

Charles Hodges(hodges11@uw.edu)

August 29, 2021

IT FDN 110: Introduction to Programming (Python)

Assignment 08

GitHub Repository Link: [https://github.com/chodges11/Assignment\\_08](https://github.com/chodges11/Assignment_08)

# Beginning Object-Oriented Programming

## Introduction

This assignment was this class' introduction to Object-Oriented Programming(OOP). For this assignment, we were given a sample script with pseudocode, instructing us to fill in the program, based on TODO instructions. We worked with CD objects, building off of every previous assignment. Although, the Delete CD functionality was removed, and not mentioned in the pseudocode, so that is a delta from the previous assignments. Finally, I wanted to say that I got your hint about String variables at the top, and used hardly any.

CD Class

```

44.     # Constructor
45.     def __init__(self, cd_id, cd_title, cd_artist):
46.         # Attributes
47.         print("\nA new CD was created!\n")
48.         self.__cd_id = cd_id
49.         self.__cd_title = cd_title
50.         self.__cd_artist = cd_artist
51.
52.     # Getter and Setter for the CD's ID number
53.     @property
54.     def cd_id(self):
55.         return self.__cd_id
56.
57.     @cd_id.setter
58.     def cd_id(self, cd_id):
59.         self.__cd_id = cd_id
60.
61.     # Getter and Setter for the CD's Title
62.     @property
63.     def cd_title(self):
64.         return self.__cd_title
65.
66.     @cd_title.setter
67.     def cd_title(self, cd_title):
68.         self.__cd_title = cd_title
69.
70.     # Getter and Setter for the CD's Artist
71.     @property
72.     def cd_artist(self):
73.         return self.__cd_artist
74.
75.     @cd_artist.setter
76.     def cd_artist(self, cd_artist):
77.         self.__cd_artist = cd_artist
78.
79.     # Methods
80.     def __str__(self):
81.         """Represents the class objects as a string."""
82.         return ('{: <7} {: <20} {: <20}'.format(
83.             str(self.__cd_id), str(self.__cd_title), str(self.__cd_artist)))

```

*Figure 1 - CD Object Body*

Here's the body of the CD object class. We start by initializing it with some attributes, for the three most important pieces of CD information. Then we create Setters and Getters for each attribute. Finally, we create a "`__str__`" method so as to create a string version of the objects that are created.

## Adding a CD

```
219.     @staticmethod
220.     def add_cd(list: lst_of_cd_objects) -> None:
221.         """Create a CD object and add it the User's Inventory'.
222.
223.         Accepts the User input of new CD information, and creates a CD
224.         object, which is appended to the list table which makes up the
225.         Inventory.
226.
227.         Args:
228.             lst_of_cd_objects(list): List of CD Objects
229.
230.         Returns:
231.             lst_of_cd_objects(list): Updated list of CD Objects
232.         """
233.         # Collect the CD ID
234.         while True:
235.             try:
236.                 cd_id = int(input("Enter the ID number of your CD: "))
237.                 break
238.             except ValueError:
239.                 print("Please only enter whole numbers.")
240.         # Collect the CD Title
241.         cd_title = input("Enter the Title of the CD: ")
242.         # Collect the CD Artist's Name
243.         cd_artist = input("Enter the Artist who created the CD: ")
244.         # Create a CD object with the provided information and
245.         # append that CD object to list of CD objects
246.         lst_of_cd_objects.append(CD(cd_id, cd_title, cd_artist))
247.         IO.show_inventory(lst_of_cd_objects)
248.         return lst_of_cd_objects
```

Figure 2 - ADD\_CD method

For the Add CD method, I had to have the “`lst_of_cd_objects.append(CD(cd_id, cd_title, cd_artist))`” located here, since there was no dataProcessing section, as in previous assignments. Creating a whole new class for a method containing only that one line was not

worth the added complexity. That one line creates a CD object, passing in the user entered attributes, and appends it to the list of CD objects.

## Error Handling

```
133.         # Loads data from file
134.         obj_file = open(str_file_name, 'rb')
135.         try:
136.             lst_of_cd_objects = pickle.load(obj_file)
137.         except EOFError:
138.             pass
139.         obj_file.close()
```

Figure 3 - EOFError

```
233.         # Collect the CD ID
234.         while True:
235.             try:
236.                 cd_id = int(input("Enter the ID number of your CD: "))
237.                 break
238.             except ValueError:
239.                 print("Please only enter whole numbers.")
```

Figure 4 - ValueError

```
252. # When program starts, read in the currently saved Inventory, if it exists.
253. # Otherwise, create the inventory file.
254. try:
255.     lst_of_cd_objects = FileIO.load_inventory(str_file_name, lst_of_cd_objects)
256. except FileNotFoundError:
257.     FileIO.create_file(str_file_name)
```

Figure 5 - FileNotFoundError

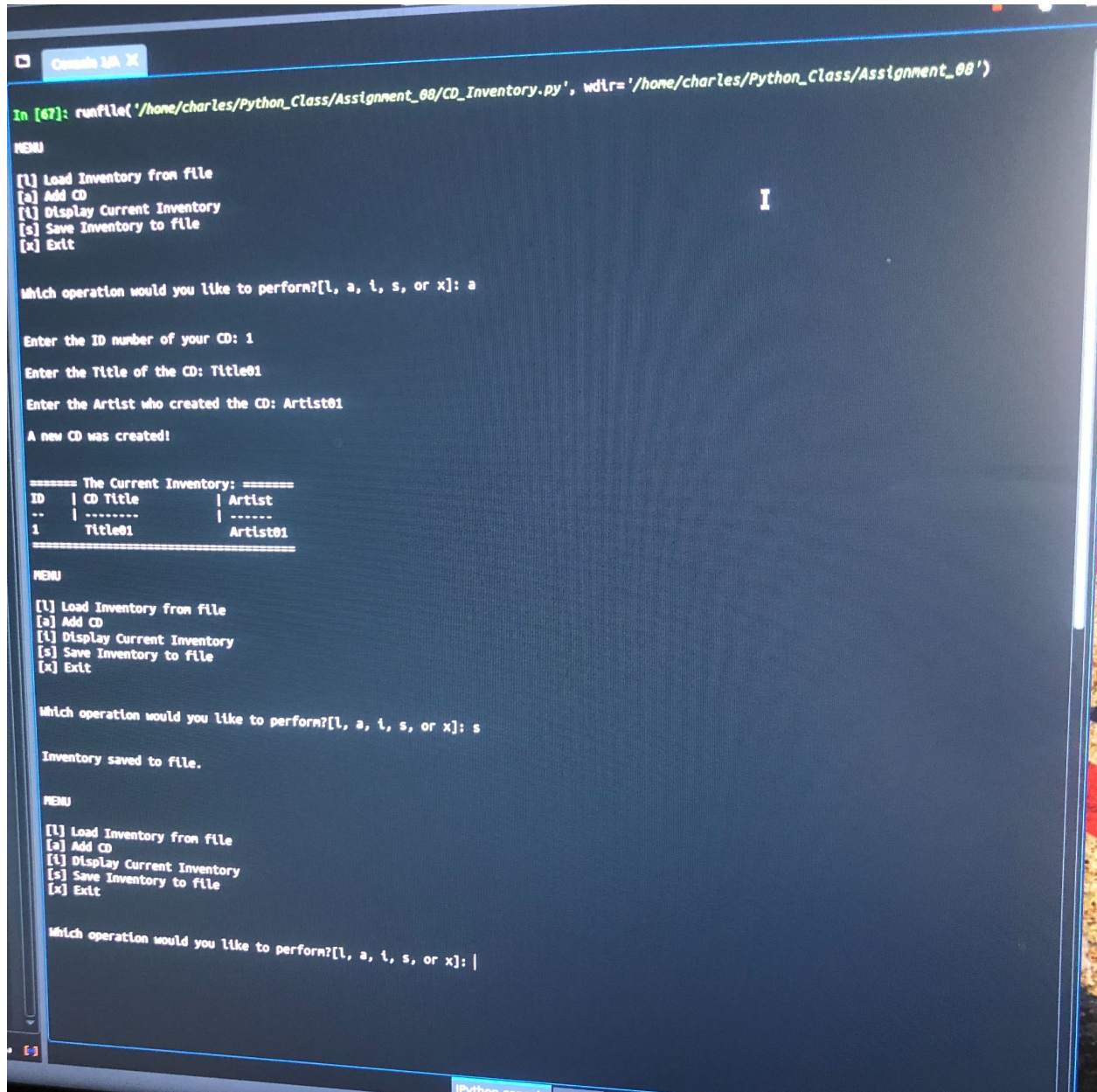
Since it was called out specifically in the Assignment PDF, I wanted to call out the use of error handling in this program. There are three different errors which are handled here, one dealing with user input and two handling file processing. Oddly, the provided pseudocode mentioned loading from the file upon initial program execution. But there was no file provided to load from. So I created the file, as I had in other programs, if a file was not already present.

# Summary

In this assignment, we were given this class' first assignment working with Object Oriented programming, by creating CD objects, in order to manipulate within our Inventory. We created CD objects, added them to a list in memory. From there, we could save our inventory to a file, or load the inventory already in a file, overwriting our edited list. In previous programming classes, I modelled cups of coffee, and elevators. Like those, modelling mundane, but everyday, practical objects helps us students wrap our heads around this next level of the programming game.

# Screenshots

## Spyder



```
In [67]: runfile('/home/charles/Python_Class/Assignment_08/CD_Inventory.py', wdir='/home/charles/Python_Class/Assignment_08')

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 the ID number of your CD: 1
Enter the Title of the CD: Title01
Enter the Artist who created the CD: Artist01
A new CD was created!

===== The Current Inventory: =====
ID | CD Title | Artist
-- | -
1 | Title01 | Artist01

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

Inventory saved to file.

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]: |
```

Figure 6 - Spyder Screenshot



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

Enter the ID number of your CD: s
Please only enter whole numbers.

Enter the ID number of your CD: 2

Enter the Title of the CD: Title02

Enter the Artist who created the CD: Artist02

A new CD was created!

===== The Current Inventory: =====
ID | CD Title | Artist
-- | -
1 | Title01 | Artist01
2 | Title02 | Artist02
=====

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

Inventory saved to file.

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

Inventory loaded from file.

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]: |
```

Figure 7 - Spyder Screenshot

```
add CD
display Current Inventory
save Inventory to file
Exit

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

68: runfile('/home/charles/Python_Class/Assignment_08/CD_Inventory.py', wdir='/home/charles/Python_Class/Assignment_08')
Inventory loaded from file.

Load Inventory from file
Add CD
Display Current Inventory
Save Inventory to file
Exit

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

Inventory loaded from file.

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 | Title01 | Artist01
2 | Title02 | Artist02
=====

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]:
```

Figure 8 - Spyder Screenshot

Terminal



```
charles@charles-XPS-13-9370: ~/Python_Class/Assignment_08
(base) charles@charles-XPS-13-9370:~/Python_Class/Assignment_08$ python CD_Inventory.py

Inventory loaded from file.

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    | Title01           | Artist01
2    | Title02           | Artist02
=====

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 the ID number of your CD: 3
Enter the Title of the CD: Title03
Enter the Artist who created the CD: Artist03

A new CD was created!

===== The Current Inventory: =====
ID   | CD Title          | Artist
--   | -----          | -
1    | Title01           | Artist01
2    | Title02           | Artist02
3    | Title03           | Artist03
=====

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

Inventory saved to file.

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

(base) charles@charles-XPS-13-9370:~/Python_Class/Assignment_08$ gnome-screenshot --window
```

*Figure 9 - Terminal Screenshot*

```
charles@charles-XPS-13-9370: ~/Python_Class/Assignment_08
-- | ----- | -----
1   Title01      Artist01
2   Title02      Artist02
3   Title03      Artist03
=====

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 the ID number of your CD: 4
Enter the Title of the CD: Title04
Enter the Artist who created the CD: Artist04

A new CD was created!

===== The Current Inventory: =====
ID | CD Title      | Artist
-- | ----- | -----
1   Title01      Artist01
2   Title02      Artist02
3   Title03      Artist03
4   Title04      Artist04
=====

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

Inventory loaded from file.

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   Title01      Artist01
2   Title02      Artist02
3   Title03      Artist03
=====

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
```

*Figure 10 - Terminal Screenshot*