Introduction to Object Oriented Programing

# Introduction

The goal this week was to get a brief introduction to Object Oriented Programing and how to apply it. This constituted of a

# Creating the CD\_Inventory Script

This week, I was given a shell script that had an outline of the task to accomplish. The program I was tasked with creating was the same CD Inventory program as had been made in previous weeks. The difference this week was there was no functionality programmed in yet, and the objective was to build the program following the principles laid out in OOP, while still applying the principles learned in the previous weeks.

The shell script provided three pre-defined classes:

‘CD’ – The class containing fields, attributes, properties, and methods related to the storage of CD information.

‘FileIO’ – This class was created to contain any functions used in the processing of data to or from a file.

‘IO’ – This class was created to hold any functions related to collecting user input, or printing any statements, processing any Input/Output data.

The last part of the outline directed creating a main python script to access the aforementioned functions to create the CD handling program

The first step I accomplished was to define and update all docu-strings I needed for functions I planned on using. This allowed me to better flesh out the types properties and functionality I wanted from each class, and thus create a more cohesive object.

After determining the rough outline, I focused on the ‘CD’ class. This class holds all the properties and methods related to the storage of CD information in the program. Since all the particular attributes related to each CD are user defined, I didn’t have any class attributes to put in the field section. In the ‘constructor’ segment, I used the ‘\_\_init\_\_()’ method and created the following attributes: ‘cd\_id’, ‘cd\_title’, and ‘cd\_artist’. I made these private variables that cannot be called directly without instantiating the CD class and calling them through their instance method.

Once the variables were initialized, I created properties to allow the user to define and get each individual term, as they cannot be directly accessed. For these I applied the @settor and @properties decorators, respectively.

Now the variables were initialized, and had properties assigned, I created instance methods to process the data in the class. The primary method I created was the ‘cd\_added’ method. This method gathered the CD’s ID, Title, and Artist variables and stored them in a dictionary. I appended this dictionary to a list created to contain dictionaries. Lastly, I utilized the lambda function to sort the dictionaries to display the CDs in a logical numerical order.

The last method created was a dunder method using the ‘\_\_str\_\_()’ function. I used thismehtod.

The creation of the FileIO and IO class were quite similar to what I did in the previous week’s modules. This was a relatively straightforward process, as each of these classes contained only static methods. This means I did not have to use a constructor to initialize any variables, create attributes or properties.

The main body of the script was relatively simple as well, as the functionality of the program was largely unaltered. This allowed the same use of an if-elif-else framework to allow the user to navigate the python program.

Throughout my program, I applied structured error handling for the FileNotFoundError, and the ValueError, as these were the most commonly encountered errors in my program that I was able to identify.

# Running Script in Spyder IDE

The figures below show the result of running the ‘CDinventory.py’ script in Spyder IDE.

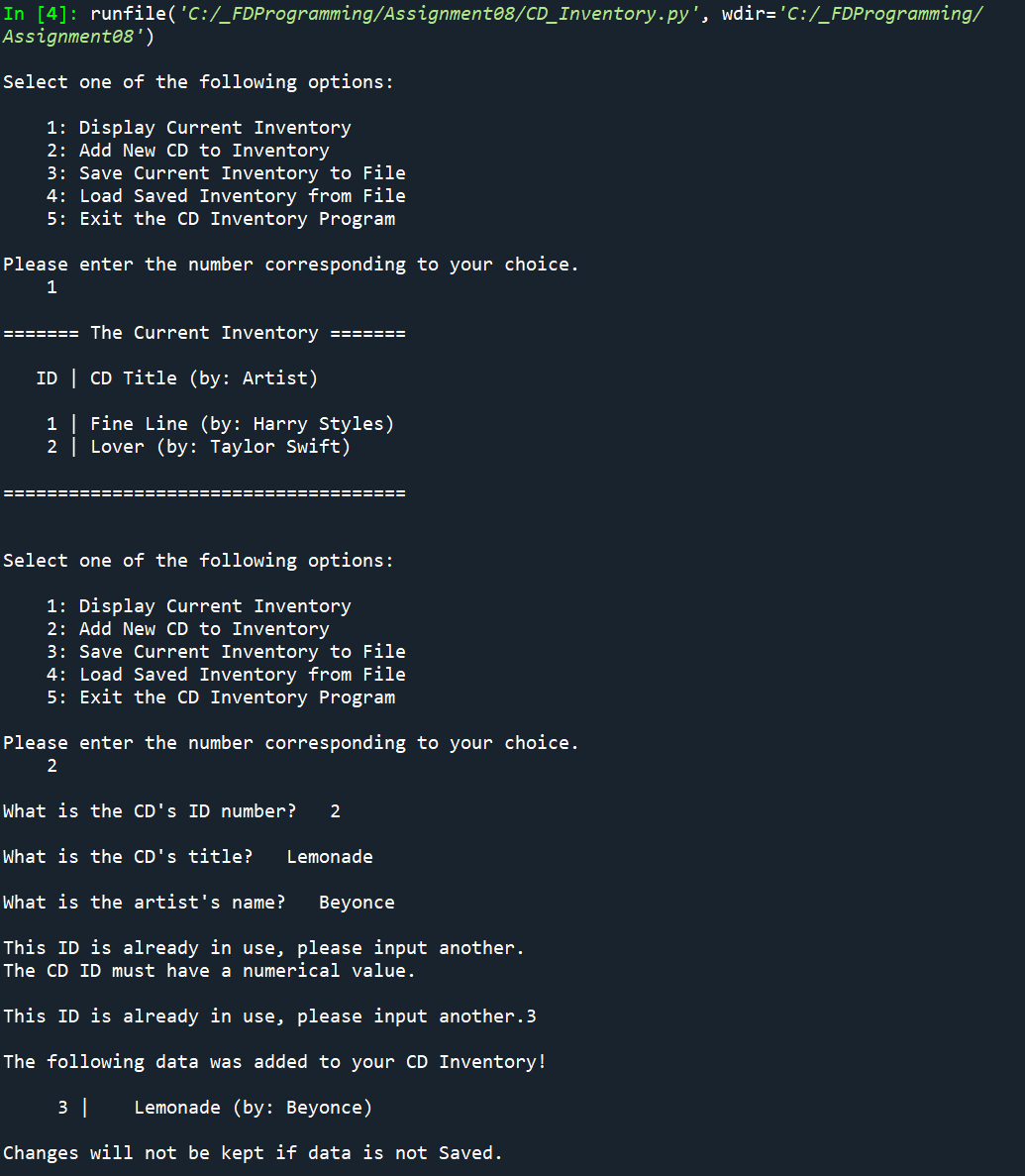


Figure - CDinventory.py program in Spyder

# Running Script in Terminal Window

The following figures are the result of running ‘CDInventory.py’ script in the terminal window.

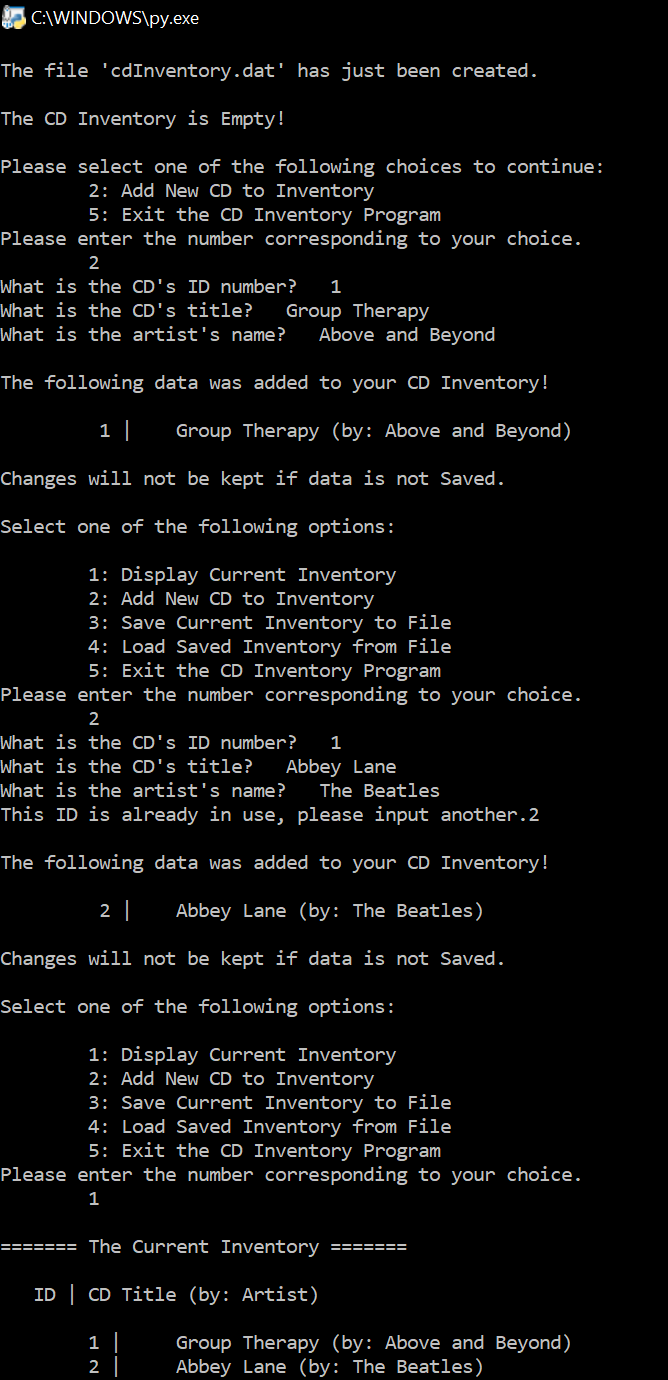


Figure 2 - CDInventory.py Program run in Terminal Window

# Github Link

Here is the Github link to my code:

<https://github.com/mxlarkin/Assignment08/>

# Summary

After the completion of this assignment I have a better understanding of what classes are and how to implement Object Oriented Programing. I have to do more experimenting to get a better understanding of this concept. For now, I only have very rudimentary knowledge.