Ben Streck

20 Nov 2022

Foundations of Programming (Python)

Assignment 06

Classes and Functions

Introduction

In this assignment, I outline the process of creating a CD inventory script in python. This script differs from Assignment04.py and Assignment05.py because it uses classes and functions to complete the tasks. The script presents users with a menu that gives them six options: 'Load Inventory from File', 'Add CD', 'Display Current Inventory', 'Delete CD from Inventory', 'Save Inventory to File' and 'Exit.' Each menu option has different functionality and is accompanied by code that appends, deletes, displays, or saves data to the CDInventory.txt file. Completing this assignment requires an understanding of classes, functions, loops, dictionaries, lists, and strings. It also draws on some skills we learned in previous weeks such as reading and writing data to external text files. In CDInventory.py, I organized my script into three main sections: Data, Processing, and Presentation (Input/ Output). I used classes to organize my functions into logical groups such as data processing, file processing, and input/output. I used a while loop with conditional statements to display the available options and run the correct functions based on user inputs. Additionally, I used open(), split(), strip(), write(), and close() to read data, add data, and save data to the text file. Lastly, we created a GitHub repository for this assignment with the intention of peer-review activities throughout the week.

Writing the Script

Data Processing

Listing 1 shows the DataProcessor() class which includes functions for adding CDs, deleting CDs, loading the CD inventory, and saving the CD inventory. All the code contained here was either moved from its original position in Assignment06_Starter.py or copied from my Assignment05 submission. I included docstrings to explain the purpose of each function along with the required input arguments. The docstrings also detail the contents of each output if applicable. The docstrings are not shown in this listing but can be seen in the appendix under Full Listing - CDInventory.py.

```
21  class DataProcessor:
26      @staticmethod
27      def add_CD(table, ID, strTitle, strArtist):
40          dicRow = {'ID': ID, 'Title': strTitle, 'Artist': strArtist}
41          table.append(dicRow)
42          print()
43          IO.show_inventory(table)
45      @staticmethod
46      def delete_CD(table, intIDDel):
58          if intIDDel < 1:
59              print('\nID Number Invalid...\n')
60              print('No Entries Deleted\n')
61          elif intIDDel > len(table):
62              print('\nID Number Invalid... There are not that many CDs in the inverntory\n')
63              print('No Entries Deleted\n')
64          else:
65              table = list(filter(lambda i: i['ID'] != intIDDel, table))
66              print('\nEntry Deleted')
67              print('Relabeling ID Numbers...')
68              i = 1
69              for row in table:
70                  row['ID'] = i
71                  i += 1
72              print('ID numbers have been updated\n')
73          IO.show_inventory(table)
74          return table
```

```
76      @staticmethod
77      def load_inventory(file_name, table):
90          print('WARNING: If you continue, all unsaved data will be lost when the Inventory is re-loaded.\n')
91          print('Type \'yes\' to continue and reload data from the file. Otherwise reload will be canceled.')
92          strYesNo = input('Would you like to continue? ')
93          if strYesNo.lower() == 'yes':
94              print('\nReloading...')
95              FileProcessor.read_file(file_name, table)
96              IO.show_inventory(table)
97          else:
98              input('Canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')
99              IO.show_inventory(table)
101     @staticmethod
102     def save_inventory(file_name, table):
114         IO.show_inventory(table)
115         strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
116         if strYesNo == 'y':
117             print('\nSaving updated inventory...')
118             FileProcessor.write_file(file_name, table)
119             print('Done')
120         else:
121             input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
```

Listing 1 – Data Processing

File Processing

Listing 2 shows the FileProcessor() class which includes functions for reading data from a text file and writing data to a text file. All the code contained here was either moved from its original position in Assignment06_Starter.py or copied from my Assignment05 submission. I included docstrings to explain the purpose of each function along with the required input arguments. The docstrings also detail the contents of each output if applicable. The docstrings are not shown in this listing but can be seen in the appendix under Full Listing - CDInventory.py.

```
124 class FileProcessor:
129     @staticmethod
130     def read_file(file_name, table):
143         table.clear()  # this clears existing data and allows to load data from file
144         try:
145             objFile = open(file_name, 'r')
146             for line in objFile:
147                 data = line.strip().split(',')
148                 dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
149                 table.append(dicRow)
150             objFile.close()
151         except:
152             print('\n{} does not exist...'.format(file_name))
153             file = open(file_name, 'w')
154             file.close()
155             print('The file has now been created!')
157     @staticmethod
158     def write_file(file_name, table):
169         objFile = open(file_name, 'w')
170         for row in lstTbl:
171             lstValues = list(row.values())
172             lstValues[0] = str(lstValues[0])
173             objFile.write(','.join(lstValues) + '\n')
174         objFile.close()
```

Listing 2 – File Processing

Input/ Output

Listing 3 shows the Input/ Output class which includes functions for printing the menu, accepting menu choices, displaying the current inventory, accepting new CD data, and accepting index numbers for CD deletions. All the code contained here was either moved from its original position in Assignment06_Starter.py or copied from my Assignment05 submission. I included docstrings to explain the purpose of each function along with the required input arguments. The docstrings also detail the contents of each output if applicable. The docstrings are not shown in this listing but can be seen in the appendix under Full Listing - CDInventory.py.

```
178 class IO:
179     """
180     Handling Input / Output
181     """
183     @staticmethod
184     def print_menu():
194         print('\nMenu\n\n[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
195         print('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit\n')
197     @staticmethod
198     def menu_choice():
208         choice = ' '
209         while choice not in ['l', 'a', 'i', 'd', 's', 'x']:
210             choice = input('Which operation would you like to perform? [l, a, i, d, s or x]: ').lower().strip()
211         print()  # Add extra space for layout
212         return choice
214     @staticmethod
215     def show_inventory(table):
225         print('======= The Current Inventory: =======')
226         print('ID\tCD Title (by: Artist)\n')
227         for row in table:
228             print('{}\t{} (by: {})'.format(*row.values()))
229         print('======================================')
231     @staticmethod
232     def new_CD_choice(table):
245         ID = len(table) + 1
246         strTitle = input('What is the CD\'s title? ').strip()
247         strArtist = input('What is the Artist\'s name? ').strip()
248         return ID, strTitle, strArtist
250     @staticmethod
251     def del_CD_choice(table):
262         print('Deleting an entry from the CD Inventory...')
263         print('What is the ID number of the entry you want to delete?\n')
264         IO.show_inventory(table)
265         intIDDel = int(input('Enter ID Number Here: ').strip())
266         return intIDDel
```

Listing 3 - Input/ Output

While Loop

Listing 4 shows the while loop that I used to execute all the functions in an organized manner. The loop has conditional statements that separate the menu options from one another. The loop structure was already present in Assignment06_Starter.py, but I moved most of the code into functions and condensed this section. Now, the while loop's only purpose is to call functions from throughout the script.

```
269 # 1. When program starts, read in the Current Inventory
270 FileProcessor.read_file(strFileName, lstTbl)
272 # 2. Start main loop
273 while True:
275     # 3. Display menu to user and get choice
276     IO.print_menu()
277     strChoice = IO.menu_choice()
279     # 4. Process menu selections
281         # 4.1 Exit
```

```
282     if strChoice == 'x':
283         print('Goodbye...')
284         break
286         # 4.2 Load Inventory
287     if strChoice == 'l':
288         DataProcessor.load_inventory(strFileName, lstTbl)
289         continue
291         # 4.3 Add a CD
292     elif strChoice == 'a':
293         ID, strTitle, strArtist = IO.new_CD_choice(lstTbl)
294         DataProcessor.add_CD(lstTbl, ID, strTitle, strArtist)
295         continue  # start loop back at top.
297         # 4.4 Display Current Inventory
298     elif strChoice == 'i':
299         IO.show_inventory(lstTbl)
300         continue  # start loop back at top.
302         # 4.5 Delete a CD
303     elif strChoice == 'd':
304         intIDDel = IO.del_CD_choice(lstTbl)
305         lstTbl = DataProcessor.delete_CD(lstTbl, intIDDel)
306         continue  # start loop back at top.
308         # 4.6 Save Inventory to File
309     elif strChoice == 's':
310         DataProcessor.save_inventory(strFileName, lstTbl)
311         continue  # start loop back at top.
313         # 4.7 Catch-All Error... Should not be possible because the user's choice gets vetted in IO
314     else:
315         print('Invalid Input...\n')
316         print('Please choose one of the options listed\n')
```

Listing 4 – While Loop

Saving the Script

As instructed, I created a folder in C:\ _PythonClass\ called ‘Assignment06’ and saved my script as CDInventory.py.

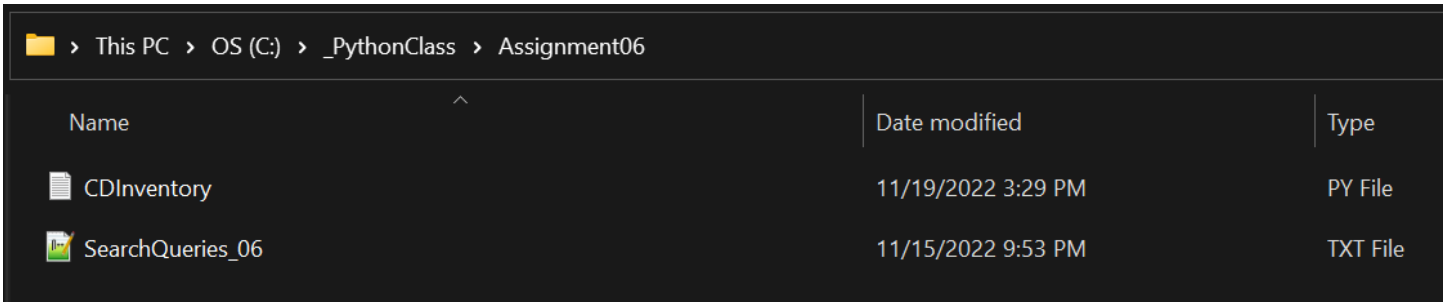

Figure 1 – Saving CDInventory.py

Running the Script

Spyder

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

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

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

Saving updated inventory...
Done

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]: i

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================
```

```
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

Deleting an entry from the CD Inventory...
What is the ID number of the entry you want to delete?

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

Enter ID Number Here: 3

Entry Deleted
Relabeling ID Numbers...
ID numbers have been updated

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
=======================================
```

```
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]: l

WARNING: If you continue, all unsaved data will be lost when the Inventory is re-loaded.

Type 'yes' to continue and reload data from the file. Otherwise reload will be canceled.

Would you like to continue? yes

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

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

Deleting an entry from the CD Inventory...
What is the ID number of the entry you want to delete?

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

Enter ID Number Here: 0

ID Number Invalid...

No Entries Deleted

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

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

Deleting an entry from the CD Inventory...
What is the ID number of the entry you want to delete?

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

Enter ID Number Here: 10

ID Number Invalid... There are not that many CDs in the inverntory

No Entries Deleted

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

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

Deleting an entry from the CD Inventory...
What is the ID number of the entry you want to delete?

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

1   Purgatory (by: Tyler Childers)
2   Vitalogy (by: Pearl Jam)
3   JT (by: James Taylor)
=======================================

Enter ID Number Here: 1

Entry Deleted
Relabeling ID Numbers...
ID numbers have been updated

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

1   Vitalogy (by: Pearl Jam)
2   JT (by: James Taylor)
=======================================

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
```

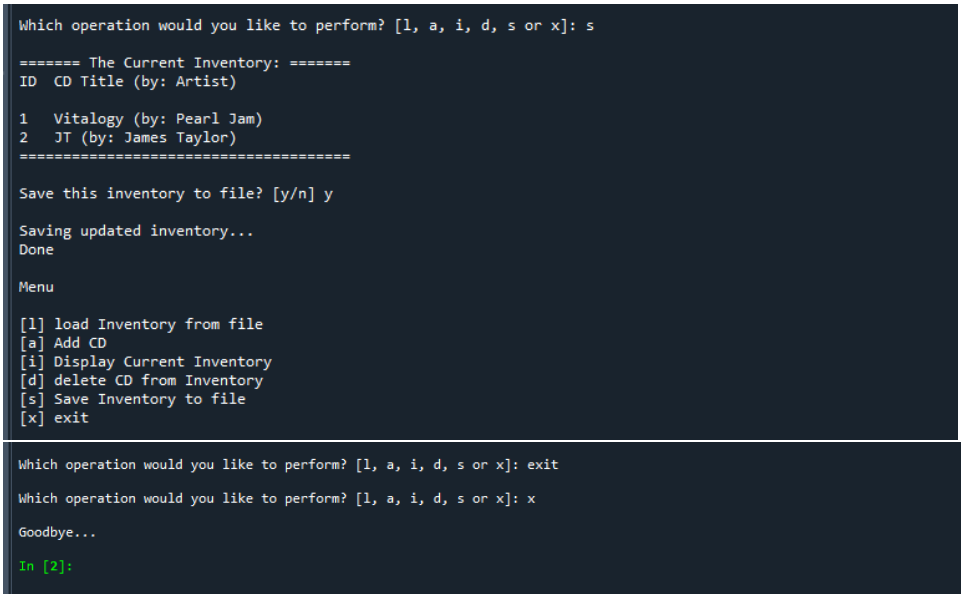

Figure 2 – Running CDInventory.py in Spyder

I opened Spyder on my Windows computer, opened CDInventory.py, and clicked F5 to execute the script. I followed the user prompts and entered 'i' to display the current CD inventory. It was empty and the program behaved correctly. Next, I used 'a' three times to add three CDs and entered the necessary information when prompted. I followed that up with the 's' command and an 'i' command to save the updated inventory and then display it to the user. Afterwards, I used 'd' to delete the third CD from the inventory. I used 'l' next to load the last save from CDInventory.txt. This undid my deletion of the third CD. Then, I showed that the delete option displays useful information when the user enters an ID number outside of the data range. I also deleted CD number one and then saved my changes. Finally, I entered 'exit' to show the script's response to an invalid input followed by 'x' to exit the program. CDInventory.py ran correctly all the way through. It accepted user inputs, read data, modified data, displayed data, and saved data as intended. Figure 2 shows that the script functions correctly while running in the Spyder IDE.

Terminal

```
Anaconda Prompt (anaconda3)

(base) C:\_PythonClass\Assignment06>python CDInventory.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]: i

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

1       Vitalogy (by: Pearl Jam)
2       JT (by: James Taylor)
=======================================

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

What is the CD's title? American Teen
What is the Artist's name? Khalid

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

1       Vitalogy (by: Pearl Jam)
2       JT (by: James Taylor)
3       American Teen (by: Khalid)
=======================================

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]: l

WARNING: If you continue, all unsaved data will be lost when the Inventory is re-loaded.

Type 'yes' to continue and reload data from the file. Otherwise reload will be canceled.
Would you like to continue? yes

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

1       Vitalogy (by: Pearl Jam)
2       JT (by: James Taylor)
=======================================

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

What is the CD's title? Ride the Lightning
What is the Artist's name? Metallica

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

1       Vitalogy (by: Pearl Jam)
2       JT (by: James Taylor)
3       Ride the Lightning (by: Metallica)
=======================================

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

What is the CD's title? Abbey Road
What is the Artist's name? The Beatles

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

1       Vitalogy (by: Pearl Jam)
2       JT (by: James Taylor)
3       Ride the Lightning (by: Metallica)
4       Abbey Road (by: The Beatles)
=======================================
```

```
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

What is the CD's title? Electric Ladyland
What is the Artist's name? Jimi Hendrix

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

1       Vitalogy (by: Pearl Jam)
2       JT (by: James Taylor)
3       Ride the Lightning (by: Metallica)
4       Abbey Road (by: The Beatles)
5       Electric Ladyland (by: Jimi Hendrix)
=======================================
```

Figure 3 – Running CDInventory.py in Terminal

I opened terminal on my Windows computer and navigated to the correct folder using the cd (change directory) command. Then, I ran the command 'python CDInventory.py' to execute the script. I followed the user prompts and entered 'i' to show that the program loaded the data correctly. I ran the script on terminal after I had already run it on Spyder, so there were two CDs in the inventory. Next, I used 'a' to add a CD and entered the necessary information when prompted. I followed that up with the 'l' command to reload the last save from CDInventory.txt. I used 'a' three more times to add three new CDs and then used 's' to save my changes. Afterwards, I used 'd' to delete the second CD from the inventory followed by 's' to save and 'x' to exit. CDInventory.py ran correctly all the way through. It accepted user inputs, read data, modified data, displayed data, and saved data as intended. Figure 3 shows that the script functions correctly while running in terminal.

Checking the Text File

After running the script in Spyder and then in a terminal window, I opened the text file to verify that it had been created, read, edited, and saved properly. As shown in Figure 4, everything worked correctly and CDInventory.txt had all the right information.

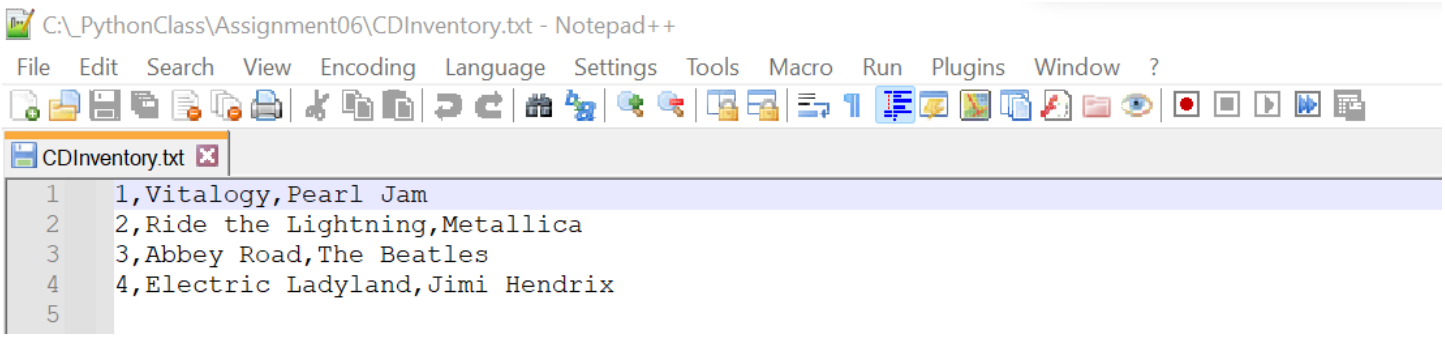

Figure 4 – Verifying Script Functionality

GitHub Repository

Link: https://github.com/BenStreck/Assignment_06

Summary

I successfully created a python script that fulfills the requirements listed in Assignment 06. I did so by using information from the textbook, the Module 06 videos, and the supplemental learning documents. The script demonstrates my understanding of classes, functions, and data manipulation.

My only confusion with this week's assignment was where to group some of the functions. Specifically, I wasn't sure whether to include load_inventoy() and save_inventory() in the DataProcessor class or the IO class. They both receive some sort of user input, but they also use those inputs to execute operations on the data. The script runs just fine, but the grouping of information seemed a bit arbitrary to me.

Appendix

Full Listing – CDInventory.py

```
#------------------------------------------#
# Title: CDInventory.py
# Desc: This is a script to store CD Inventory Data
#       This script demonstrates my understanding of how to work with
#       classes and functions
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# BStreck, 2022-Nov-16, Started adding functionality in the 'TO-DO' sections
# BStreck, 2022-Nov-19, Finished adding functionality in the 'TO-DO' sections
#------------------------------------------#

# -- DATA -- #
strChoice = '' # User input
lstTbl = []  # list of lists to hold data
dicRow = {}  # list of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object


# -- PROCESSING -- #
class DataProcessor:
```

```
    """
    Processing the data during runtime
    """

    @staticmethod
    def add_CD(table, ID, strTitle, strArtist):
        """
        Function to add a new CD to the current inventory and show the updated inventory afterwards.

        Args:
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
            ID (integer): the ID number of the new CD being added to the inventory
            strTitle (string): the title of the new CD being added to the inventory
            strArtist (string): the artist of the new CD being added to the inventory

        Returns:
            None
        """
        dicRow = {'ID': ID, 'Title': strTitle, 'Artist': strArtist}
        table.append(dicRow)
        print()
        IO.show_inventory(table)

    @staticmethod
    def delete_CD(table, intIDDel):
        """
        Function to delete a CD from the current inventory and show the updated inventory afterwards.
        It also relabels the ID numbers to prevent discontinuities in the inventory.

        Args:
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
            intIDDel (integer): the ID number of the CD being deleted from the inventory

        Returns:
            table (list of dict): updated 2D data structure (list of dicts) that holds the data during runtime
        """
        if intIDDel < 1:
            print('\nID Number Invalid...\n')
            print('No Entries Deleted\n')
        elif intIDDel > len(table):
            print('\nID Number Invalid... There are not that many CDs in the inverntory\n')
            print('No Entries Deleted\n')
        else:
            table = list(filter(lambda i: i['ID'] != intIDDel, table))
            print('\nEntry Deleted')
            print('Relabeling ID Numbers...')
            i = 1
            for row in table:
                row['ID'] = i
                i += 1
            print('ID numbers have been updated\n')
        IO.show_inventory(table)
        return table

    @staticmethod
    def load_inventory(file_name, table):
        """
        Function managing the FileProcessor.read_file() function.
        This helps prevent unintentional overwriting of data in the current inventory.
```

```
81          It also shows the current inventory after it has been loaded.
82
83          Args:
84              file_name (string): name of file used to read the data from
85              table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
86
87          Returns:
88              None
89          """
90          print('WARNING: If you continue, all unsaved data will be lost when the Inventory is re-loaded.\n')
91          print('Type \'yes\' to continue and reload data from the file. Otherwise reload will be canceled.')
92          strYesNo = input('Would you like to continue? ')
93          if strYesNo.lower() == 'yes':
94              print('\nReloading...')
95              FileProcessor.read_file(file_name, table)
96              IO.show_inventory(table)
97          else:
98              input('Canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')
99              IO.show_inventory(table)
100
101     @staticmethod
102     def save_inventory(file_name, table):
103         """
104         Function managing the FileProcessor.write_file() function.
105         It shows the current inventory prior to saving which allows users to verify they are saving the correct data.
106
107         Args:
108             file_name (string): name of file used to read the data from
109             table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
110
111         Returns:
112             None
113         """
114         IO.show_inventory(table)
115         strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
116         if strYesNo == 'y':
117             print('\nSaving updated inventory...')
118             FileProcessor.write_file(file_name, table)
119             print('Done')
120         else:
121             input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
122
123
124 class FileProcessor:
125     """
126     Processing the data to and from text file
127     """
128
129     @staticmethod
130     def read_file(file_name, table):
131         """
132         Function to manage data intake from the text file to a list of dictionaries.
133         The function reads data from the file identified by 'file_name' into a 2D table
134         (list of dicts). One line in the file represents one dictionary row in table.
135
136         Args:
137             file_name (string): name of file used to read the data from
138             table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
139
```

```
        Returns:
            None
        """
        table.clear()  # this clears existing data and allows to load data from file
        try:
            objFile = open(file_name, 'r')
            for line in objFile:
                data = line.strip().split(',')
                dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
                table.append(dicRow)
            objFile.close()
        except:
            print('\n{} does not exist...'.format(file_name))
            file = open(file_name, 'w')
            file.close()
            print('The file has now been created!')

    @staticmethod
    def write_file(file_name, table):
        """
        Function to manage data writing from the list of dictionaries to a text file.

        Args:
            file_name (string): name of file used to read the data from
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            None
        """
        objFile = open(file_name, 'w')
        for row in lstTbl:
            lstValues = list(row.values())
            lstValues[0] = str(lstValues[0])
            objFile.write(','.join(lstValues) + '\n')
        objFile.close()


# -- PRESENTATION (Input/Output) -- #
class IO:
    """
    Handling Input / Output
    """

    @staticmethod
    def print_menu():
        """
        Displays a menu of choices to the user

        Args:
            None

        Returns:
            None
        """
        print('\nMenu\n\n[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
        print('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit\n')

    @staticmethod
    def menu_choice():
```

```
        """
        Gets user input for menu selection

        Args:
            None

        Returns:
            choice (string): a lower case string of the users input out of the choices l, a, i, d, s or x
        """
        choice = ' '
        while choice not in ['l', 'a', 'i', 'd', 's', 'x']:
            choice = input('Which operation would you like to perform? [l, a, i, d, s or x]: ').lower().strip()
        print()  # Add extra space for layout
        return choice

    @staticmethod
    def show_inventory(table):
        """
        Displays the current inventory table

        Args:
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            None
        """
        print('======= The Current Inventory: =======')
        print('ID\tCD Title (by: Artist)\n')
        for row in table:
            print('{}\t{} (by: {})'.format(*row.values()))
        print('======================================')

    @staticmethod
    def new_CD_choice(table):
        """
        Function to accept user inputs for a new CD.
        The data will be added to the current inventory using the DataProcessor.add_CD() function.

        Args:
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            ID (integer): the ID number of the new CD being added to the inventory
            strTitle (string): the title of the new CD being added to the inventory
            strArtist (string): the artist of the new CD being added to the inventory
        """
        ID = len(table) + 1
        strTitle = input('What is the CD\'s title? ').strip()
        strArtist = input('What is the Artist\'s name? ').strip()
        return ID, strTitle, strArtist

    @staticmethod
    def del_CD_choice(table):
        """
        Function to accept user inputs for deleting a CD.
        The chosen CD will be removed from the current inventory using the DataProcessor.delete_CD() function.

        Args:
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
```

```
258
259         Returns:
260             intIDDel (integer): the ID number of the CD being deleted from the inventory
261         """
262         print('Deleting an entry from the CD Inventory...')
263         print('What is the ID number of the entry you want to delete?\n')
264         IO.show_inventory(table)
265         intIDDel = int(input('Enter ID Number Here: ').strip())
266         return intIDDel
267
268
269 # 1. When program starts, read in the Current Inventory
270 FileProcessor.read_file(strFileName, lstTbl)
271
272 # 2. Start main loop
273 while True:
274
275     # 3. Display menu to user and get choice
276     IO.print_menu()
277     strChoice = IO.menu_choice()
278
279     # 4. Process menu selections
280
281         # 4.1 Exit
282     if strChoice == 'x':
283         print('Goodbye...')
284         break
285
286         # 4.2 Load Inventory
287     if strChoice == 'l':
288         DataProcessor.load_inventory(strFileName, lstTbl)
289         continue
290
291         # 4.3 Add a CD
292     elif strChoice == 'a':
293         ID, strTitle, strArtist = IO.new_CD_choice(lstTbl)
294         DataProcessor.add_CD(lstTbl, ID, strTitle, strArtist)
295         continue  # start loop back at top.
296
297         # 4.4 Display Current Inventory
298     elif strChoice == 'i':
299         IO.show_inventory(lstTbl)
300         continue  # start loop back at top.
301
302         # 4.5 Delete a CD
303     elif strChoice == 'd':
304         intIDDel = IO.del_CD_choice(lstTbl)
305         lstTbl = DataProcessor.delete_CD(lstTbl, intIDDel)
306         continue  # start loop back at top.
307
308         # 4.6 Save Inventory to File
309     elif strChoice == 's':
310         DataProcessor.save_inventory(strFileName, lstTbl)
311         continue  # start loop back at top.
312
313         # 4.7 Catch-All Error... Should not be possible because the user's choice gets vetted in IO
314     else:
315         print('Invalid Input...\n')
316         print('Please choose one of the options listed\n')
```