<Sayali Subodh Shinde>

<March 20 2022>

<Foundations of Programming (Python)>

<Assignment 08>

To add properties and methods to Class CD for Object Instantiation and then use Class IO and class FileIO to display the functionality as user chooses

Introduction

Brief overview about this module is I learnt in detail about the OOP like class and its parameters like fields, attributes, methods, __str__ method of class. I also learnt the difference in using the self and @static_method. I also understood the objects use class functions and attributes, so class is like template/blueprint for objects to use where self is on the object. One important parameter I learnt is that the constructor __init__ is called when the object is instantized.
I also understood the use of properties like getter and setter where setter is assignment and getter is for formatting and both are class properties and is a good practice to create a getter and setter for each attribute and this are abstraction of gives protection to the objects for their attributes. I also understood that for printing we should use the __str__method, like object.__str__().

Aim of Assignment

The aim of this assignment is to create a list of objects. So a class with the CD attributes like ID, Title, Artist needs to be defined. Then other functionality like Menu, choice, display and add CDs needs to be added in class IO The other part is to create the class FileIO for us to save this list of objects in the cdinventory.txt file and read the file. The second part of the assignment deals with the Error Handling incase of ID not an integer and if the file doesn't exist.

GIST:

The important point is here there are 3 different classes serving the purpose of data processing. In the class CD we are using the constructor to create the attributes like ID, Title and Artist for objects to be instantized using that attributes. Inside class IO there is method add_list_of_obj() for appending all data to the lstofCDObjects and also for printing the data there is a method called __str__. 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 IO 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

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 binary 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 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 and the error handling is also taken care of.

Class CD

Here I have created the properties for ID, Title and Artist as asked and a constructor when each object is created. I have added 2 functions one to add the CDinfo from the user as we get to append to a list add_list_of_obj which would be cd objects and the second function is of the str dunder function to add for printing the list

```
class CD:
    """Stores data about a CD:

    properties:
        cd_id: (int) with CD ID
        cd_title: (string) with the title of the CD
        cd_artist: (string) with the artist of the CD
    methods:

    """
    # DONE Add Code to the CD class
    def __init__(self,CD_ID,CD_TITLE,CD_ARTIST):
        self.__cd_id = CD_ID
        self.__cd_title = CD_TITLE
        self.__cd_artist = CD_ARTIST

    @property
    def cd_id(self):
        return int(self.__cd_id)
    @cd_id.setter
    def cd_id(self,ID):
        if type(ID) == 'int':
            self.__cd_id = ID
        else:
           raise Exception ('Not an integer')

    @property
    def cd_title(self):
        return self.__cd_title
    @cd_title.setter
    def cd_title(self,Title):
        self.__cd_title = Title

    @property
    def cd_artist(self):
        return self.__cd_artist
    @cd_artist.setter
    def cd_artist(self,Artist):
            self.__cd_artist = Artist

    #methods
    def add_list_of_obj(cdObj):
          lstOfCDObjects.append(cdObj)
          return lstOfCDObjects
    def __str__(self):
        str_new = ''
        str_new = str_new +  str(self.__cd_id) + ', ' + self.__cd_title + ', ' + self.__cd_artist +'\n'
        return str_new
```

Figure 1. Assignment 08 Screenshot of Class CD with error handling

Class FileIO

I have added the functions as save_inventory and load_inventory and for save inventory I have used the object row to access the attributes and write to the file. For load inventory file I have split the string to the list and then appended the list of objects by calling CD class on Cd info object. I have also done error handling in this part to check if the file wasn't present or if it was present but empty.

```python
# -- PROCESSING -- #
class FileIO:
    """Processes data to and from file:

    properties:

    methods:
        save_inventory(file_name, lst_Inventory): -> None
        load_inventory(file_name): -> (a list of CD objects)

    """
    # DONE Add code to process data from a file
    # DONE Add code to process data to a file
    @staticmethod
    def save_inventory(file_name, lst_Inventory):

            file1 =open(file_name,'w')
            for row in lst_Inventory:
                 str1 = str(row.cd_id) + ', ' + row.cd_title + ', ' + row.cd_artist + '\n'
                 file1.write(str1)
            file1.close()


    @staticmethod
    def load_inventory(file_name):
        lstOfCDObjects.clear()
        try :
            file1 =open(file_name,'r')
            for row in file1:
                data = row.strip().split(',')
                CdInfo = CD(int(data[0]),data[1],data[2])
                lstOfCDObjects.append(CdInfo)
            file1.close()
        except FileNotFoundError:
            print("Filenot_found,please make sure file is present")
        except EOFError:
                print("File has no contents.Please add data to the file and then read")
        except ValueError:
                        print("Check the values")
        return lstOfCDObjects
```

Figure 2. Assignment08 Screenshot of Class FileIO with functions added

Class IO

In Class IO I have added the doc string and the functions to print menu, choice of user, display the current inventory and add inventory to get inputs from the user. In show inventory I had to call the __str__ on the row as we are printing. In the add_inventory function I had to create object with the attributes and I return that object.

```python
# -- PRESENTATION (Input/Output) -- #
class IO:
    # Done add docstring
    # DONE add code to show menu to user
    # DONE add code to captures user's choice
    # DONE add code to display the current data on screen
    # DONE add code to get CD data from user
    """Handling Input / Output

    properties:

    methods:
        print_menu(): -> None
        menu_choice(): -> (choice user has selcted from the menu)
        show_inventory(): -> (Shows the current inventory )
        add_inventory(): -> (CDObject -> Object of class CD with all attributes.)

    """
```

Figure 3 Assignment08 Screenshot of the Class IO Docstring

```python
    @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: =======')
        print('ID\tCD Title Artist\n')
        for row in table:
              print(row.__str__())
        print('======================================')
    @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:CDObject -> Object of class CD with all attributes.

        '''
        intID = None
        strTitle = ''
        stArtist = ''
        while type(intID) != int:
            try:
                intID = int(input('Enter ID: ').strip())
            except ValueError :
                print("This ID is not integer type , please enter integer ")

        strTitle = input('What is the CD\'s title? ').strip()
        stArtist = input('What is the Artist\'s name? ').strip()
        CdObject = CD (intID, strTitle, stArtist)
        return  CdObject
```

Figure 4 Assignment08 Screenshot of the method add_inventory and show_inventory

Script

Below is the script from Spyder the consolidated and filled in for Assignemnt08 filled in for the required asks and renamed to CD_Inventory.py,as the script is huge I couldn't fit all lines.

C:\Users\sayaliss\.spyder-py3\Mod8\CD_Inventory.py

listing3.py* × | poll.py × | CDInventory.py - Mod6 × | untitled14.py* × | CDInventory.py - Mod7 × | CD_Inventory.py × | cdInventory.txt* × | CdInventory.txt ×

```python
#-----------------------------------------#
# Title: CD_inventory.py
# Desc: CD_inventory - Working with classes
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, created file
# DBiesinger, 2030-Jan-01, added pseudocode to complete assignment 08
# Sayali, 2022-March-20, completed the TODOs
#-----------------------------------------#

# -- DATA -- #
strFileName = 'cdInventory.txt'
lstOfCDObjects = []

class CD:
    """Stores data about a CD:

    properties:
        cd_id: (int) with CD ID
        cd_title: (string) with the title of the CD
        cd_artist: (string) with the artist of the CD
    methods:

    """
    # DONE Add Code to the CD class
    def __init__(self,CD_ID,CD_TITLE,CD_ARTIST):
        self.__cd_id = CD_ID
        self.__cd_title = CD_TITLE
        self.__cd_artist = CD_ARTIST

    @property
    def cd_id(self):
        return int(self.__cd_id)
    @cd_id.setter
    def cd_id(self,ID):
        if type(ID) == 'int':
            self.__cd_id = ID
        else:
           raise Exception ('Not an integer')

    @property
    def cd_title(self):
        return self.__cd_title
    @cd_title.setter
    def cd_title(self,Title):
        self.__cd_title = Title

    @property
    def cd_artist(self):
        return self.__cd_artist
    @cd_artist.setter
    def cd_artist(self,Artist):
            self.__cd_artist = Artist
```

Figure 5 Assignment 08 Screenshot of the Spyder Program

Execution of Program

As requested in the assignemnt08 I have executed the script in Spyder and in Command Prompt.

```
(IPdb [22]): runfile('C:/Users/sayaliss/.spyder-py3/Mod8/CD_Inventory.py', wdir='C:/Users/sayaliss/.spyder-py3/Mod8')
Menu

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


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


Enter ID: 1

What is the CD's title? the big river

What is the Artist's name? Runrig
======= The Current Inventory: =======
ID  CD Title Artist

1, the big river, Runrig

======================================
Menu

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


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

Figure 6 Assignment08 Screenshot of Execution of Adding data in Sypder

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

======= The Current Inventory: =======
ID  CD Title Artist

1, the big river, Runrig

2, Bad, Michael Jackson

======================================

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

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

Figure 7 Assignment08 Execution screenshot of Saving inventory in the file in Sypder

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


Enter ID: rt
This ID is not integer type , please enter integer

Enter ID: 3

What is the CD's title? forever

What is the Artist's name? taylor Swift
======= The Current Inventory: =======
ID  CD Title Artist

1,  the big river,  Runrig

2,  Bad,  Michael Jackson

3, forever, taylor Swift

=====================================
Menu

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

Figure 8 Assignment08 Execution screenshot of Error check while adding inventory in Spyder

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

======= The Current Inventory: =======
ID  CD Title Artist

1,  the big river,  Runrig

2,  Bad,  Michael Jackson

3, forever, taylor Swift

=====================================
Menu

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

Figure 9 Assignment08 Execution screenshot o displaying inventory in Spyder

```
Which operation would you like to perform? [l, a, i, 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 Artist

1,  the big river,  Runrig

2,  Bad,  Michael Jackson

=====================================
Menu

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

Figure 10 Assignemnt08 Execution Screenshot of loading data from memory if not saved lost while loading

```
=====================================
Menu

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


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


(IPdb [23]):
```

Figure 11 Assignment08 Execution in Spyder for exit if x option chosen

```
C:\Users\sayaliss\.spyder-py3\Mod8\cdInventory.txt
listing3.py* × | poll.py × | CDInventory.py - Mod6 × | untitled14.py* × | CDInventory.py - Mod7 × | CD_Inventory.py × | cdInventory.txt × | CdInventory.txt ×
1  1, the big river, Runrig
2  2, Bad, Michael Jackson
3
```

Figure 12Assignment08 CdInventory.txt file Screenshot

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\sayaliss\.spyder-py3>cd Mod8

(base) C:\Users\sayaliss\.spyder-py3\Mod8>python CD_Inventory.py
Menu

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

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

======= The Current Inventory: =======
ID      CD Title Artist

1,  the big river,  Runrig

2,  Bad,  Michael Jackson

======================================
Menu

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

Which operation would you like to perform? [l, a, i, 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 Artist

1,  the big river,  Runrig

2,  Bad,  Michael Jackson

======================================
Menu

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

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

Enter ID: 3
What is the CD's title? Forever
What is the Artist's name? Taylor Swift
======= The Current Inventory: =======
ID      CD Title Artist

1,  the big river,  Runrig

2,  Bad,  Michael Jackson

3, Forever, Taylor Swift

======================================
Menu

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

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

======= The Current Inventory: =======
ID      CD Title Artist

1,  the big river,  Runrig

2,  Bad,  Michael Jackson

3, Forever, Taylor Swift

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

Figure 13Assignment08 Execution of Python code in 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 started with OOP concepts, class, and its usage. Using dunder functions like __init__constructor and __str__ for printing are new concepts in OOP I learnt I also learnt that self is that object itself.

I have uploaded the Gitlab code : https://github.com/sayalisu/Assignment_082

Appendix

Script

```
#-----------------------------------------#
# Title: CD_inventory.py
# Desc: CD_inventory - Working with classes
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, created file
# DBiesinger, 2030-Jan-01, added pseudocode to complete assignment 08
# Sayali, 2022-March-20, completed the TODOs
#-----------------------------------------#

# -- DATA -- #
strFileName = 'cdInventory.txt'
lstOfCDObjects = []

class CD:
    """Stores data about a CD:

    properties:
        cd_id: (int) with CD ID
        cd_title: (string) with the title of the CD
        cd_artist: (string) with the artist of the CD
    methods:

    """
    # DONE Add Code to the CD class
    def __init__(self,CD_ID,CD_TITLE,CD_ARTIST):
        self.__cd_id = CD_ID
        self.__cd_title = CD_TITLE
        self.__cd_artist = CD_ARTIST

    @property
    def cd_id(self):
        return int(self.__cd_id)
    @cd_id.setter
    def cd_id(self,ID):
        if type(ID) == 'int':
            self.__cd_id = ID
        else:
          raise Exception ('Not an integer')

```

Retrieved 2022-Feb-04

2 Retrieved 2022-Feb-27

```
    @property
    def cd_title(self):
        return self.__cd_title
    @cd_title.setter
    def cd_title(self,Title):
        self.__cd_title = Title

    @property
    def cd_artist(self):
        return self.__cd_artist
    @cd_artist.setter
    def cd_artist(self,Artist):
            self.__cd_artist = Artist

    #methods
    def add_list_of_obj(cdObj):
          lstOfCDObjects.append(cdObj)
          return lstOfCDObjects
    def __str__(self):
        str_new = ''
        str_new = str_new +  str(self.__cd_id) + ', ' + self.__cd_title + ', ' + self.__cd_artist +'\n'
        return str_new

# -- PROCESSING -- #
class FileIO:
    """Processes data to and from file:

    properties:

    methods:
        save_inventory(file_name, lst_Inventory): -> None
        load_inventory(file_name): -> (a list of CD objects)

    """
    # DONE Add code to process data from a file
    # DONE Add code to process data to a file
    @staticmethod
    def save_inventory(file_name, lst_Inventory):
        # try :

            file1 =open(file_name,'w')
            for row in lst_Inventory:
                str1 = str(row.cd_id) + ', ' + row.cd_title + ', ' + row.cd_artist + '\n'
                file1.write(str1)
            file1.close()
        # except AttributeError:
        #     print("'CD' object has no attribute '_FileIO__cd_id'")

    @staticmethod
    def load_inventory(file_name):
        lstOfCDObjects.clear()
        try :
            file1 =open(file_name,'r')
            for row in file1:
                data = row.strip().split(',')
                CdInfo = CD(int(data[0]),data[1],data[2])
                lstOfCDObjects.append(CdInfo)
            file1.close()
        except FileNotFoundError:
```

```
        print("Filenot_found,please make sure file is present")
    except EOFError:
            print("File has no contents.Please add data to the file and then read")
    except ValueError:
                print("Check the values")
    return lstOfCDObjects


# -- PRESENTATION (Input/Output) -- #
class IO:
    # Done add docstring
    # DONE add code to show menu to user
    # DONE add code to captures user's choice
    # DONE add code to display the current data on screen
    # DONE add code to get CD data from user
    """Handling Input / Output

    properties:

    methods:
        print_menu(): -> None
        menu_choice(): -> (choice user has selcted from the menu)
        show_inventory(): -> (Shows the current inventory )
        add_inventory(): -> (CDObject -> Object of class CD with all attributes.)

    """
    @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('[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, s or x

        """
        choice = ' '
        while choice not in ['l', 'a', 'i',  's', 'x']:
            choice = input('Which operation would you like to perform? [l, a, i, 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: =======')
        print('ID\tCD Title Artist\n')
        for row in table:
            print(row.__str__())
        print('======================================')
    @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:CDObject -> Object of class CD with all attributes.

        '''
        intID = None
        strTitle = ''
        stArtist = ''
        while type(intID) != int:
            try:
                intID = int(input('Enter ID: ').strip())
            except ValueError :
                print("This ID is not integer type , please enter integer ")

        strTitle = input('What is the CD\'s title? ').strip()
        stArtist = input('What is the Artist\'s name? ').strip()
        CdObject = CD (intID, strTitle, stArtist)
        return  CdObject



# -- Main Body of Script -- #
# DONE Add Code to the main body

lstOfCDObjects =FileIO.load_inventory(strFileName)

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...')
            FileIO.load_inventory(strFileName)
            IO.show_inventory(lstOfCDObjects)
        else:
            input('canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')
            IO.show_inventory(lstOfCDObjects)
        continue  # start loop back at top.
    # 3.3 process add a CD
    elif strChoice == 'a':
        CDObject = IO.add_inventory()
        lstOfCDObjects =CD.add_list_of_obj(CDObject)
        IO.show_inventory(lstOfCDObjects)
        continue  # start loop back at top.
    # 3.4 process display current inventory
    elif strChoice == 'i':
        IO.show_inventory(lstOfCDObjects)
        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(lstOfCDObjects)
        strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
        # 3.6.2 Process choice
        if strYesNo == 'y':
            FileIO.save_inventory(strFileName, lstOfCDObjects )
            # 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')
```