<Sayali Subodh Shinde>

<March 6 2022>

<Foundations of Programming (Python)>

<Assignment 06>

To add functions to classes created in the program and call them appropriately to achieve the Program functionality

Introduction

Brief overview about this module is I learnt in detail about the functions. The variables passed to functions are called arguments/parameters. The function can return values which can be one or many. One can assign the function return values to variables/lists/dictionaries. I also learnt about local and global variables and how printing values before assigning values throws a Python error. I also leant about positional arguments.

I also leant about the default parameter values set in function definition and how it can be used. The return type can also be checked in functions. I also learnt how to create docstring for the functions. I have briefly understood classes which holds different functions and can be accessed as class. Functions(arguments).I also leant about the function overloading.

This Module also describes working with Github and creating your code for different developers to review and work on.

Menu selection based on users' choice and implementation using functions and classes

The aim of this assignment is to add functions like write to the file in the class FileProcessor and add functions like delete and another function to add data to table in memory in class DataProcessor and to add function to take inputs about the new CD Inventory in the function add_inventory in the class IO.

1. Add functions like write to the file in the class FileProcessor
2. Add functions like delete and another function to add data to table in memory in class DataProcessor
3. Add function to take inputs about the new CD Inventory in the function add_inventory in the class IO

   As this data to be in Memory and not lose its content, it should be in while loop, to execute the data contents like displaying or saving in txt file and call the appropriate functions as the user selects the choice.

GIST:

In the While loop True all time as its condition is true, the menu will be displayed asking for users to choose. Based on what to select from below like **a** to add CD data**,** **i** to display current CD data**,** **s** to save Cd data , and **d** to delete the dict in the list that the user wants,and **l** to load the data from a text file into the memory and **x** to exit with the help of functions and classes. Here the imp point to note is lstbl which is going to hold all individual cd dict is initialized outside while loop to empty list. If this is in the while loop being empty at every iteration of add we would get a extra empty list appended, so it must be out of while.

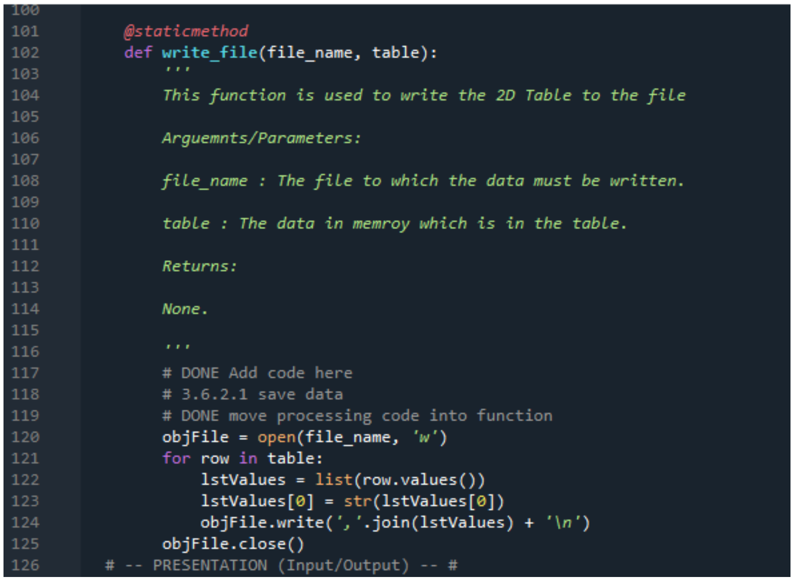
The important point is here there are 3 different classes serving the purpose of data processing in the memory (adding newly entered data by the user to the 2D table, deleting the data from the memory if the user wishes and has entered the ID). Another class for IO operations like displaying the menu helping chose from the menu, displaying the current inventory, getting inputs about new inventory (like ID, Title, Artist). The last class is for File Processing like reading from the file and writing to the file. The while loop has the options from memory iterated with help of continue and break if x option is chosen.

The functions are called in the respective chosen choices with the arguments as defined in the function calls. The Program is in SOC (separation of concerns), with data, processing and presentation skills as mentioned.

Add the write function in the class File Processor

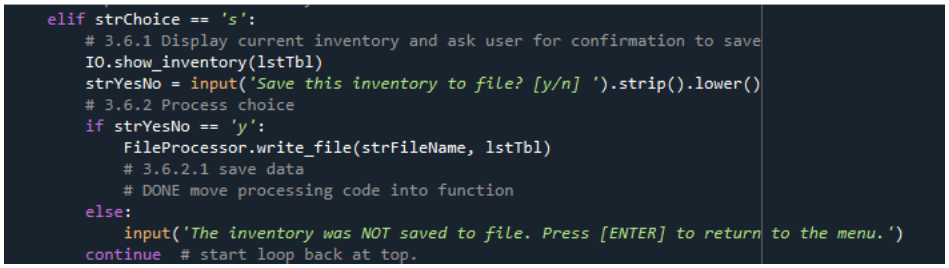
Now we need to save the data to the file CdInventory.txt if user chose option s. To save data I have created a function called write_file with arguments file_name and table and returns None. I have moved this code from the write in while to the newly created function.

Here the choice of writing is suitable rather than appending to a file as we will add redundant data and can have duplicates.



```
    @staticmethod
    def write_file(file_name, table):
        '''
        This function is used to write the 2D Table to the file

        Arguemnts/Parameters:

        file_name : The file to which the data must be written.

        table : The data in memroy which is in the table.

        Returns:

        None.

        '''
        # DONE Add code here
        # 3.6.2.1 save data
        # DONE move processing code into function
        objFile = open(file_name, 'w')
        for row in table:
            lstValues = list(row.values())
            lstValues[0] = str(lstValues[0])
            objFile.write(','.join(lstValues) + '\n')
        objFile.close()
  # -- PRESENTATION (Input/Output) -- #
```

Figure 1 Assignment06 Screenshot of write_function in class_FileProcessor

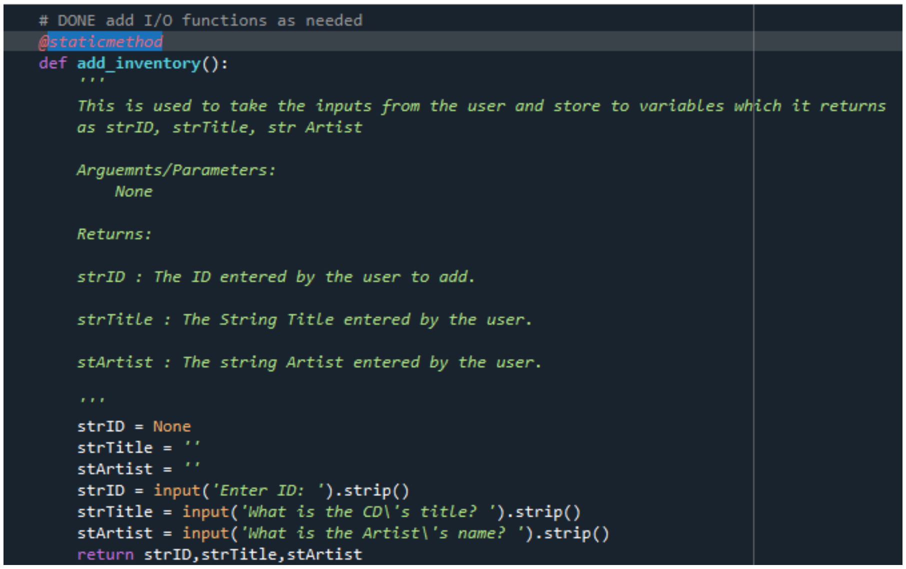


```
    elif strChoice == 's':
        # 3.6.1 Display current inventory and ask user for confirmation to save
        IO.show_inventory(lstTbl)
        strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
        # 3.6.2 Process choice
        if strYesNo == 'y':
            FileProcessor.write_file(strFileName, lstTbl)
            # 3.6.2.1 save data
            # DONE move processing code into function
        else:
            input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
        continue  # start loop back at top.
```

Figure 2 Assignment 06 Screenshot of function call in savein the while loop

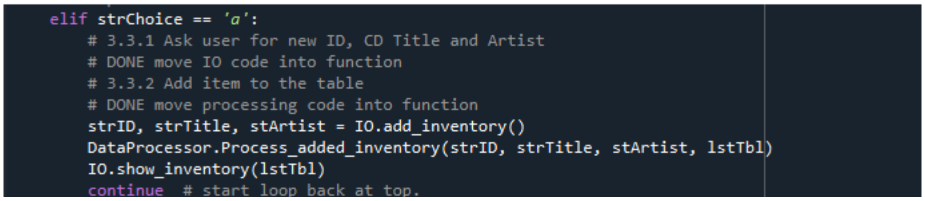
Add Data to the Table when user has data to add CD Inventory in the Class IO

Now to add the data to the list we need to take the required inputs from the user like ID, CD Title and Artist Name.in the IO.add_inventory() function, which returns stride,strTitle, stArtist.I have also defined this stride,strTitle, stArtist as local variables with None and empty strings. This function should be called when the choice is ‘a’; hence this function is called there in Figure 4. The code in the function definition is moved from the while loop



```
    # DONE add I/O functions as needed
    @staticmethod
    def add_inventory():
        '''
        This is used to take the inputs from the user and store to variables which it returns
        as strID, strTitle, str Artist

        Arguemnts/Parameters:
            None

        Returns:

        strID : The ID entered by the user to add.

        strTitle : The String Title entered by the user.

        stArtist : The string Artist entered by the user.

        '''
        strID = None
        strTitle = ''
        stArtist = ''
        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 3. Assignment 06 Screenshot of the fucntion add_inventory in IO Class

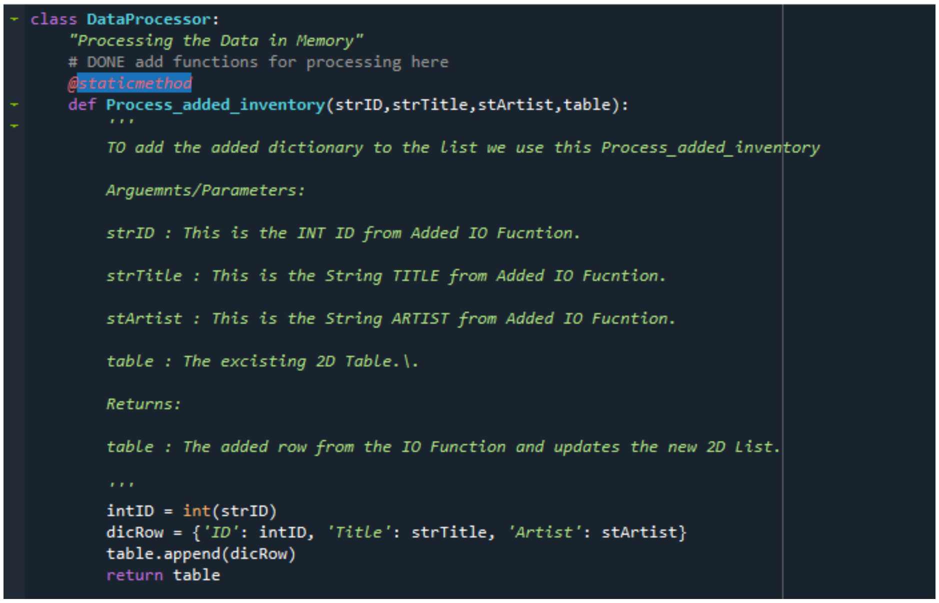


```
    elif strChoice == 'a':
        # 3.3.1 Ask user for new ID, CD Title and Artist
        # DONE move IO code into function
        # 3.3.2 Add item to the table
        # DONE move processing code into function
        strID, strTitle, stArtist = IO.add_inventory()
        DataProcessor.Process_added_inventory(strID, strTitle, stArtist, lstTbl)
        IO.show_inventory(lstTbl)
        continue  # start loop back at top.
```

Figure 4. Assignment 06 Screenshot of the function call in 'a' choice of while loop

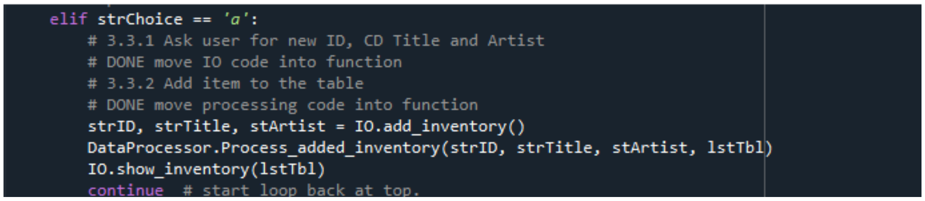
Add function in class DataProcessor to add the newly input data the ID, Title and Artist from the user in the IO class so that the up-to-date table data is in memory

Now.to process that newly given users inputs for the CD row to be added to the table in the memory I have created the function called Process_added_inventory() in the class DataProcessor with arguments like strID, strTitle and strArtist which is the return values from the add_inventory in the IO Class and original list 2D of the dictionaries. This function returns the table with the added row. So when in while if user chooses option a, first the IO class add_inventory function is called and the input values are returned which are used in the Process_added_inventory in the Data Processor class and the table is returned which is as shown in Fig 6.The code in the function definition is moved from the while loop.



```
class DataProcessor:
    "Processing the Data in Memory"
    # DONE add functions for processing here
    @staticmethod
    def Process_added_inventory(strID,strTitle,stArtist,table):
        '''
        TO add the added dictionary to the list we use this Process_added_inventory

        Arguemnts/Parameters:

        strID : This is the INT ID from Added IO Fucntion.

        strTitle : This is the String TITLE from Added IO Fucntion.

        stArtist : This is the String ARTIST from Added IO Fucntion.

        table : The excisting 2D Table.\.

        Returns:

        table : The added row from the IO Function and updates the new 2D List.

        '''
        intID = int(strID)
        dicRow = {'ID': intID, 'Title': strTitle, 'Artist': stArtist}
        table.append(dicRow)
        return table
```

Figure 5 Assignemnt06 Process_added Inventory function in class Data Processor Screenshot



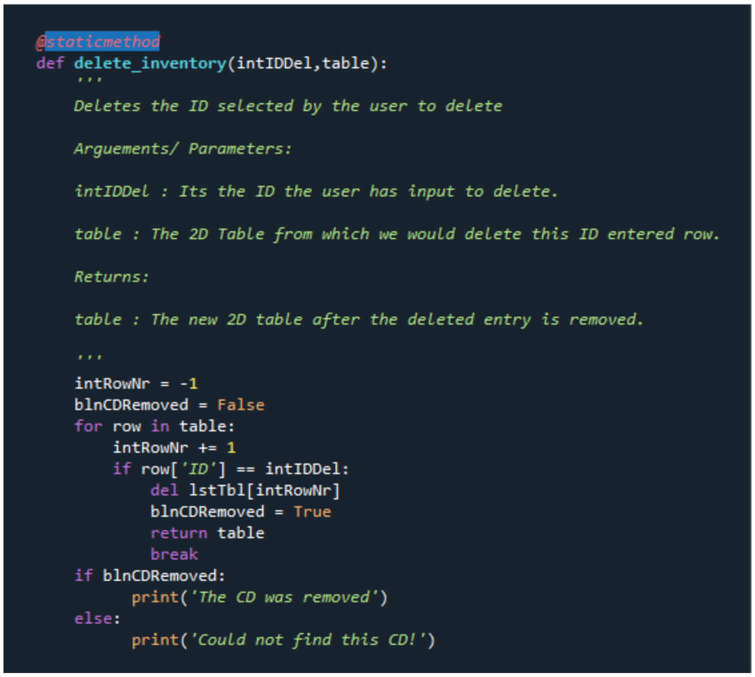
```
    elif strChoice == 'a':
        # 3.3.1 Ask user for new ID, CD Title and Artist
        # DONE move IO code into function
        # 3.3.2 Add item to the table
        # DONE move processing code into function
        strID, strTitle, stArtist = IO.add_inventory()
        DataProcessor.Process_added_inventory(strID, strTitle, stArtist, lstTbl)
        IO.show_inventory(lstTbl)
        continue  # start loop back at top.
```

Figure 6. Assignemnt06 Screenshot of option a in while loop for function calls

Add Function to delete the selected ID from the Table in the DataProcessor Class

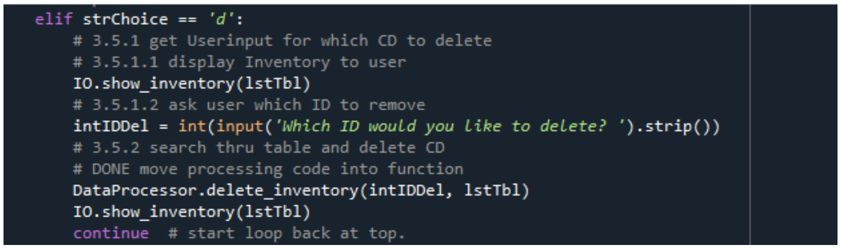
Now to delete inventory if user selects I have created the delete_inventory() function in the class DataProcessor. I have moved the code from while loop and changed lsttbl to table.This function has arguments table and intIDDel which is an user input asked in the while loop to know which ID needs to be deleted.

.The function calls in the while section of the if condition 'd', where previous inventory is shown ID to delete is asked from the user and then delete_inventory_function is called and then again after deleting the show inventory function is called which shows the deleted entry was removed in the Fig8.



```
@staticmethod
def delete_inventory(intIDDel,table):
    '''
    Deletes the ID selected by the user to delete

    Arguements/ Parameters:

    intIDDel : Its the ID the user has input to delete.

    table : The 2D Table from which we would delete this ID entered row.

    Returns:

    table : The new 2D table after the deleted entry is removed.

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

Figure 7 Assignment06 Screenshot of the delete_inventory function

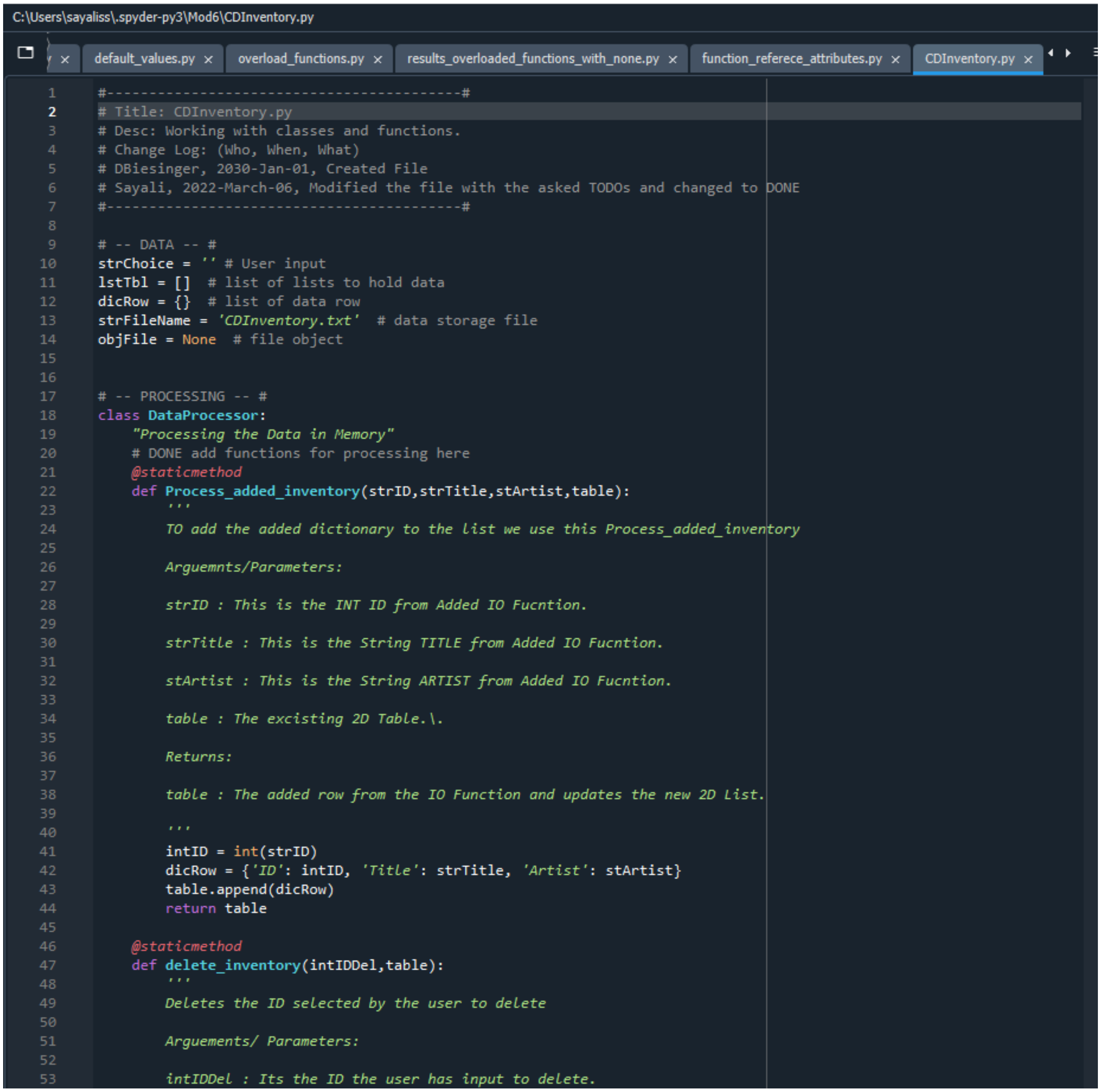


```
elif strChoice == 'd':
    # 3.5.1 get Userinput for which CD to delete
    # 3.5.1.1 display Inventory to user
    IO.show_inventory(lstTbl)
    # 3.5.1.2 ask user which ID to remove
    intIDDel = int(input('Which ID would you like to delete? ').strip())
    # 3.5.2 search thru table and delete CD
    # DONE move processing code into function
    DataProcessor.delete_inventory(intIDDel, lstTbl)
    IO.show_inventory(lstTbl)
    continue  # start loop back at top.
```

Figure 8 Assignemnt06 Screenshot of the delete_inventory function call

Script

Below is the script from Spyder the consolidated and filled in for Assignemnt06_starter.py filled in for the required asks and renamed to CDInventory.py,as the script is huge I couldn't fit all lines.

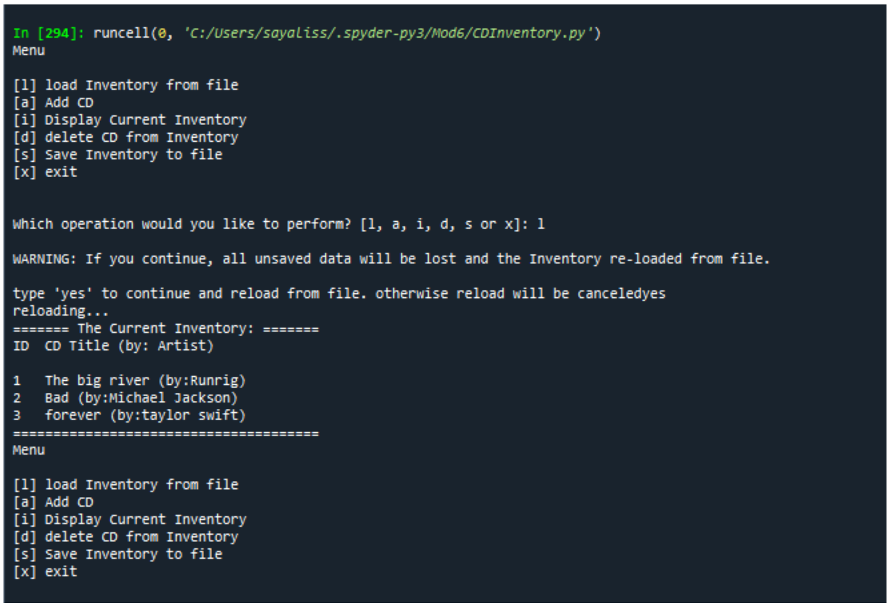

C:\Users\sayaliss\.spyder-py3\Mod6\CDInventory.py
default_values.py
overload_functions.py
results_overloaded_functions_with_none.py
function_referece_attributes.py
CDInventory.py

```python
#------------------------------------------#
# Title: CDInventory.py
# Desc: Working with classes and functions.
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# Sayali, 2022-March-06, Modified the file with the asked TODOs and changed to DONE
#------------------------------------------#

# -- 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 in Memory"
    # DONE add functions for processing here
    @staticmethod
    def Process_added_inventory(strID,strTitle,stArtist,table):
        '''
        TO add the added dictionary to the list we use this Process_added_inventory

        Arguemnts/Parameters:

        strID : This is the INT ID from Added IO Fucntion.

        strTitle : This is the String TITLE from Added IO Fucntion.

        stArtist : This is the String ARTIST from Added IO Fucntion.

        table : The excisting 2D Table.\.

        Returns:

        table : The added row from the IO Function and updates the new 2D List.

        '''
        intID = int(strID)
        dicRow = {'ID': intID, 'Title': strTitle, 'Artist': stArtist}
        table.append(dicRow)
        return table

    @staticmethod
    def delete_inventory(intIDDel,table):
        '''
        Deletes the ID selected by the user to delete

        Arguements/ Parameters:

        intIDDel : Its the ID the user has input to delete.
```

Figure 9 Assignment 06 Screenshot of the Spyder Program

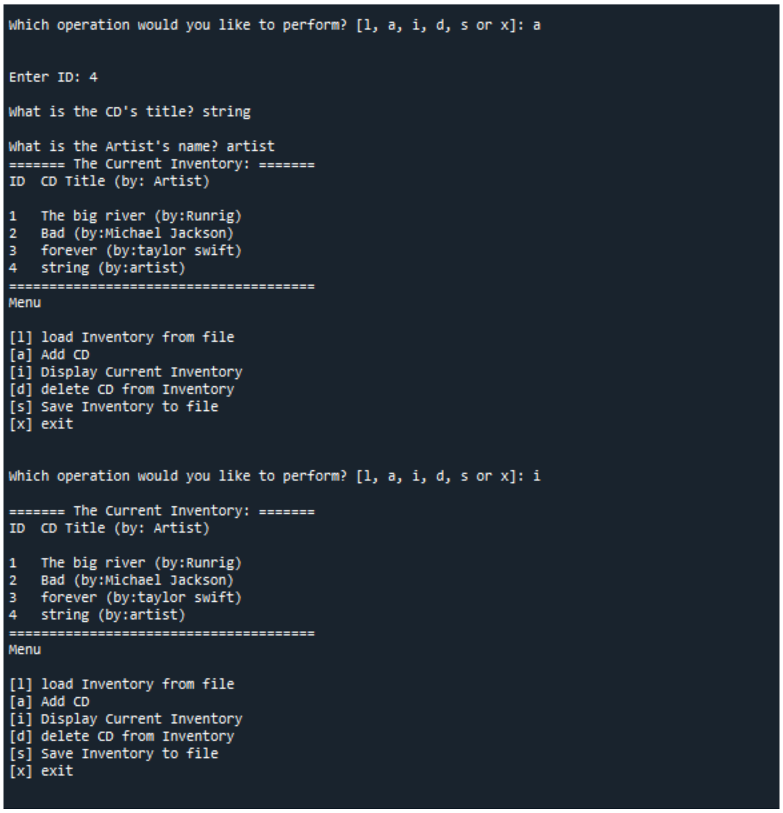
Execution of Program

As requested in the assignemnt06 I have executed the script in Spyder and in Command Prompt. I have also excluded snippets of display the current inventory and if the user has entered any other menu option apart from the menu given to the user.



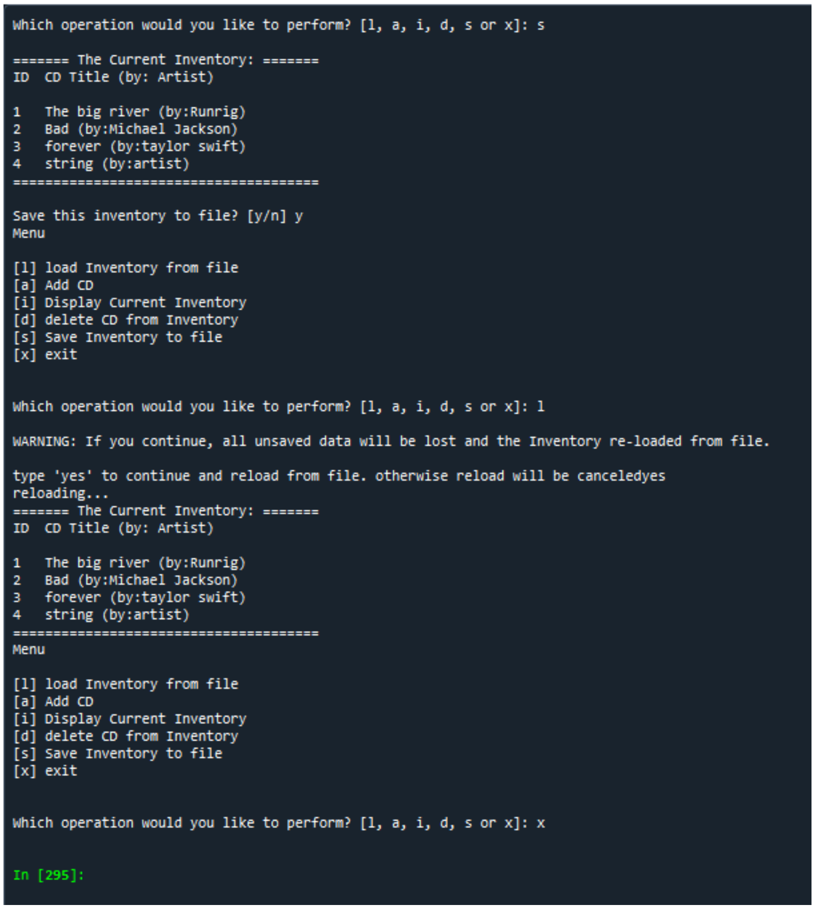
```
In [294]: runcell(0, 'C:/Users/sayaliss/.spyder-py3/Mod6/CDInventory.py')
Menu

[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]: 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 canceledyes
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   The big river (by:Runrig)
2   Bad (by:Michael Jackson)
3   forever (by:taylor swift)
======================================
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 10 Assignemnt06 Screenshot of execution in Spyder of loading data operation from file



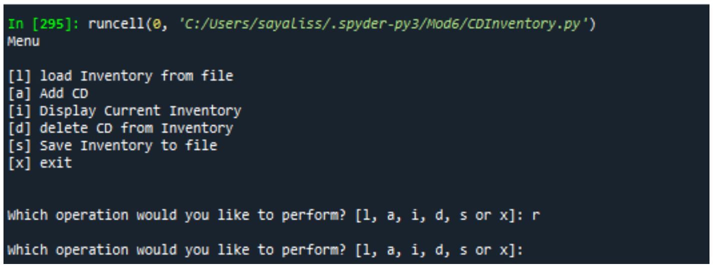
```
Which operation would you like to perform? [l, a, i, d, s or x]: a


Enter ID: 4

What is the CD's title? string

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

1   The big river (by:Runrig)
2   Bad (by:Michael Jackson)
3   forever (by:taylor swift)
4   string (by:artist)
======================================
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   The big river (by:Runrig)
2   Bad (by:Michael Jackson)
3   forever (by:taylor swift)
4   string (by:artist)
======================================
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 11 Assignment06 Screenshot of Execution in Spyder of adding data to the inventory and displaying it



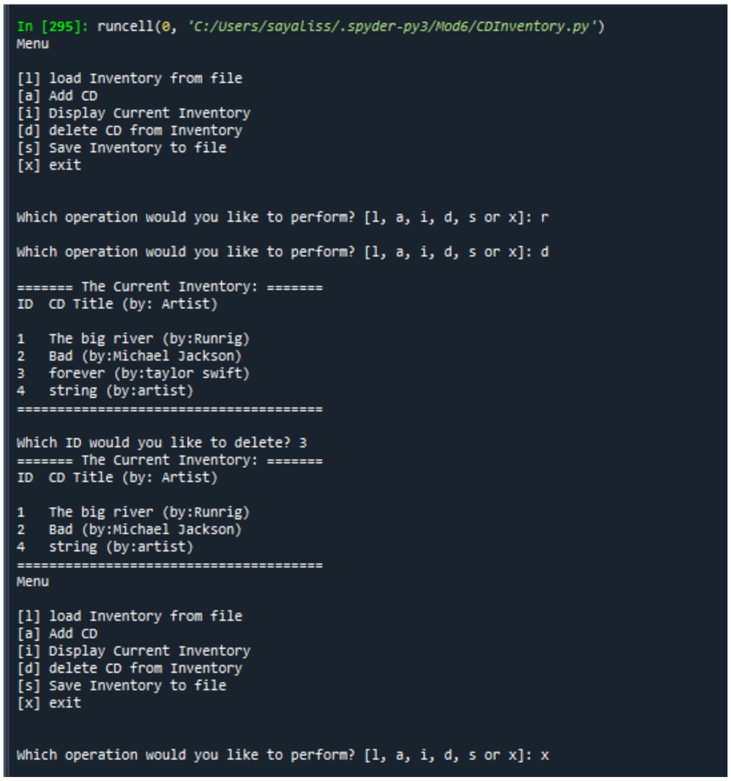
```
Which operation would you like to perform? [l, a, i, d, s or x]: s

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

1   The big river (by:Runrig)
2   Bad (by:Michael Jackson)
3   forever (by:taylor swift)
4   string (by:artist)
======================================

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]: 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 canceledyes
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   The big river (by:Runrig)
2   Bad (by:Michael Jackson)
3   forever (by:taylor swift)
4   string (by:artist)
======================================
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 [295]:
```

Figure 12. Assignment 06 Screenshot of Execution in Spyder of saving this added data in filer and then loading the data from file and exiting



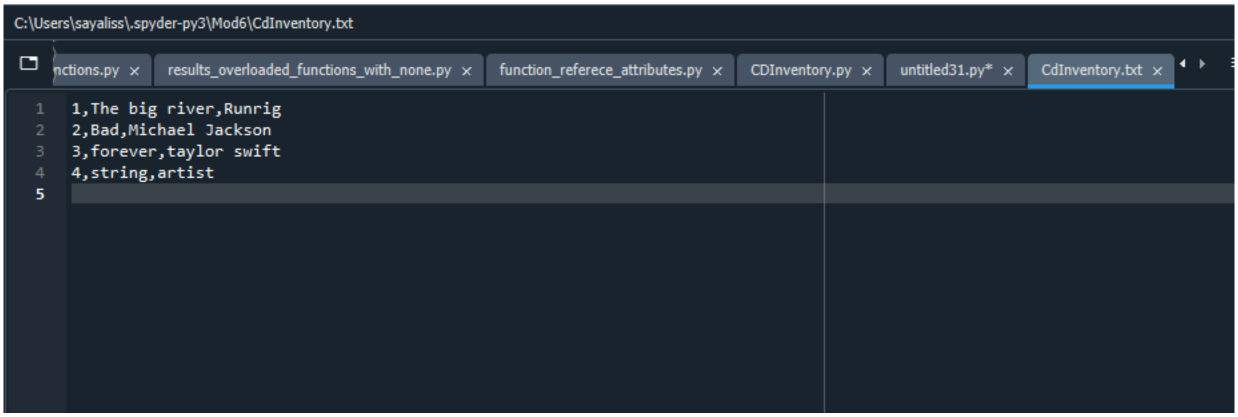
```
In [295]: runcell(0, 'C:/Users/sayaliss/.spyder-py3/Mod6/CDInventory.py')
Menu

[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]: r

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

Figure 13. Assignemnt06 Screenshot of Execution in Spyder of the entering any value other than the Menu



```
In [295]: runcell(0, 'C:/Users/sayaliss/.spyder-py3/Mod6/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]: r

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 river (by:Runrig)
2   Bad (by:Michael Jackson)
3   forever (by:taylor swift)
4   string (by:artist)
======================================

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

1   The big river (by:Runrig)
2   Bad (by:Michael Jackson)
4   string (by:artist)
======================================
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
```

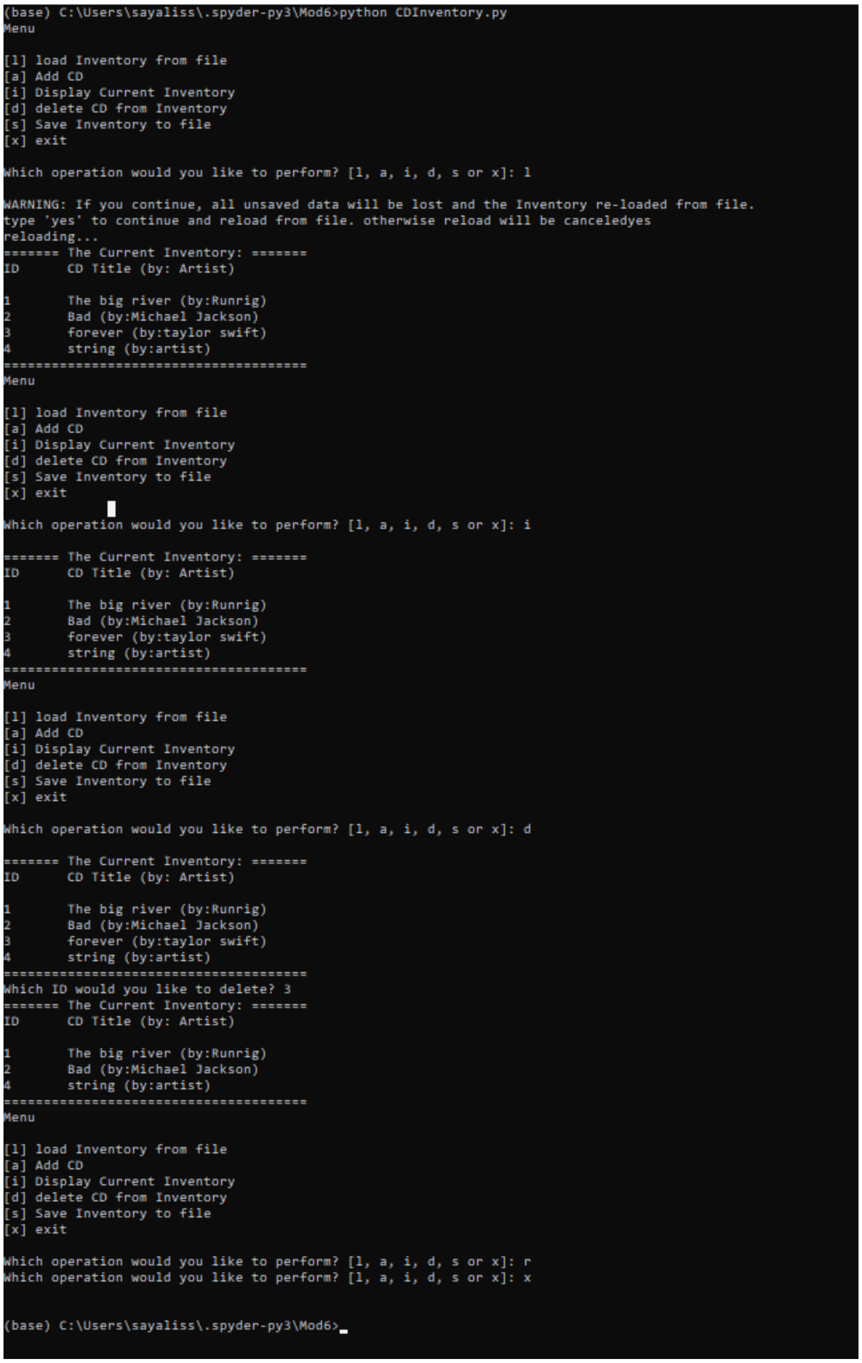
Figure 14 Assignemnt06 Screenshot of Execution in Spyder of the delete function



C:\Users\sayaliss\.spyder-py3\Mod6\CdInventory.txt
nctions.py ×
results_overloaded_functions_with_none.py ×
function_referece_attributes.py ×
CDInventory.py ×
untitled31.py* ×
CdInventory.txt ×

```
1,The big river,Runrig
2,Bad,Michael Jackson
3,forever,taylor swift
4,string,artist

```

Figure 15 Assignemnt06 Screenshot of the file CDInventory.txt file with added values saved



```
(base) C:\Users\sayaliss\.spyder-py3\Mod6>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]: 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 canceledyes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The big river (by:Runrig)
2       Bad (by:Michael Jackson)
3       forever (by:taylor swift)
4       string (by:artist)
======================================
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       The big river (by:Runrig)
2       Bad (by:Michael Jackson)
3       forever (by:taylor swift)
4       string (by:artist)
======================================
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 river (by:Runrig)
2       Bad (by:Michael Jackson)
3       forever (by:taylor swift)
4       string (by:artist)
======================================
Which ID would you like to delete? 3
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The big river (by:Runrig)
2       Bad (by:Michael Jackson)
4       string (by:artist)
======================================
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]: r
Which operation would you like to perform? [l, a, i, d, s or x]: x


(base) C:\Users\sayaliss\.spyder-py3\Mod6>
```

Figure 16 Assignemnt06 Execution of the code for all functions on the Command Prompt

I have used Syntax Highlighters (External Reference) 1webpage, to standardize and it displays text, especially script, in different colors and fonts according to the Language.

Summary

I have learnt use of function definition and calling it.Also the arguments play an important role for the function use and returns values which are used as outputs. I also learnt class definition and class access of functions.

I have uploaded the Gitlab code : https://github.com/sayalisu/Assignment_062

Retrieved 2022-Feb-04

2 Retrieved 2022-Feb-27

Appendix

Script

```
#------------------------------------------#
# Title: CDInventory.py
# Desc: Working with classes and functions.
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# Sayali, 2022-March-06, Modified the file with the asked TODOs and changed to DONE
#------------------------------------------#

# -- 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 in Memory"
    # DONE add functions for processing here
    @staticmethod
    def Process_added_inventory(strID,strTitle,stArtist,table):
        '''
        TO add the added dictionary to the list we use this Process_added_inventory

        Arguemnts/Parameters:

        strID : This is the INT ID from Added IO Fucntion.

        strTitle : This is the String TITLE from Added IO Fucntion.

        stArtist : This is the String ARTIST from Added IO Fucntion.

        table : The excisting 2D Table.\.

        Returns:

        table : The added row from the IO Function and updates the new 2D List.

        '''
        intID = int(strID)
        dicRow = {'ID': intID, 'Title': strTitle, 'Artist': stArtist}
        table.append(dicRow)
        return table

    @staticmethod
    def delete_inventory(intIDDel,table):
        '''
        Deletes the ID selected by the user to delete

        Arguements/ Parameters:

        intIDDel : Its the ID the user has input to delete.

        table : The 2D Table from which we would delete this ID entered row.

        Returns:

        table : The new 2D table after the deleted entry is removed.

        '''
        intRowNr = -1
        blnCDRemoved = False
        for row in table:
            intRowNr += 1
            if row['ID'] == intIDDel:
                del lstTbl[intRowNr]
                blnCDRemoved = True
                return table
                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 runtime

        Returns:
            None.
        """
        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):
        '''
        This function is used to write the 2D Table to the file

        Arguemnts/Parameters:

        file_name : The file to which the data must be written.

        table : The data in memroy which is in the table.

        Returns:

        None.

        '''
        # DONE Add code here
        # 3.6.2.1 save data
        # DONE move processing code into function
        objFile = open(file_name, 'w')
        for row in table:
            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


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

        Returns:
            None.

        """
        print('======= The Current Inventory: =======')
```

```python
        print('ID\tCD Title (by: Artist)\n')
        for row in table:
            print('{}\t{} (by:{})'.format(*row.values()))
        print('======================================')

    # DONE add I/O functions as needed
    @staticmethod
    def add_inventory():
        '''
        This is used to take the inputs from the user and store to variables which it returns
        as strID, strTitle, str Artist

        Arguemnts/Parameters:
            None

        Returns:

        strID : The ID entered by the user to add.

        strTitle : The String Title entered by the user.

        stArtist : The string Artist entered by the user.

        '''
        strID = None
        strTitle = ''
        stArtist = ''
        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
# 1. When program starts, read in the currently saved Inventory
FileProcessor.read_file(strFileName, lstTbl)

# 2. start main loop
while True:
    # 2.1 Display Menu to user and get choice
    IO.print_menu()
    strChoice = IO.menu_choice()

    # 3. Process menu selection
    # 3.1 process exit first
    if strChoice == 'x':
        break
    # 3.2 process load inventory
    if strChoice == 'l':
        print('WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.')
        strYesNo = input('type \'yes\' to continue and reload from file. otherwise reload will be canceled')
        if strYesNo.lower() == 'yes':
            print('reloading...')
            FileProcessor.read_file(strFileName, lstTbl)
            IO.show_inventory(lstTbl)
        else:
            input('canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')
            IO.show_inventory(lstTbl)
        continue  # start loop back at top.
    # 3.3 process add a CD
    elif strChoice == 'a':
        # 3.3.1 Ask user for new ID, CD Title and Artist
        # DONE move IO code into function
        # 3.3.2 Add item to the table
        # DONE move processing code into function
        strID, strTitle, stArtist = IO.add_inventory()
        DataProcessor.Process_added_inventory(strID, strTitle, stArtist, lstTbl)
        IO.show_inventory(lstTbl)
        continue  # start loop back at top.
    # 3.4 process display current inventory
    elif strChoice == 'i':
        IO.show_inventory(lstTbl)
        continue  # start loop back at top.
    # 3.5 process delete a CD
    elif strChoice == 'd':
        # 3.5.1 get Userinput for which CD to delete
        # 3.5.1.1 display Inventory to user
        IO.show_inventory(lstTbl)
        # 3.5.1.2 ask user which ID to remove
        intIDDel = int(input('Which ID would you like to delete? ').strip())
        # 3.5.2 search thru table and delete CD
        # DONE move processing code into function
        DataProcessor.delete_inventory(intIDDel, lstTbl)
        IO.show_inventory(lstTbl)
        continue  # start loop back at top.
    # 3.6 process save inventory to file
    elif strChoice == 's':
        # 3.6.1 Display current inventory and ask user for confirmation to save
        IO.show_inventory(lstTbl)
        strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
        # 3.6.2 Process choice
        if strYesNo == 'y':
            FileProcessor.write_file(strFileName, lstTbl)
            # 3.6.2.1 save data
            # DONE move processing code into function
        else:
            input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
        continue  # start loop back at top.
    # 3.7 catch-all should not be possible, as user choice gets vetted in IO, but to be save:
    else:
        print('General Error')
```