Andrew Yeo
March 11, 2020
Foundations of Programming: Python
Github Repository: https://github.com/andyhyeo/assignment_08

Module 08
Knowledge Document

Intro

In this week's assignment, we go over Object Oriented Programming. Object Oriented Programming makes use of classes which are the blueprints for objects. The class maintains then data and the operations of the object. The object is an "instantiation" of the class. And the "features" of the objects are inherited from its class (attributes, properties, and methods)

Details

In this week's assignment we are instructed to build the CD inventory script from scratch. I l largely already used what the script that we have been using in order to focus on the parts that were the topics of this weeks modules: object oriented programming, defining classes, methods, properties, and attributes.

Class CD

```
# -- DATA -- #
# DBiesinger, 2030-Jan-01, created file
from os import path
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:

    """
    # -- Contructor / Initializer -- #
    def __init__(self, cd_id, cd_title, cd_artist):
        self.id = cd_id
        self.title = cd_title
        self.artist = cd_artist
```

Figure 1: Building the CD class with attributes of id, title, and artist.

The CD class was built simply by adding parameters. Even though I have gotten a little bit of purchase on Object Oriented Programming concepts, I still had to build this largely by emulating the script within the module notes. I tested it within the console to make sure that it probably stored data when called.

```
    @staticmethod
    def load_inventory(file_name, lstOfCDObjects):
        """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

        Returns:
            None.
        """
        try:
            if path.exists('cdInventory.txt'):  #Corrects for a bug in the original file whereb
                lstOfCDObjects.clear()  # this clears existing data and allows to load data fro
                objFile = open(file_name, 'r')
                for line in objFile:
                    data = line.strip().split(',')
                    dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
                    lstOfCDObjects.append(dicRow)
                objFile.close()
                return lstOfCDObjects
        except FileNotFoundError:
            pass
```

Figure 2: Building the FileIO class with the method of load_inventory.

Class File IO with Method load_inventory

Figure 2 shows the part of the script where I built the load_inventory method of within the class FileIO. The module goes over concepts and best practices of writing class such as writing constructors, attributes, and properties but I was not precisely sure how to apply that and if I should have applied it. It was largely a copy and paste from the previous scripts. For simplicity, I opted to use the non-pickling loading strategy. And as shown, I wrote in the structured error handling to continue with the script if the file is not found.

Class File IO with Method save_inventory

```
86      @staticmethod
87      def save_inventory(file_name, table):
88          """Writes the inventory of IDs, CD Names, and Artists to a text file
89
90          Args:
91              file_name (string): The name of the file that it will write to
92              table (list of dict): 2D data structure (list of dicts) that holds the data during
93
94          Returns:
95              None but saves a file in the directory of the python script
96
97          """
98          objFile = open(file_name, 'w')
99          for row in table:
100             strRow = ''
101             for item in row.values():
102                 strRow += str(item) + ','
103             strRow = strRow[:-1] + '\n'
104             objFile.write(strRow)
105         objFile.close()
```

Figure 3: Building the FileIO class with the method of save_inventory.

The load_inventory script was an amalgamation of previous scripts and so to was the the save_inventory.

Try-Except Error Structure Handling for ValueError in IO.ask() function

I also successfully substituted the try-except structure within the IO.ask() function which took a fair amount of trial error. I am learning that I can use all of the Spyder console in order to help me debug a code. For instance, I can run just line by line of the code. I can use the variable explorer as feedback if a variable was defined or changed the way that I wanted to.

```
181     @staticmethod
182     def ask():
183         """Ask user for new ID, CD Title and Artist
184 
185         Args:
186             None
187 
188         Returns:
189             dicRow (dictionary):  A dictionary entry with ID (int): integer that holds
190             the ID tag,title (string): string that holds the name of the CD
191                 and an artist (string): string that holds the name of the Artist.
192         """
193         while True:
194             strID = input('Enter ID: ').strip()
195             try:
196                 strID = int(strID)
197                 break
198             except ValueError:
199                 print('That is not an integer')
200         strTitle = input('What is the CD\'s title? ').strip()
201         stArtist = input('What is the Artist\'s name? ').strip()
202         dicRow = {'ID': strID, 'CD Title': strTitle, 'Artist': stArtist}
203 
204         return dicRow
```

Figure 4: Substituting Structured Error Handling in the Ask Function

Try-Except Error Structure Handling to handle not-integers when deleting

Figure 5 shows the part of the CD Inventory script which prompts the user to that to put in an integer when they have not done so succesfully.

```
247     elif strChoice == 'd':
248         # 3.5.1 get Userinput for which CD to delete
249         # 3.5.1.1 display Inventory to user
250         IO.show_inventory(lstTbl)
251         # 3.5.1.2 ask user which ID to remove
252         while True:
253             try:
254                 intIDDel = int(input('Which ID would you like to delete? ').strip())
255                 break
256             except ValueError:
257                 print('That is not an integer')
258         # 3.5.2 search thru table and delete CD
259         DataProcessor.delete_cd(intIDDel,lstTbl)
260         IO.show_inventory(lstTbl)
261         continue  # start loop back at top.
```

Figure 4: Error Structure Handling to handle not-integers when prompted to delete

Summary

As Dirk mentions in his notes and lectures, I do not yet follow appreciate the power of Object Oriented Programming. Dirk mentions that it takes more than just one time using it appreciate its power and beauty. I figure that some of the ways that I can come to appreciate it is to try and building really difficult code without it. Also as Dirk mentions, its usage is probably so standardized and par for the course that it might be difficult to "learn" it again.

Some of the future investigations can include adding the features of pickling to this code in order to increase efficiency of data storage and management. Another feature that can be looked into adding is the feature of preventing duplicate IDs to be stored.

References

https://www.youtube.com/watch?v=lkXfgP-fAkY&feature=youtu.be
https://www.youtube.com/watch?v=KZdFvyCOLUQ&feature=youtu.be
https://www.youtube.com/watch?v=qvRyls8NX-E&feature=youtu.be
https://www.youtube.com/watch?v=QuwU34OT4XA&feature=youtu.be
https://www.youtube.com/watch?v=swAXTwW6xoA&feature=youtu.be
https://www.youtube.com/watch?v=5MM6IaESdQ0&feature=youtu.be
https://realpython.com/python3-object-oriented-programming/
https://www.youtube.com/watch?v=IHaTbJPdB-s&feature=youtu.be

Appendix: The Code

Made using Planet B's Syntax Highlighter

```
1.  #------------------------------------------#
2.  # Title: Assignmen08.py
3.  # Desc: Assignnment 08 - Working with classes
4.  # Change Log: (Who, When, What)
5.  # AYeo, 2020-Mar-10 9:06 PM,  Added working CD class
6.  # AYeo, 2020-
    Mar-10 1:52 PM,  Added Main Body from previous script
7.  # AYeo, 2020-Mar-10 2:99 PM,  Added Structured Error Handling
8.
9.  #------------------------------------------#
10.
11. # -- DATA -- #
12. # DBiesinger, 2030-Jan-01, created file
13. from os import path
14. strFileName = 'cdInventory.txt'
15. lstOfCDObjects = []
16.
17. class CD:
18.     """Stores data about a CD:
19.
20.     properties:
21.         cd_id: (int) with CD ID
22.         cd_title: (string) with the title of the CD
23.         cd_artist: (string) with the artist of the CD
24.     methods:
25.
26.     """
27.     # -- Contructor / Initializer -- #
28.     def __init__(self, cd_id, cd_title, cd_artist):
29.         self.id = cd_id
30.         self.title = cd_title
31.         self.artist = cd_artist
32.
33. #  TODO Add Error Handling
34. # Initializer / Instance Attributes
35. #    @property
36. #    def ID(self):
37. #        return self.__id.title()
38. #
39. #    @ID.setter
40. #    def ID(self, value):
41. #        if str(value).isString():
```

```
#             raise Exception('The ID can only be an integer')
#         else:
#             self.__id = value

# -- 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)

    """
    # TODO Add code to process data from a file
    @staticmethod
    def load_inventory(file_name, lstOfCDObjects):
        """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.
        """
        try:
            if path.exists('cdInventory.txt'):  #Corrects for a bug in the original file whereby the script did not run, if there was not already .txt file
                lstOfCDObjects.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]}
```

```
                    lstOfCDObjects.append(dicRow)
                objFile.close()
                return lstOfCDObjects
        except FileNotFoundError:
            pass

    # TODO Add code to process data to a file
    @staticmethod
    def save_inventory(file_name, table):
        """Writes the inventory of IDs, CD Names, and Artists to a text file

        Args:
            file_name (string): The name of the file that it will write to
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            None but saves a file in the directory of the python script

        """
        objFile = open(file_name, 'w')
        for row in table:
            strRow = ''
            for item in row.values():
                strRow += str(item) + ','
            strRow = strRow[:-1] + '\n'
            objFile.write(strRow)
        objFile.close()
# -- PRESENTATION (Input/Output) -- #
class IO:
    """Handling Input / Output"""
    # TODO add docstring
    # TODO add code to show menu to user
    # TODO add code to captures user's choice
    # TODO add code to display the current data on screen
    # TODO add code to get CD data from user
    @staticmethod
    def print_menu():
        """Displays a menu of choices to the user

        Args:
            None.

```

```
121.         Returns:
122.             None.
123.         """
124.
125.         print('Menu\n\n[l] load Inventory from file\n[a] Add CD\n
    [i] Display Current Inventory')
126.         print('[d] delete CD from Inventory\n[s] Save Inventory t
    o file\n[x] exit\n')
127.
128.     @staticmethod
129.     def menu_choice():
130.         """Gets user input for menu selection
131.
132.         Args:
133.             None.
134.
135.         Returns:
136.             choice (string): a lower case sting of the users inpu
    t out of the choices l, a, i, d, s or x
137.
138.         """
139.         choice = ' '
140.         while choice not in ['l', 'a', 'i', 'd', 's', 'x']:
141.             choice = input('Which operation would you like to per
    form? [l, a, i, d, s or x]: ').lower().strip()
142.         print()  # Add extra space for layout
143.         return choice
144.
145.     @staticmethod
146.     def show_inventory(table):
147.         """Displays current inventory table
148.
149.
150.         Args:
151.             table (list of dict): 2D data structure (list of dict
    s) that holds the data during runtime.
152.
153.         Returns:
154.             None.
155.
156.         """
157.         print('======= The Current Inventory: =======')
158.         print('ID\tCD Title (by: Artist)\n')
159.         for row in table:
160.             print('{}\t{} (by:{})'.format(*row.values()))
161.         print('======================================')
```

```
162.
163.     @staticmethod
164.     def add_cd(row, table):
165.         """Adds a dictionary row to the inventory
166.
167.         Args:
168.             row (dictionary): dictionary that holds the name of t
    he ID, cd, and artist
169.             table (list of dict): 2D data structure (list of dict
    s) that holds the data during runtime
170.
171.         Returns:
172.             None.
173.
174.         """
175.         table.append(row)
176.         return table
177.
178.
179.     @staticmethod
180.     def ask():
181.         """Ask user for new ID, CD Title and Artist
182.
183.         Args:
184.             None
185.
186.         Returns:
187.             dicRow (dictionary):  A dictionary entry with ID (int
    ): integer that holds
188.             the ID tag,title (string): string that holds the name
     of the CD
189.                 and an artist (string): string that holds the nam
    e of the Artist.
190.         """
191.         while True:
192.             strID = input('Enter ID: ').strip()
193.             try:
194.                 strID = int(strID)
195.                 break
196.             except ValueError:
197.                 print('That is not an integer')
198.         strTitle = input('What is the CD\'s title? ').strip()
199.         stArtist = input('What is the Artist\'s name? ').strip()

200.         dicRow = {'ID': strID, 'CD Title': strTitle, 'Artist': st
    Artist}
```

```
201.
202.        return dicRow
203.
204.class DataProcessor:
205.    @staticmethod
206.    def delete_cd(intIDDel,table):
207.        """Deletes a CD row from the table
208.
209.        Args:
210.            intIDDel (int): ID which indicate which entry user wo
    uld like to delete
211.            table (list of dict): 2D data structure (list of dict
    s) that holds the data during runtime
212.
213.        Returns:
214.              table (list of dict): 2D data structure (list of di
    cts) that holds the data during runtime
215.        """
216.        intRowNr = -1
217.        blnCDRemoved = False
218.        for row in table:
219.            intRowNr += 1
220.            if row['ID'] == intIDDel:
221.                del table[intRowNr]
222.                blnCDRemoved = True
223.                break
224.            if blnCDRemoved:
225.                print('The CD was removed')
226.            else:
227.                print('Could not find this CD!')
228.        return table
229.
230.# -- Main Body of Script -- #
231.# TODO Add Code to the main body
232.# Load data from file into a list of CD objects on script start
233.FileIO.load_inventory(strFileName, lstOfCDObjects)
234.
235.# 2. start main loop
236.while True:
237.    # 2.1 Display Menu to user and get choice
238.    IO.print_menu()
239.    strChoice = IO.menu_choice()
240.
241.    # 3. Process menu selection
242.    # 3.1 process exit first
243.    if strChoice == 'x':
```

```
244.         break
245.     # 3.2 process load inventory
246.     if strChoice == 'l':
247.         print('WARNING: If you continue, all unsaved data will be
     lost and the Inventory re-loaded from file.')
248.         strYesNo = input('type \'yes\' to continue and reload fro
    m file. otherwise reload will be canceled')
249.         if strYesNo.lower() == 'yes':
250.             print('reloading...')
251.             FileIO.load_inventory(strFileName, lstOfCDObjects)
252.             IO.show_inventory(lstOfCDObjects)
253.         else:
254.             input('canceling... Inventory data NOT reloaded. Pres
    s [ENTER] to continue to the menu.')
255.             IO.show_inventory(lstOfCDObjects)
256.         continue  # start loop back at top.
257.     # 3.3 process add a CD
258.     elif strChoice == 'a':
259.         # 3.3.1 Ask user for new ID, CD Title and Artist
260.         dicRow = IO.ask()
261.         # 3.3.2 Add item to the table
262.         IO.add_cd(dicRow,lstOfCDObjects)
263.         IO.show_inventory(lstOfCDObjects)
264.         continue  # start loop back at top.
265.     # 3.4 process display current inventory
266.     elif strChoice == 'i':
267.         IO.show_inventory(lstOfCDObjects)
268.         continue  # start loop back at top.
269.     # 3.5 process delete a CD
270.     elif strChoice == 'd':
271.         # 3.5.1 get Userinput for which CD to delete
272.         # 3.5.1.1 display Inventory to user
273.         IO.show_inventory(lstOfCDObjects)
274.         # 3.5.1.2 ask user which ID to remove
275.         while True:
276.             try:
277.                 intIDDel = int(input('Which ID would you like to
    delete? ').strip())
278.                 break
279.             except ValueError:
280.                 print('That is not an integer')
281.         # 3.5.2 search thru table and delete CD
282.         DataProcessor.delete_cd(intIDDel,lstOfCDObjects)
283.         IO.show_inventory(lstOfCDObjects)
284.         continue  # start loop back at top.
285.     # 3.6 process save inventory to file
```

```
286.     elif strChoice == 's':
287.         # 3.6.1 Display current inventory and ask user for confir
    mation to save
288.         IO.show_inventory(lstOfCDObjects)
289.         strYesNo = input('Save this inventory to file? [y/
    n] ').strip().lower()
290.         # 3.6.2 Process choice
291.         if strYesNo == 'y':
292.             # 3.6.2.1 save data
293.             FileIO.save_inventory(strFileName, lstOfCDObjects)
294.         else:
295.             input('The inventory was NOT saved to file. Press [EN
    TER] to return to the menu.')
296.         continue  # start loop back at top.
297.     # 3.7 catch-
    all should not be possible, as user choice gets vetted in IO, but
     to be save:
298.     else:
299.         print('General Error')
300.
```

Appendix: The Spyder Output

```
runfile('/Users/andyyeo/Google Drive/Intro to Python/Mod_08/Assignment08/Assignment_08_Starter.py', wdir='/Users/andyyeo/Google Drive/Intro to Python/Mod_08/Assignment08')
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: 1

What is the CD's title? Breaker

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

1       Breaker (by:Broken)
======================================
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       Breaker (by:Broken)
=====================================
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)

1       Breaker (by:Broken)
=======================================

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]:
d

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

1       Breaker (by:Broken)
=====================================

Which ID would you like to delete? 1
======= The Current Inventory: =======
ID      CD Title (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]:
l

WARNING: If you continue, all unsaved data will be lost and the I
nventory re-loaded from file.

type 'yes' to continue and reload from file. otherwise reload wil
l be canceledyes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Breaker (by:Broken)
=======================================
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
```

Appendix: The Terminal Output

```
ID  CD Title (by: Artist)

1   Breaker (by:Broken)
1   Dance Dance (by:FOB)
=====================================
Which ID would you like to delete? 1
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Dance Dance (by:FOB)
=====================================
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

(base) Andys-MacBook-Pro:assignment08 andyyeo$ clear

(base) Andys-MacBook-Pro:assignment08 andyyeo$ cd desktop/
assignment08
-bash: cd: desktop/assignment08: No such file or directory
(base) Andys-MacBook-
Pro:assignment08 andyyeo$ python Assignment_08_Starter.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]:
a

Enter ID: 1
What is the CD's title? Dance Dance
```

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

1   Dance Dance (by:FOB)
=======================================
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)

1   Dance Dance (by:FOB)
=======================================
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]:
d

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

1   Dance Dance (by:FOB)
=======================================
Which ID would you like to delete? 1
======= The Current Inventory: =======
ID  CD Title (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]:
l

WARNING: If you continue, all unsaved data will be lost and the I
nventory re-loaded from file.
type 'yes' to continue and reload from file. otherwise reload wil
l be canceledyes
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Dance Dance (by:FOB)
=======================================
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

(base) Andys-MacBook-Pro:assignment08 andyyeo$
```