Daisy Pandey
August 19, 2020
Foundations of Programming: Python
Assignment 06

# CD Inventory Program Python Script: Using Classes and Functions

## Introduction

Module 6 introduces functions and classes. It covers what are functions, creating your own functions, using parameters and return values, the difference between a global and a local variable, and using a docstring (documentation string) to describe what the specific function does.

This document provides the steps I took to modify the last week's CD Inventory program by using the example solution of last week's assignment (provided by the instructor). The following are the modifications that are made:

- Created new functions
- Added new codes
- Moved existing code to the specific functions
- Created new docstrings
- Modified and added a few comments

## Modifying the CD Inventory Python Script

In this module, we were asked to modify the CD Inventory Python script. Assignment 06 is similar to Assignment 05; however, it uses functions and classes. A function is a block of organized, reusable code that is used to perform a specific action/task. Creating and using your own functions allows you to break your code into smaller chunks, making the program more readable, manageable, and organized. To create a function, you start with the keyword *def*, followed by the function name, parenthesis (), colon, and block of statements. In addition, functions also use parameters that allow them to receive a value for processing.

https://www.learnpython.org/en/Functions [1] (external site)
https://www.programiz.com/python-programming/function [2] (external site)

To modify the CD Inventory Python script, I used the example solution of last week's assignment. First, I saved the script and used the Spyder as IDE. Second, I created a header that includes a few comments that list the title of the program, describes what the program is about, and stores the history of change logs with the name of programmer, date modified, and a brief description of changes (Figure 1).

```
#------------------------------------------#
# Title: CDInventory.py
# Desc: Working with classes and functions.
# Change Log: (Who, When, What)
# Daisy Pandey, August 16, 2020, Modifying CD Inventory Program script
# Daisy Pandey, August 16, 2020, Added code, added new functions, moved existing code to those functions
# Daisy Pandey, August 17, 2020, Added docstrings for add_data, del_data, and write_file functions, modified/added comments
# Daisy Pandey, August 19, 2020, Added get_UserInput function and docstring, added code to check if file exists
#------------------------------------------#
```

***Figure 1. Header with Comments***

[1] Retrieved August 18, 2020
[2] Retrieved August 18, 2020

Lastly, I made some modifications and organized the code.

- Created new functions and added docstrings:
  The DataProcessor class object groups two functions: add_data() and del_data(), shown in Figures 2 and 3.It allows to add CD data to the inventory and delete CD data from the inventory.
    - **add_data():** The function adds data to the 2D table (list of dictionaries). It receives its values through four parameters (strID, strTitle, stArtist, table).
    - **del_data():** The function asks user input and deletes CD data from inventory.

```
# -- PROCESSING -- #
class DataProcessor:
    """Adding CD data to the inventory and deleting CD data from inventory"""

    @staticmethod
    def add_data(strID, strTitle, stArtist, table):
        """Function to add data to the 2D table (list of dictionaries)

        Args:
            StrID (string): Input parameter for CD ID.
            Strtitle (string): Input parameter for CD Title.
            StArtist (string): Input parameter for CD Artist.
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            None.
        """
        # Add item to the table
        intID = int(strID)
        dicRow = {'ID': intID, 'Title': strTitle, 'Artist': stArtist}
        lstTbl.append(dicRow)
```

*Figure 2. add_data() function with docstring*

```
    @staticmethod
    def del_data():
        """Function to ask user input and delete CD data from inventory

        Args:
            None

        Returns:
            None
        """
        # Ask user which ID to remove
        intIDDel = int(input('Which ID would you like to delete? ').strip())

        # Search thru table and delete CD
        intRowNr = -1
        blnCDRemoved = False
        for row in lstTbl:
            intRowNr += 1
            if row['ID'] == intIDDel:
                del lstTbl[intRowNr]
                blnCDRemoved = True
                break
        if blnCDRemoved:
            print('The CD was removed')
        else:
            print('Could not find this CD!')
```

*Figure 3. del_data() function with docstring*

    - **get_userInput():** The function belongs to the IO class object. It gets user input for ID, CD title, and CD artist and returns three values (strID, strTitle, stArtist) through the return statement, shown in Figure 4.

```
    @staticmethod
    def get_userInput():
        """Function to get user input for ID, CD title, and CD artist

        Args:
            None.

        Returns:
            StrID (string): Input for CD ID.
            Strtitle (string): Input for CD Title.
            StArtist (string): Input for CD Artist.
        """
        # Ask user for new ID, CD Title and Artist
        strID = input('Enter ID: ').strip()
        strTitle = input('What is the CD\'s title? ').strip()
        stArtist = input('What is the Artist\'s name? ').strip()
        return strID, strTitle, stArtist
```

*Figure 4. get_userInput() function with docstring*

The following are other classes and functions that are defined in the program.

- **FileProcessor:** This class object processes the data to and from a text file. It groups two functions: read_file() and write_file().
    - **read_file():** This function loads the current CD data from the file into memory. It also uses parameters (file_name and table) that allow the function to receive values for processing. With the split() function, it returns the list of strings read in from the text file, separating the strings with a comma, and using the strip() function removes both the leading and trailing whitespace characters.
    - **write_file():** The function allows to write the data to a file. Like read_file() function, it also uses parameters (file_name and table) that allow the function to receive values for processing.

- **IO:** This class object handles inputs and outputs (I/O). It groups four functions: print_menu(), menu_choice(), show_inventory(), and get_userInput().
    - **print_menu():** The function displays a menu of choices to the user. The menu structure includes adding CD data, loading inventory from a file, viewing the current inventory, storing data to a text file, deleting an entry from inventory, and exiting the program.
    - **menu_choice():** The function gets user input for menu selection. It returns the choice (user-input) as a string type through the return statement.
    - **show_inventory():** The function displays the current inventory table returning only the dictionary values.

The while loop with continue is used to create the program for multiple inputs. The loop repeats a statement or group of statements while the condition is TRUE. Also, the program includes an if statement with the use of the elif clause.

### Running Python Script in Spyder

First, I executed the script in Spyder to ensure all the options in the script were functioning. In addition, I located the CDInventory.txt file, opened it in a text editor, and verified the program was saving the data to the file correctly. However, I received code review feedback saying that my program does not run if the CDInventory.txt file is not present in the directory. I modified my code with os.path library to check if the file does not exist then create a new empty file as shown in Figure 5. Then, I reran my script (Figures 6, 7, 8, 9, 10) and verified the data was written to the file correctly (Figure 11).

```
188     # If file does not exist in currrent folder, Create one
189     if not os.path.exists(strFileName):
190         newfile = open(strFileName, 'a')
191         newfile.close()
```

***Figure 5. Creating a new empty file if a file does not exist***

```
In [1]: runfile('C:/Users/daisy/CDInventory.py', wdir='C:/Users/daisy')
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 and the Inventory re-loaded from file.

type 'yes' to continue and reload from file. otherwise reload will be canceled: yes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
3       Smile (by:Katy Perry)
======================================
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]:
```

***Figure 6. Running script in Spyder with load (l) option***

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


Enter ID: 4

What is the CD's title? V

What is the Artist's name? Maroon 5
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
3       Smile (by:Katy Perry)
4       V (by:Maroon 5)
======================================
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]:
```

***Figure 7. Running script in Spyder with add (a) option***

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

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

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
3       Smile (by:Katy Perry)
4       V (by:Maroon 5)
======================================
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]:
```

***Figure 8. Running script in Spyder with display (i) option***

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

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

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
3       Smile (by:Katy Perry)
4       V (by:Maroon 5)
=======================================

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

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
4       V (by:Maroon 5)
=======================================
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 9. Running script in Spyder with delete (d) option*

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

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

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
4       V (by:Maroon 5)
=======================================

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 [2]:
```

*Figure 10. Running script in Spyder with save (s) and exit (x) options*

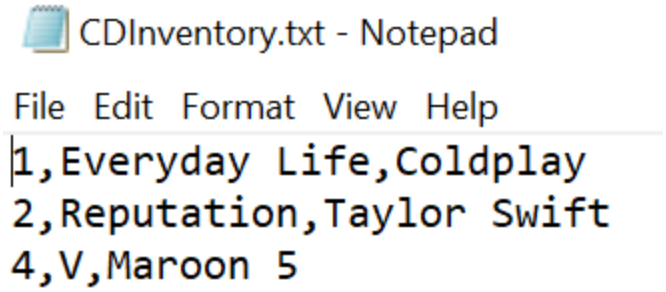


***Figure 11. Data Written in a CDInventory.txt File***

## Running Python Script in Prompt

I reran the script in Anaconda Prompt (Figures 12-16) and opened the text editor to ensure the data I had entered in prompt has been written to the file correctly, highlighted in Figure 17.

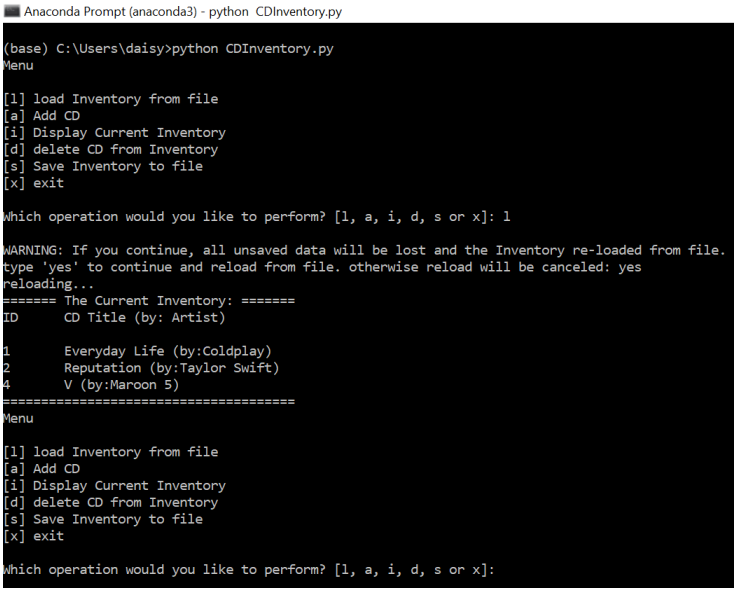


***Figure 12. Running script in prompt with load (l) option***

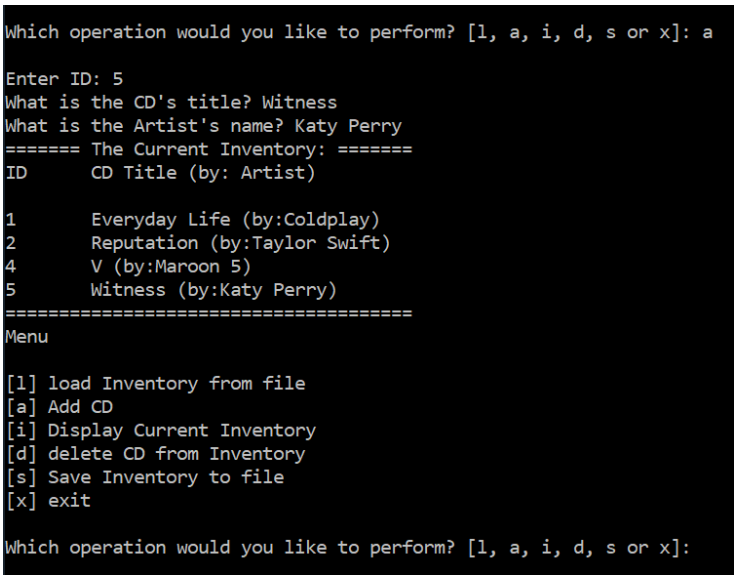


***Figure 13. Running script in prompt with add (a) option***

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

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

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
4       V (by:Maroon 5)
5       Witness (by:Katy Perry)
======================================
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]:
```

*Figure 14. Running script in prompt with display (i) option*

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

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

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
4       V (by:Maroon 5)
5       Witness (by:Katy Perry)
======================================
Which ID would you like to delete? 4
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
5       Witness (by:Katy Perry)
======================================
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]:
```

*Figure 15. Running script in prompt with delete (d) option*

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

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

1       Everyday Life (by:Coldplay)
2       Reputation (by:Taylor Swift)
5       Witness (by:Katy Perry)
======================================
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\daisy>
```

*Figure 16. Running script in prompt with save (s) and exit (x) options*

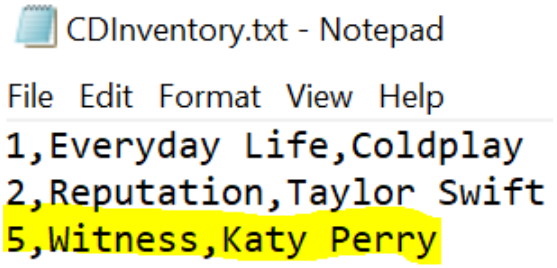


***Figure 17. Data Written in a CDInventory.txt File***

### GitHub Account

I have created a repository, "Assignment_06", where I have uploaded my knowledge document and CD Inventory Python script. The link to my GitHub repository is https://github.com/daisypandey/Assignment_06.

### Summary

With Module 6, I learned what is a function, what are parameters and return values, the difference between parameters and arguments, and the difference between a local and a global variable. After reviewing the Python Programming for the Absolute Beginner Textbook (Chapter 6), FDN_Py_Module_06 pdf document, videos, and few websites, I was able to successfully modify the CD Inventory Program Python Script by using classes and functions. This assignment demonstrates my knowledge on writing our own functions, calling these functions, using parameters and return values, creating docstrings, and using functions to organize code.

### Appendix

The following is the modified and final CD Inventory Program Python Script in the Planet B website. http://www.planetb.ca/syntax-highlight-word [3] (external site)

```
1.  #------------------------------------------#
2.  # Title: CDInventory.py
3.  # Desc: Working with classes and functions.
4.  # Change Log: (Who, When, What)
5.  # Daisy Pandey, August 16, 2020, Modifying CD Inventory Program script
6.  # Daisy Pandey, August 16, 2020, Added code, added new functions, moved existing code to those functi
    ons
7.  # Daisy Pandey, August 17, 2020, Added docstrings for add_data, del_data, and write_file functions, m
    odified/added comments
8.  # Daisy Pandey, August 19, 2020, Added get_UserInput function and docstring, added code to check if f
    ile exists
9.  #------------------------------------------#
10.
11. import os.path
12.
13. # -- DATA -- #
14. strChoice = '' # User input
15. lstTbl = []  # list of lists to hold data
16. dicRow = {}  # list of data row
17. strFileName = 'CDInventory.txt'  # data storage file
18. objFile = None  # file object
19.
20. # -- PROCESSING -- #
21. class DataProcessor:
22.     """Adding CD data to the inventory and deleting CD data from inventory"""
23.
24.     @staticmethod
```

[3] Retrieved August 19, 2020

```
    def add_data(strID, strTitle, stArtist, table):
        """Function to add data to the 2D table (list of dictionaries)

        Args:
            StrID (string): Input parameter for CD ID.
            Strtitle (string): Input parameter for CD Title.
            StArtist (string): Input parameter for CD Artist.
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtim
e

        Returns:
            None.
        """
       # Add item to the table
        intID = int(strID)
        dicRow = {'ID': intID, 'Title': strTitle, 'Artist': stArtist}
        lstTbl.append(dicRow)

    @staticmethod
    def del_data():
        """Function to ask user input and delete CD data from inventory

        Args:
            None

        Returns:
            None
        """
        # Ask user which ID to remove
        intIDDel = int(input('Which ID would you like to delete? ').strip())

        # Search thru table and delete CD
        intRowNr = -1
        blnCDRemoved = False
        for row in lstTbl:
            intRowNr += 1
            if row['ID'] == intIDDel:
                del lstTbl[intRowNr]
                blnCDRemoved = True
                break
        if blnCDRemoved:
            print('The CD was removed')
        else:
            print('Could not find this CD!')

class FileProcessor:
    """Processing the data to and from text file"""

    @staticmethod
    def read_file(file_name, table):
        """Function to manage data ingestion from file to a list of dictionaries

        Reads the data from file identified by file_name into a 2D table
        (list of dicts) table one line in the file represents one dictionary row in table.

        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 runtim
e

        Returns:
            None.
        """
        # Load exisitng data from file
        table.clear()  # this clears existing data and allows to load data from file
        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()

    @staticmethod
    def write_file(file_name, table):
        """Function to save data to a file

        Writes the data to file identified by file_name into a 2D table
              (list of dicts) table one line in the file represents one dictionary row in table.

              Args:
                  file_name(string): name of file used to write the data to
                  table(list of dict): 2D data structure (list of dicts) that hold the data during r
    untime

              Returns:
                  None
              """
              # Save data to a file
              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('Menu\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 sting 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 current inventory table
```

```
155.
156.
157.                 Args:
158.                     table (list of dict): 2D data structure (list of dicts) that holds the data during
    runtime.
159.
160.                 Returns:
161.                     None.
162.                 """
163.                 # Display current inventory
164.                 print('======= The Current Inventory: =======')
165.                 print('ID\tCD Title (by: Artist)\n')
166.                 for row in table:
167.                     print('{}\t{} (by:{})'.format(*row.values()))
168.                 print('======================================')
169.
170.             @staticmethod
171.             def get_userInput():
172.                 """Function to get user input for ID, CD title, and CD artist
173.
174.                 Args:
175.                     None.
176.
177.                 Returns:
178.                     StrID (string): Input for CD ID.
179.                     Strtitle (string): Input for CD Title.
180.                     StArtist (string): Input for CD Artist.
181.                 """
182.                 # Ask user for new ID, CD Title and Artist
183.                 strID = input('Enter ID: ').strip()
184.                 strTitle = input('What is the CD\'s title? ').strip()
185.                 stArtist = input('What is the Artist\'s name? ').strip()
186.                 return strID, strTitle, stArtist
187.
188.         # If file does not exist in currrent folder, Create one
189.         if not os.path.exists(strFileName):
190.             newfile = open(strFileName, 'a')
191.             newfile.close()
192.
193.         # When program starts, read in the currently saved Inventory
194.         FileProcessor.read_file(strFileName, lstTbl)
195.
196.         # Start main loop
197.         while True:
198.             # Display Menu to user and get choice
199.             IO.print_menu()
200.             strChoice = IO.menu_choice()
201.
202.         # Process menu selection
203.             # Process exit first
204.             if strChoice == 'x':
205.                 break
206.
207.             # Process load inventory
208.             if strChoice == 'l':
209.                 print('WARNING: If you continue, all unsaved data will be lost and the Inventory re-
    loaded from file.')
210.                 strYesNo = input('type \'yes\' to continue and reload from file. otherwise reload will
     be canceled: ')
211.                 if strYesNo.lower() == 'yes':
212.                     print('reloading...')
213.                     FileProcessor.read_file(strFileName, lstTbl)
214.                     IO.show_inventory(lstTbl) # Display Inventory to user
215.                 else:
216.                     input('canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the
    menu.')
217.                     IO.show_inventory(lstTbl)
218.                 continue  # start loop back at top.
```

```
219.
220.           # Process add a CD
221.           elif strChoice == 'a':
222.               # Store user inputs
223.               userInputId, userInputTitle, userInputArtist = IO.get_userInput()
224.               # Adds data to the 2D table (list of dictionaries)
225.               dicRow = DataProcessor.add_data(userInputId, userInputTitle, userInputArtist, lstTbl)
226.               IO.show_inventory(lstTbl)
227.               continue  # start loop back at top.
228.
229.           # Process display current inventory
230.           elif strChoice == 'i':
231.               IO.show_inventory(lstTbl)
232.               continue  # start loop back at top.
233.
234.           # Process delete a CD
235.           elif strChoice == 'd':
236.               IO.show_inventory(lstTbl)
237.               DataProcessor.del_data() # Deletes data from inventory
238.               IO.show_inventory(lstTbl)
239.               continue  # start loop back at top.
240.
241.           # Process save inventory to file
242.           elif strChoice == 's':
243.               # Display current inventory and ask user for confirmation to save
244.               IO.show_inventory(lstTbl)
245.               strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
246.               # Process choice
247.               if strYesNo == 'y':
248.                   FileProcessor.write_file(strFileName, lstTbl) # Calling write_file function
249.               else:
250.                   input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
251.               continue  # start loop back at top.
252.
253.           # Catch-all should not be possible, as user choice gets vetted in IO, but to be save:
254.           else:
255.               print('General Error')
```