Evan Anderson

12/8/2021

Foundations of Programming, Python

Assignment08

CD Inventory Program (Software Objects)

Introduction

In the latest iteration of the CD Inventory program, software objects were utilized for managing CDs within the inventory. The methodology is called object-oriented programming (OOP), a computer programming model that organizes software design around data, or objects, rather than functions and logic. As such, an object can be defined as a data field that has unique attributes and behavior. With limited coding, this type of class can be instantiated infinite times to create new objects within the program.

Creating a CD Class

The main objective of this assignment was to continue the CD Inventory program with CD's saved as instantiated objects within the program. To do this, a CD class was created so that a constructor could define the attributes of the CD (ID, Title & Artist). The string method was also called within the class to define the string associated with the object. For simplicity, the program separates the attributes by parenthesis.

```
26      def __init__(self, ID, Title, Artist): # Constructor
27          # -- Attributes -- #
28          self.cd_id = ID
29          self.cd_title = Title
30          self.cd_artist = Artist
31
32      def __str__(self):
33          return '{}, {}, {}'.format(self.cd_id,self.cd_title,self.cd_artist)
```

Figure 1

Object Obstacles

Because the CD's were now being saved as objects, one of the biggest issues that arose was converting the objects to strings in order to export them to a text file. This was overcome by calling the __str__ method which converted the attributes to their string format.

```
51          objFile = open(file_name, 'w')
52          for row in lst_Inventory:
53              objFile.write(row.__str__() + '\n')
54          objFile.close()
```

Figure 2

The other obstacle was converting the text file strings into lists so that the values could be assigned to object attributes. Once the new object was created, it could then be appended to the existing list of objects within the program.

```
76          for line in objFile:
77              data = line.strip().split(',')
78              ObjectName = CD(data[0],data[1],data[2])
79              table.append(ObjectName)
80          objFile.close()
```

Figure 3

Testing the Program (in Spyder)

After the major obstacles were solved, it was time to test the program out in Spyder:

```
In [1]: runfile('C:/_FDProgramming/Mod_08/Assignment08/Assignment08.py', wdir='C:/_FDProgramming/Mod_08/Assignment
Source file not found! New text document 'CDInventory.txt' created.

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to file
[x] Exit
```

Figure 4

Testing out adding a CD:

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

Enter ID: 1

What is the CD's title? Test 1

What is the Artist's name? Test 2

======= The Current Inventory: =======
ID, CD Title, Artist
1, Test 1, Test 2
======================================
```

Figure 5

Loading the inventory:

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

======= The Current Inventory: =======
ID, CD Title, Artist
1, Test 1, Test 2
======================================
```

Figure 6

Saving the inventory to the external text document:

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

======= The Current Inventory: =======
ID, CD Title, Artist
1, Test 1, Test 2
======================================


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

Figure 7

Lastly, loading the contents of the text file to the program:

```
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 will be reloaded from file.

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

Reloading...

======= The Current Inventory: =======
ID, CD Title, Artist
1,  Test 1,  Test 2
======================================
```

Testing the Program (in Terminal)

Opening and adding a CD:

```
(base) C:\_FDProgramming\Mod_08\Assignment08>python Assignment08.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]: a
Enter ID: 2
What is the CD's title? Test 2
What is the Artist's name? Test 3

======= The Current Inventory: =======
ID, CD Title, Artist
1,  Test 1,  Test 2
2, Test 2, Test 3
======================================
```

Figure 8

Viewing inventory:

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

======= The Current Inventory: =======
ID, CD Title, Artist
1,  Test 1,  Test 2
2, Test 2, Test 3
3, Test 3, Test 4
======================================
```

Figure 9

Saving inventory to text file:

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

======= The Current Inventory: =======
ID, CD Title, Artist
1,  Test 1,  Test 2
2, Test 2, Test 3
3, Test 3, Test 4
======================================

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

Figure 10

Loading inventory from text file:

```
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 will be reloaded from file.
Type 'yes' to continue and reload from file. Otherwise reload will be canceled: yes

Reloading...

======= The Current Inventory: =======
ID, CD Title, Artist
1,  Test 1,  Test 2
2,  Test 2,  Test 3
3,  Test 3,  Test 4
======================================
```

Figure 11

Conclusion

In conclusion, this assignment showed us how object oriented programming can create significant efficiencies in programs when there are many objects with similar attributes within a program. The benefits include modularity for easier troubleshooting, reusable of code through inheritance, flexibility through polymorphism and effective problem solving.

Github link: https://github.com/Evanderson34/Assignment08.git