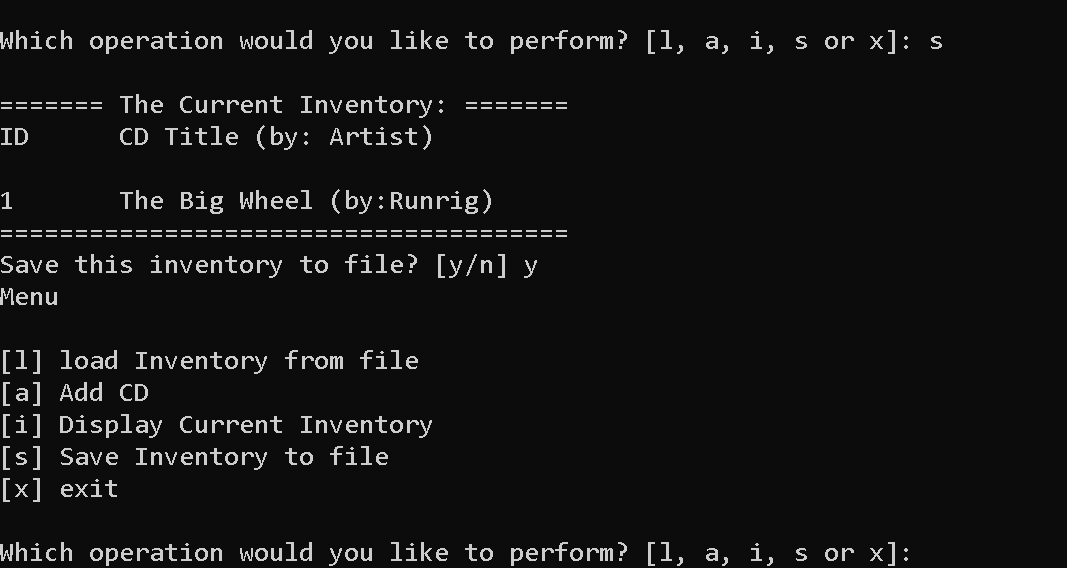


Figure - Save to File (Terminal)

**GitHub Repository Link:**  https://github.com/Sgavon/Assignment08

**Summary:**

The introduction of classes adds the ability to create objects in python that have its own attributes which allows users to store information in a more efficient manner. The introduction of object-oriented programming gives python a huge potential when creating objects that can mimic real world objects, it sounds very interesting. The assignment is a great way to tie in OOP concepts with exercises that we have been working with.