

Benjamin Reinhart

February 21, 2021

IT FDN 110 A

Assignment 06

Moving CD Inventory Python Script to Functions

Introduction

In this assignment I will cover how to modify a script in Python that manages a CD Inventory. I will modify the script to use classes separated by data processing, file processing and I/O and move the functionality of adding a CD, deleting a CD, and saving a CD to a file to new functions within those classes. I will use the information learned in Module 06 about functions in order to do this. Here is a link to the GitHub repository where this is saved:

https://github.com/reinhartben/Assignment_06.git

Pseudocode

We were provided the `CDInventory_Starter.py` file in order to get us started on this program. This was a version of the CD Inventory that had all the functionality we used in Assignment 05, but also included the Pseudocode for what we needed to modify and add in Assignment 06. The Pseudocode here, helped me define what exact actions I needed to take in order to update the script.

Modifying the Script

We want to update the script to move the functionality into functions. The starter code already has three classes defined, `DataProcessor`, `FileProcessor`, and `IO`. These classes allow us to group our functions based on functionality and better format our script¹. We will keep all input and output to the `IO` class, all interaction with files in the `FileProcessor` class, and all manipulation of data in the `DataProcessor` class.

The first functionality I moved was to delete an entry from the inventory. This is manipulating the data, so I created a function in the `DataProcessor` class called `del_CD`. This function takes in two parameters that allow us to pass in values for processing². The first parameter, `delRow` (integer), specifies which row needs to be deleted and the second parameter, `table` (list of

¹ Dirk Biesinger, Foundations of Programming (Python), Module 06 Page 21

² Biesinger, Foundations of Programming (Python), Module 06 Page 3

dictionaries), is the 2D data table that gets manipulated. With these parameters, we can move the code into this function.

```
41 def del_CD(delRow, table):
42     """Function to delete a row of data from the inventory
43     """
44     """Takes a user inputted row ID, delRow, to remove from 2D table (list of dicts), table.
45     """
46     Args:
47         delRow (integer): user inputted integer of CD ID to delete
48         table (list of dicts): 2D data structure (list of dicts) that hold the data during runtime
49     """
50     Returns:
51         None.
52     """
53     """
54     intRowNr = -1
55     blnCDRemoved = False
56     for row in table:
57         intRowNr += 1
58         if row['ID'] == delRow:
59             del table[intRowNr]
60             blnCDRemoved = True
61             break
62     if blnCDRemoved:
63         print('The CD was removed')
64     else:
65         print('Could not find this CD!')
```

Figure 1: del_CD function

The next modification was to move the save CD functionality to a function, write_file. Since this saves the data to a file, I put this in the FileProcessor class. The function takes in two parameters, file_name (string), that gives us the name of the file to save the data, and table (list of dictionaries) which is the table of data that we are saving. These parameters allowed us to move this functionality successfully.

```
93 @staticmethod
94 def write_file(file_name, table):
95     """Function to manage data saving from a list of dictionaries to a file
96     """
97     """Saves the data from a 2D table (list of dicts) identified by table.
98     Each dictionary row in table is saved as a line to the file identified by file_name.
99     """
100     Args:
101         file_name (string): name of file used to save the data to
102         table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
103     """
104     Returns:
105         None.
106     """
107     """
108     objFile = open(file_name, 'w')
109     for row in table:
110         lstValues = list(row.values())
111         lstValues[0] = str(lstValues[0])
112         objFile.write(','.join(lstValues) + '\n')
113     objFile.close()
```

Figure 2: write_file function

The final functionality we needed to move to a function was to add a cd. This takes in input on the new CD from the user and then adds this to our inventory table. This is input and data processing, so I know I need to make two functions, one in the IO class and the other in the DataProcessor class.

Starting with IO class function, `cd_input`. There are no parameters in this function, but it does return a list of the string inputs by the user. We return this string and save it in order to immediately use it in our next function³.

```
169 .....@staticmethod
170 .....def cd_input():
171 .....    """Gets user input for adding a CD to inventory
172 .....    """
173 .....    Args:
174 .....    .....None.
175 .....    Returns:
176 .....    .....[strID (string), strTitle (string), strArtist (string)]:
177 .....    .....a list of strings of the users input for the new CD ID, Title and Artist
178 .....    """
179 .....    strID = input('Enter ID: ').strip()
180 .....    strTitle = input('What is the CD\'s title? ').strip()
181 .....    strArtist = input('What is the Artist\'s name? ').strip()
182 .....    return [strID, strTitle, strArtist]
```

Figure 3: `cd_input` function

The other part of the add CD functionality, I put in the `DataProcessor` class in the function `add_CD`. This takes in parameters, `cdData`, which is the list of strings from our `cd_input` function in the script, and `table`, which is the 2D data table that stores our inventory during processing.

```
22 .....@staticmethod
23 .....def add_CD(cdData, table):
24 .....    """Function to add a new row of data to the inventory
25 .....    """
26 .....    Takes in the data from the cdData list and puts it in a new row (dictionary) to add to the
27 .....    2D table (list of dicts) specified by table
28 .....    Args:
29 .....    .....cdData (list of strings): list of user inputted data (strings) for the new cd
30 .....    .....table (list of dicts): 2D data structure (list of dicts) that holds the data during runtime
31 .....    Returns:
32 .....    .....None.
33 .....    """
34 .....    dicRow = {'ID': int(cdData[0]), 'Title': cdData[1], 'Artist': cdData[2]}
35 .....    table.append(dicRow)
```

Figure 4: `add_cd` function

Doc Strings

For all these functions that I created, I also added Doc Strings. These add additional information to our functions that explain what they do and how they work⁴. In Figure 5 below, there is an example.

³ Biesinger, Foundations of Programming (Python), Module 06 Page 5

⁴ Biesinger, Foundations of Programming (Python), Module 06 Pages 18-19

```

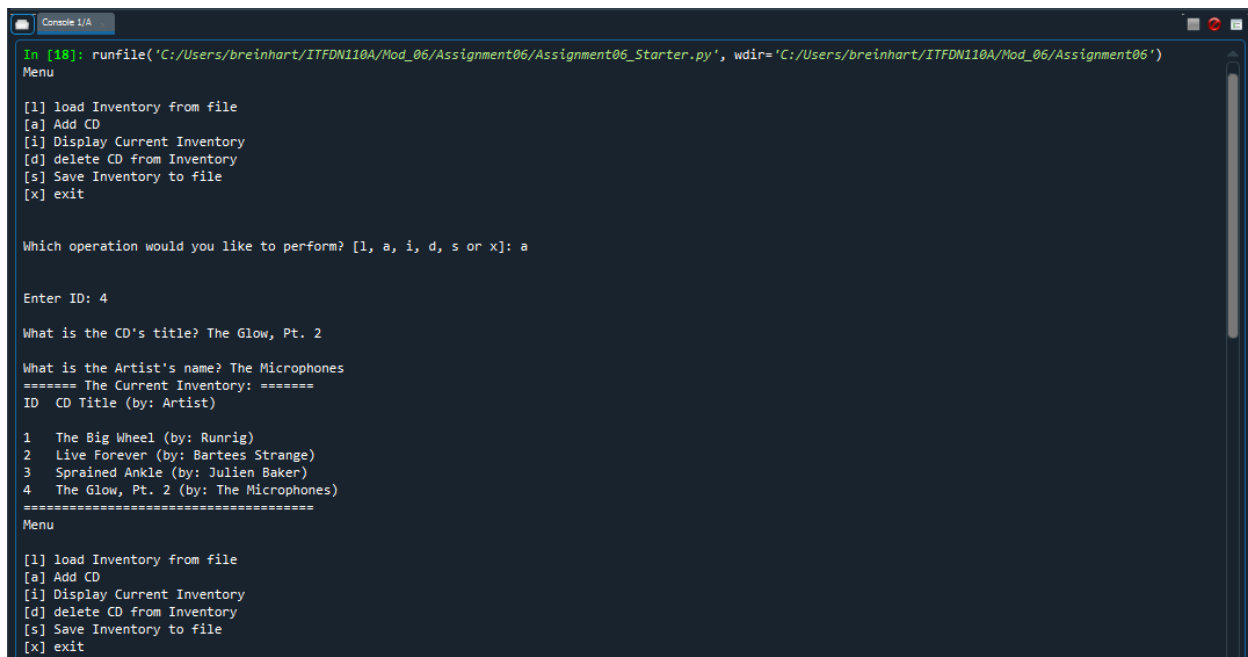
23     ....def add_CD(cdData, table):
24     ....    """Function to add a new row of data to the inventory
25     ....    """
26     ....    Takes in the data from the cdData list and puts it in a new row (dictionary) to add to the
27     ....    2D table (list of dicts) specified by table
28     ....    """
29     ....    Args:
30     ....    ....cdData (list of strings): List of user inputted data (strings) for the new cd
31     ....    ....table (list of dicts): 2D data structure (list of dicts) that holds the data during runtime
32     ....    """
33     ....    Returns:
34     ....    ....None.
35     ....    """
36     ....    dicRow = {'ID': int(cdData[0]), 'Title': cdData[1], 'Artist': cdData[2]}
37     ....    table.append(dicRow)
38

```

Figure 5: Doc Strings example

Running the Script - Spyder

To test the modifications I made, I first ran the program in Spyder. I wanted to test all functionality, so I loaded in a current file with a couple CDs then displayed, added, deleted, saved and exited. In Figures 6 through 9 below you can see the code running smoothly in Spyder.



```

In [18]: runfile('C:/Users/breinhard/ITFDN110A/Mod_06/Assignment06/Assignment06_Starter.py', wdir='C:/Users/breinhard/ITFDN110A/Mod_06/Assignment06')
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: a

Enter ID: 4

What is the CD's title? The Glow, Pt. 2

What is the Artist's name? The Microphones
===== The Current Inventory: =====
ID  CD Title (by: Artist)

1   The Big Wheel (by: Runrig)
2   Live Forever (by: Barteas Strange)
3   Sprained Ankle (by: Julien Baker)
4   The Glow, Pt. 2 (by: The Microphones)
=====
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

```

Figure 6: Add CD in Spyder

```
Console I/A
Menu
[1] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [1, a, i, d, s or x]: d

===== The Current Inventory: =====
ID  CD Title (by: Artist)

1   The Big Wheel (by: Runrig)
2   Live Forever (by: Barteas Strange)
3   Sprained Ankle (by: Julien Baker)
4   The Glow, Pt. 2 (by: The Microphones)
=====

Which ID would you like to delete? 3
The CD was removed
===== The Current Inventory: =====
ID  CD Title (by: Artist)

1   The Big Wheel (by: Runrig)
2   Live Forever (by: Barteas Strange)
4   The Glow, Pt. 2 (by: The Microphones)
=====
```

Figure 7: Successful Delete in Spyder

```
Console I/A
Menu
[1] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [1, a, i, d, s or x]: d

===== The Current Inventory: =====
ID  CD Title (by: Artist)

1   The Big Wheel (by: Runrig)
2   Live Forever (by: Barteas Strange)
4   The Glow, Pt. 2 (by: The Microphones)
=====

Which ID would you like to delete? 5
Could not find this CD!
===== The Current Inventory: =====
ID  CD Title (by: Artist)

1   The Big Wheel (by: Runrig)
2   Live Forever (by: Barteas Strange)
4   The Glow, Pt. 2 (by: The Microphones)
=====
```

Figure 8: Unsuccessful Delete in Spyder

```
=====
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: s

===== The Current Inventory: =====
ID   CD Title (by: Artist)
1    The Big Wheel (by: Runrig)
2    Live Forever (by: Barteas Strange)
4    The Glow, Pt. 2 (by: The Microphones)
=====

Save this inventory to file? [y/n] y
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: x

In [19]: |
```

Figure 9: Save to file and exit in Spyder

To ensure the file is running correctly, we also need to check the text is updated accordingly. Here is the text file after the above is run in Spyder where we read in three lines of data, added a fourth, then deleted a line before saving. It has three lines with one of the original lines having been deleted.

```
CDInventory.txt - Notepad
File Edit Format View Help
1,The Big Wheel,Runrig
2,Live Forever,Barteas Strange
4,The Glow, Pt. 2,The Microphones

Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

Figure 10: Spyder Output

Running the Script - Terminal

We also ran the program through the Terminal to finish testing. First, we added a CD then deleted a CD. Finally, we saved to the file then we exited the program. You can see this successful run in Figures 11 through 13 below.

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\breinhart>cd C:\Users\breinhart\ITFDN110A\Mod_06\Assignment06
(base) C:\Users\breinhart\ITFDN110A\Mod_06\Assignment06>python Assignment06.py
Menu
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: a

Enter ID: 3
What is the CD's title? The Body, The Blood, The Machine
What is the Artist's name? The Thermals
===== The Current Inventory: =====
ID      CD Title (by: Artist)
1       The Big Wheel (by: Runrig)
2       Live Forever (by: Barteas Strange)
4       The Glow (by: Pt. 2)
3       The Body, The Blood, The Machine (by: The Thermals)
=====
```

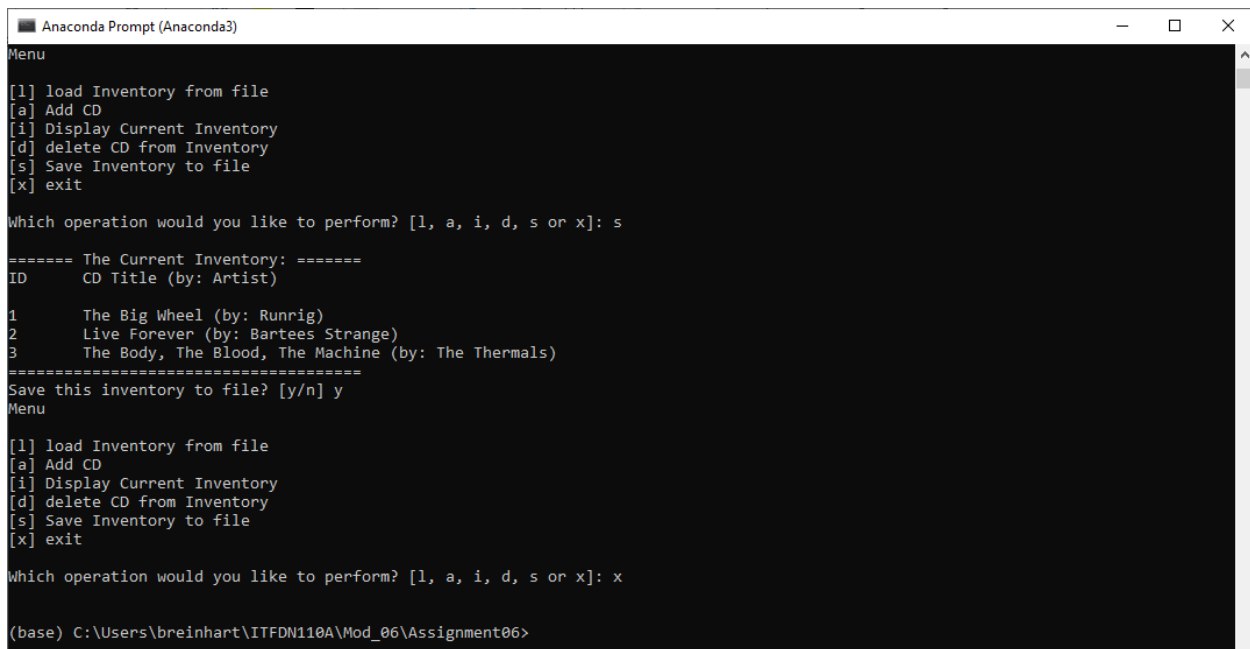
Figure 11: Add CD in Terminal

```
Anaconda Prompt (Anaconda3)
Menu
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: d

===== The Current Inventory: =====
ID      CD Title (by: Artist)
1       The Big Wheel (by: Runrig)
2       Live Forever (by: Barteas Strange)
4       The Glow (by: Pt. 2)
3       The Body, The Blood, The Machine (by: The Thermals)
=====
Which ID would you like to delete? 4
The CD was removed
===== The Current Inventory: =====
ID      CD Title (by: Artist)
1       The Big Wheel (by: Runrig)
2       Live Forever (by: Barteas Strange)
3       The Body, The Blood, The Machine (by: The Thermals)
=====
```

Figure 12: Delete CD in Terminal



```
Anaconda Prompt (Anaconda3)
Menu
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: s

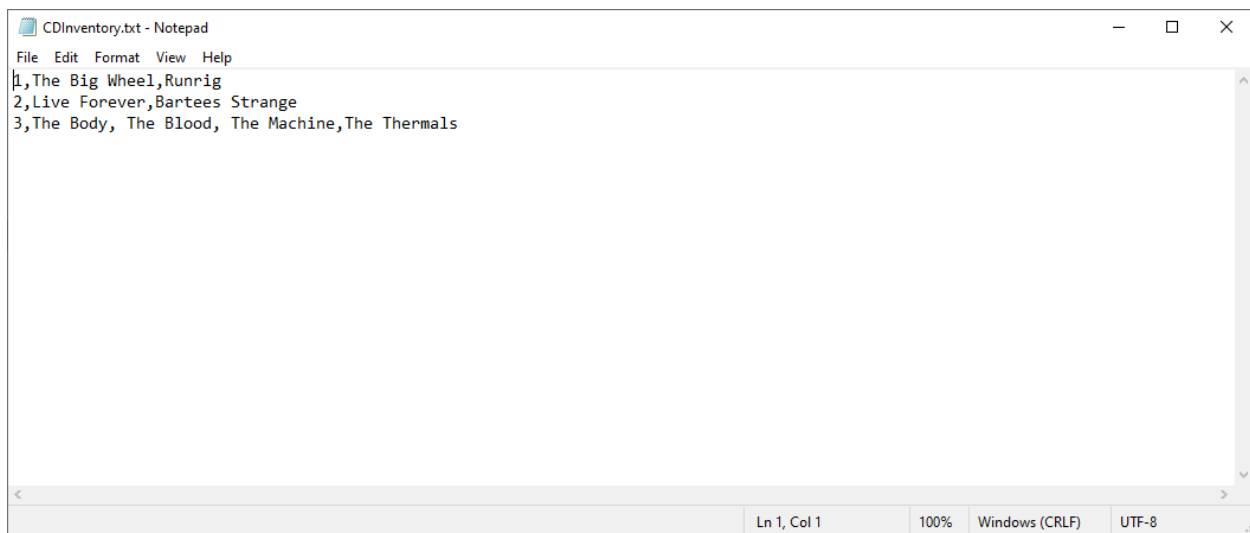
===== The Current Inventory: =====
ID      CD Title (by: Artist)
1       The Big Wheel (by: Runrig)
2       Live Forever (by: Barteas Strange)
3       The Body, The Blood, The Machine (by: The Thermals)
=====
Save this inventory to file? [y/n] y
Menu
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: x

(base) C:\Users\breinhardt\ITFDN110A\Mod_06\Assignment06>
```

Figure 13: Save and Exit in Terminal

Finally, we want to ensure the output saved to the file is correct. We used the file that we finished with on our Spyder run, added a CD and deleted a CD, so we have the correct file below.



```
CDInventory.txt - Notepad
File Edit Format View Help
1,The Big Wheel,Runrig
2,Live Forever,Barteas Strange
3,The Body, The Blood, The Machine,The Thermals

Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Figure 14: Terminal File Output

Summary

Given the information provided throughout Module 06, I was able to modify this starter script that helps a user manage a CD inventory by moving functionality to functions inside of separate classes.