Emeka Osuji
August 28th, 2022
Foundations of Programming, Python
Assignment 8

CD INVENTORY VIA LIST OF OBJECTS

INTRODUCTION

For this assignment we will be implementing the use of objects as a means of storing data and attributes. In previous assignment, we used a list of dictionaries to load, store and save data. This assignment requires the use of the `_init_()` function in order to create attributes and therefore generate a list of objects. Other modifications will also be implemented into the code. These include, but not limited to, loading the file from the start of the script as well as modifying how object attribute are called for the purposes of displaying the data or saving it. The script will be run in sypder and in the anaconda user terminal for validation purposes. Furthermore, the documentation, along with the script, will be posted on GitHub as per the requirements.

I OBJECT!

The concept of creating objects appears straight forward. Essentially, objects are created from classes. Within the class, an `_init_()` function is called in order to declare the attributes of the object. So, for this assignment, the attributes of a CD entry would be the CD ID number, the CD title and finally the CD artist name. An example of this can be seen in figure 1. Based on the structure of the Assignment 8 starter code, the class used for this will be within the CD class. In the initial stages, the `_init_()` function was erroneously placed in the IO class, which create some difficulty in adhering to the initial format of the starter script.

```python
def __init__(self, iD, title, artist):
    self.__iD = iD
    self.__title = title
    self.__artist = artist


@property
def ID(self):
    return self.__iD

@ID.setter
def ID(self, iD):
    self.__iD = iD

@property
def title(self):
    return self.__title

@title.setter
def title(self, title):
    self.__title = title

@property
def artist(self):
    return self.__artist

@artist.setter
def artist(self, artist):
    self.__artist = artist
```

Figure 1: _init_() function implementation and using "setters" and "getters"

Once the attributes have been declared, it is now left to set up the “setters” and “getters”. This portion allows for the declaration of attribute values once they have been created. However, an unclear aspect of developing the setters and getters is regarding decorators. A YouTube video by Corey Schafer provided some clarity but it would be clearer once the features of these decorators are made more apparent through re-world scenarios.

Another aspect of developing setters and getter which was unclear would be regarding the naming format. Per the module as well as other online resources, it is stated that using a double underscore denotes the attribute as private. It is not entirely clear how this works in general, but it was implemented in order to be consistent with the module lessons. Further clarification on decorators and the naming convention for private attribute would be much appreciated.

Access Denied

With the list of objects, it was now left for the implementation of code that would allow the script to display or save the data to a text file and in the desired format. This part of the script was personally the trickiest aspect of this assignment. For example, when attempting to print out a certain attribute of an object you must call the attribute in the appropriate manner. So, if one had a loop that stated “`For row in table`:” but this loop was in a different class than the

CD class, retrieving the attribute would be expressed as row._CD__iD. Similarly, if the For loop were to be run in the same class from which the object was derived, one would only need to use row.__iD since the call is being made from within the existing class

Details such as this required a lot of trial and error. However, a built-in function that made this process somewhat easier was the use of the `dir()` function. This function displays a list of the object's attributes and methods. With the script now completed, the link to the script along with this documentation can be fount on GitHub. The script was run in spyder and the anaconda terminal as seen in Appendix I and II.

SUMMARY

For this week, we modified our CD inventory script to us a list of objects rather than a list of dictionaries. This process was developed by creating objects from the CD class. To do this, the `_init_()` function is used to declare the attribute then the setters and getters are implemented to set the values for the attribute. Certain difficulties include being able to properly call the CD attribute values and being able to understand the decorator feature.

APPENDIX

I)

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

Enter ID: 2
What is the CD's title? Bad
What is the Artist's name? Michael Jackson
1
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1    The Bad Wheel (by:Runrig)
2    Bad (by:Michael Jackson)
=====================================
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 Bad Wheel (by:Runrig)
2    Bad (by:Michael Jackson)
=====================================
Which ID would you like to delete? 1
The CD was removed
======= The Current Inventory: =======
ID  CD Title (by: Artist)

2    Bad (by:Michael Jackson)
=====================================
Menu

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

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

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

2    Bad (by:Michael Jackson)
=====================================
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
```

II)

```
(base) C:\Users\Emeka\.spyder-py3>python Assignment8.py
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2        Bad (by:Michael Jackson)
=======================================
Menu

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

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

Enter ID: 3
What is the CD's title? emeka
What is the Artist's name? emeka
2
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2        Bad (by:Michael Jackson)
3        emeka (by:emeka)
=======================================
Menu

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

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

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

2        Bad (by:Michael Jackson)
3        emeka (by:emeka)
=======================================
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
```