John Stevens

March 19, 2020

IT FDN 100 B

Assignment 08

CD Inventory(w/Classes and Objects) Script

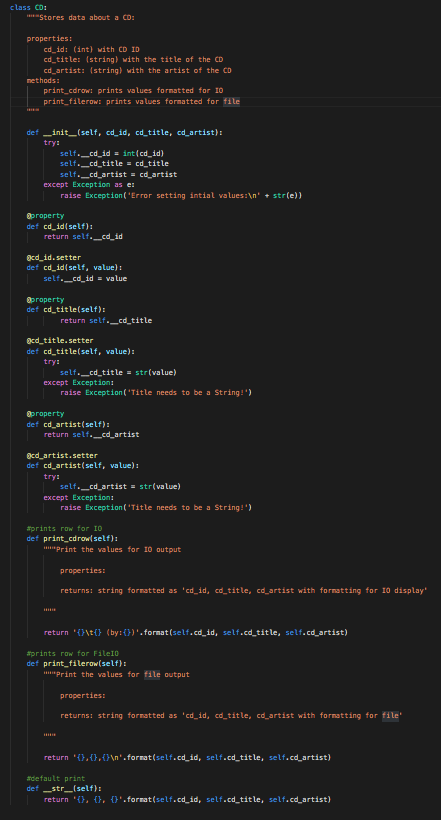
# Introduction

This document will show the modification of a pseudocoded script in Python to add the following functionality, ask a user to input an inventory of CDs, display the inventory, load the inventory from a file, and then save the inventory list to a file. The script also uses structured error handling to capture error to provide a more stable environment and Object oriented Classes.

# Script Creation

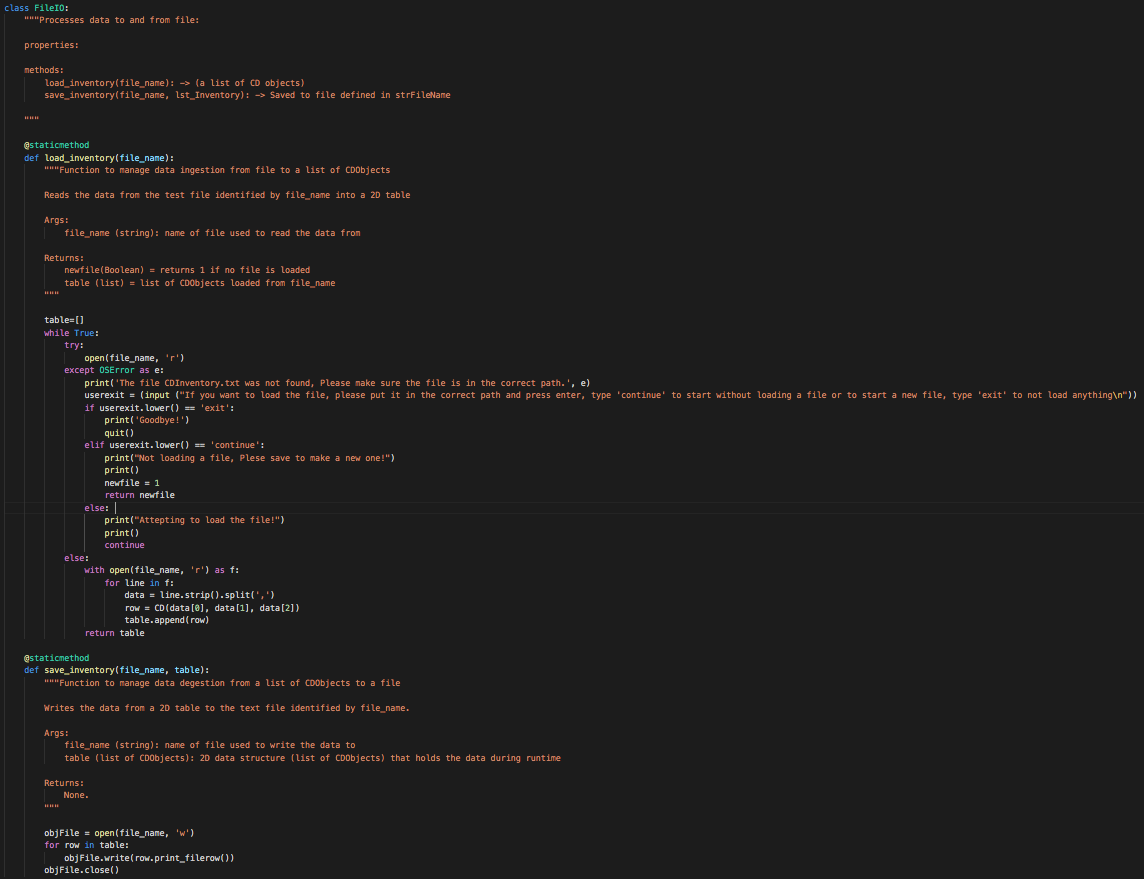
I will cover each class in the script, the first if these is the CD class, which is using an object oriented approach (PYTHON PROGRAMMING FOR THE ABSOLUTE BEGINNER 3RD ADDITION, MICHAEL DAWSON, CENGAGE LEARNING 2010, P. 217-246) .

The CD class,



The CD class begins with a constructor def \_\_init\_\_(self, cd\_id, cd\_title, cd\_artist), here were are defining what variable to be use in the class then try setting them to private attributes for each object. Next we set up properties for each of the private attributes we set before to provide indirect access. Then using the .setter attribute we can set the value of the private attributes we just got access to. Next we set some methods for displaying the values of the attributes for each object in different formats for printing, print\_cdrow returns the values for formatted for IO display, print\_filerow returns the values formatted for file storage and \_\_str\_\_ is the default format for the class.

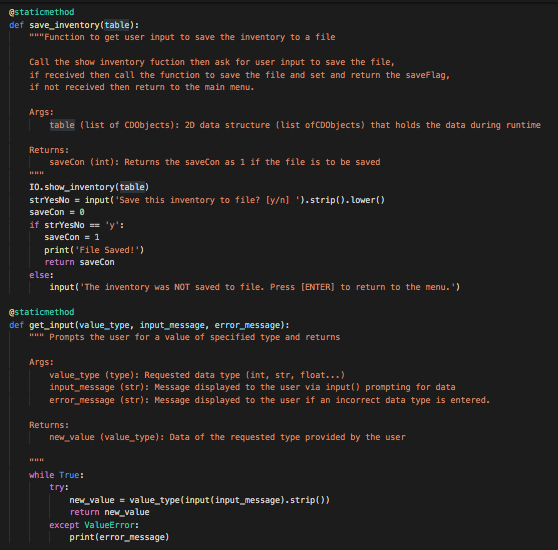
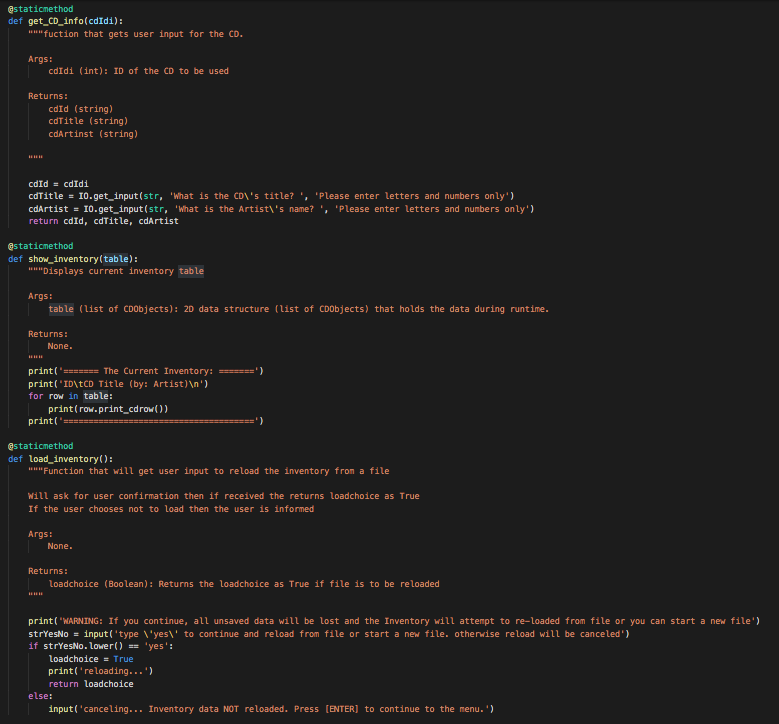
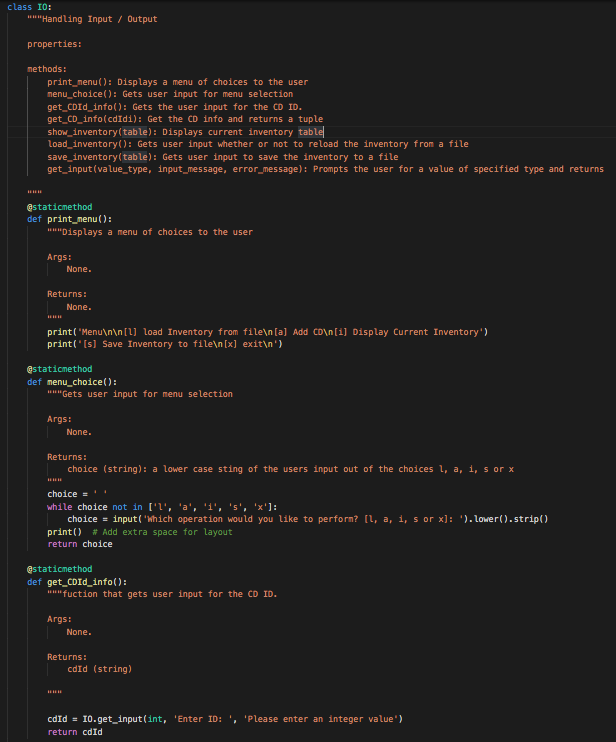
Next is the FileIO class,



Here the FileIO class has two @staticmethods https://realpython.com/instance-class-and-static-methods-demystified/[[1]](#footnote-1)1. The first is load\_inventory and takes a file\_name variable and attempt to load that file and return the list of CDobjects. If the file is not found then the script asks if the user wants to continue without loading or to exit the script.

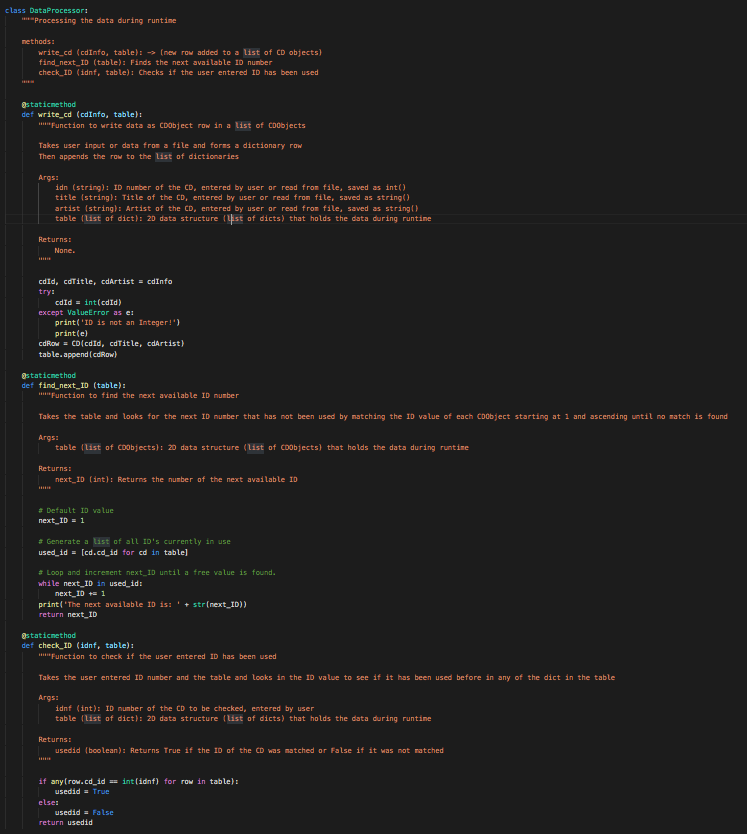
The next save\_inventory takes the list of CDObjects and writes the list to a file using the cd.print\_filerow() method we defined in the CD class.

The next class is IO­,



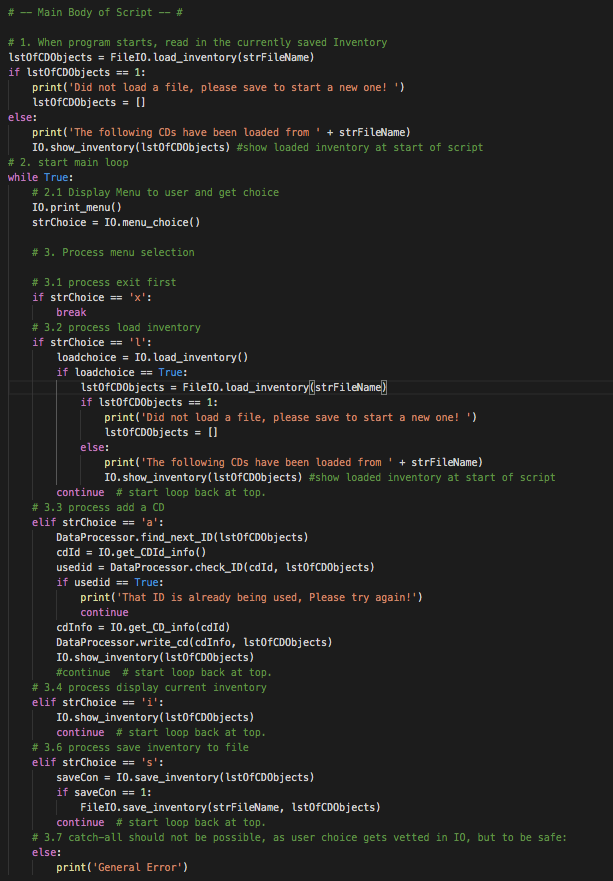
The class IO has 8 @staticmethods, The first two print\_menu() and menu\_choice() display the menu and handle the user input of their choice. The get\_CDId\_info function returns the ID of the cd to be used and then is used in get\_CD\_info tat retries the other user input for the cd and returns a tuple. The show\_inventory method displays the current inventory in the list of CDObjects using the cd.print\_cdrow() method in the CD class. While load\_inventory and save\_inventory both ask for user confirmation before loading/saving the file of CDs. The get\_input method takes the uses input it is passed and check that the input matches the type or an exception is raised.

The final class is DataProcessor,­



The DataProcessor class has three @staticmethods, the write\_cd class takes the tuple of the entered cdinfo and using the CD class forms a CDObject and appends that to the list of CDObjects. The find\_next\_ID method looks in the list of CDObjects and returns the next unused available ID starting at 1. Check\_ID looks to see if the user entered ID has been used and returns the result.

The main body of the script,



starts by calling the FileIO.load\_inventory() with the variables “strFileName” and “lstOfCDObject” to load and display the inventory from the file, if the file is not found then the user choose to exit or continue with no file loaded. Then the IO.print\_menu is displayed and user input is requested with the IO.Menu\_choice function. The users input is stored into “strChoice”. Then depending on the users input the script will do the following, If the user enters ‘x’ the script breaks. If ‘l’ is entered then the function IO.load\_inventory is called, if it returns Ture then FileIO. load\_inventory is called and if it returns 1 then “lstOfCDObject” is reset. Else the inventor is loaded into “lstOfCDObject”. If ‘a’ is entered then then DataProcessor.find\_next\_ID() displays the next available ID and then the user is asked to the CDID that they want to use and that is used in the DataProcessor.check\_ID method. If that ID had been use then the script tells the user that and returns to the main menu. If not then the user is asked for more cdinfo with IO.get\_CD\_info and then that is used in DataProcessor.write\_cd to write the cd to the table and IO.show\_inventory displays the inventory to the user. If ‘I;’ is entered then IO.show\_inventory displays the inventory to the user. And if ‘s’ is entered then IO.Save\_inventory is called, if 1 is returned FileIO.save\_inventory is called and saves the list of CDObjects to the file.

# Performing the Script

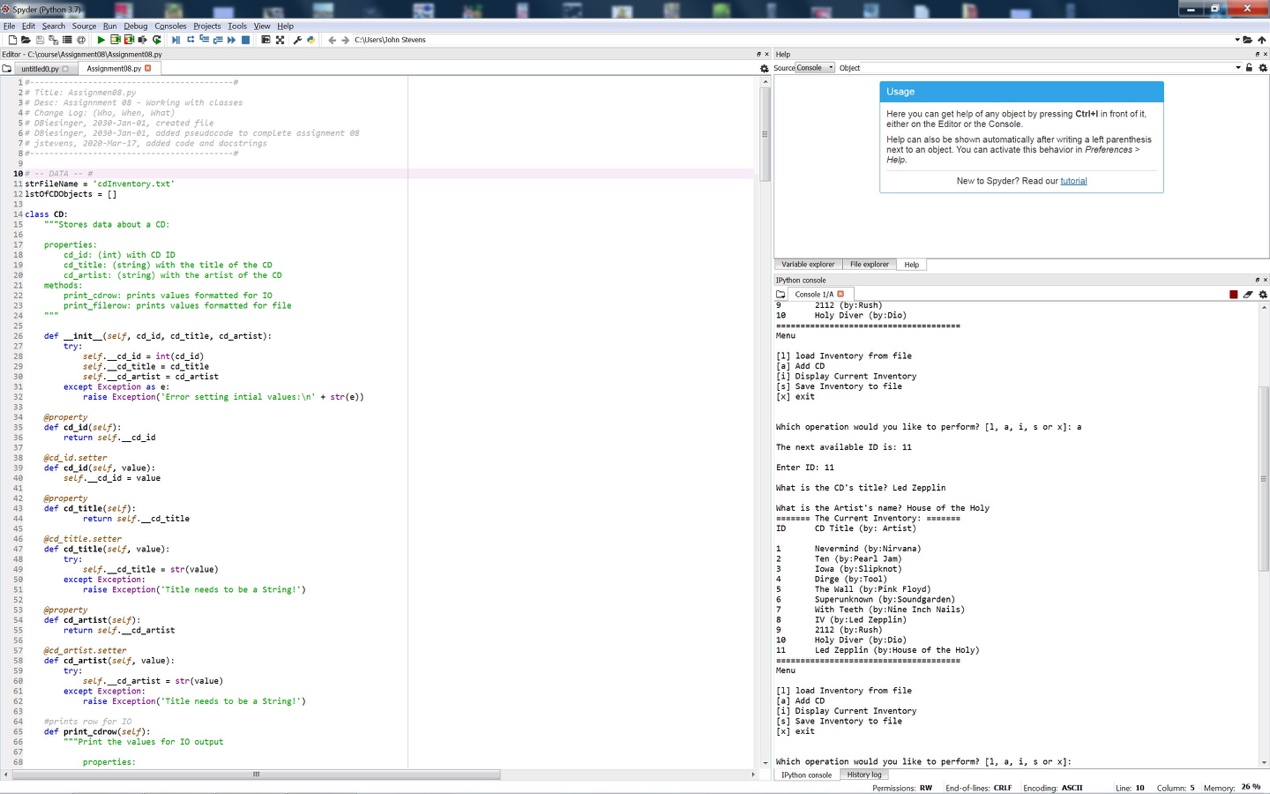


Figure - Script performing in Spyder.

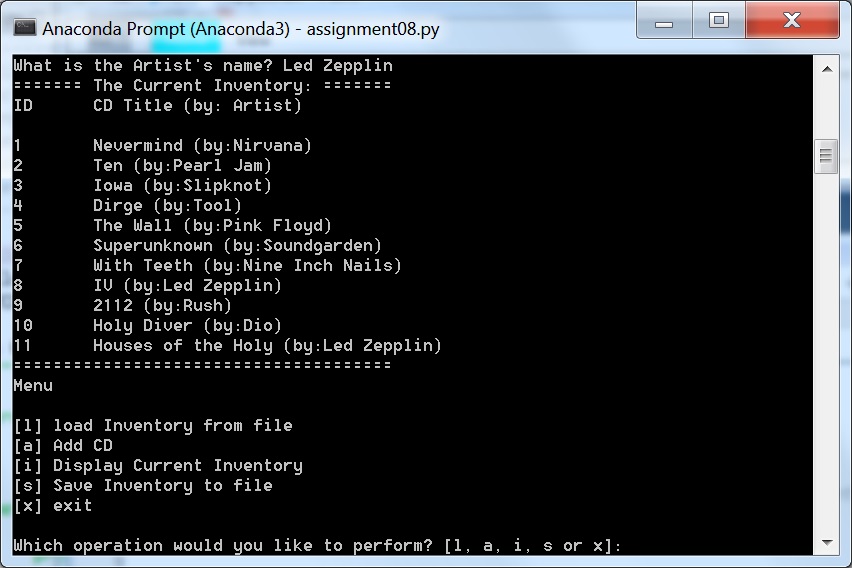


Figure 3 - Script performing in terminal

As seen in Figure 2 and Figure 3 I ran the script in Spyder and Terminal respectively. I loaded 10 test CDs from the file CDInventory.dat, then added 11th CD, displayed the list and saved the list to a file.

# Summary

In this assignment I was able to modify an pseudocoded script to loading and saving data to a file and add in structured error handling to provide expanded functionality to load, add, show and save CDs to a list and added andobject class for the CD info. This script and document and posted at https://github.com/pjfan73/Assignment08

1. 1 [↑](#footnote-ref-1)