Sheena Harms

Dec. 4, 2022

IT FDN 110 B

Assignment 08

Objects

# Introduction

I will explain the layout for the assignment 8 script. I will highlight how it has changed and what my greatest stumbling blocks were.

# CD inventory Redux

I used a lot of code from the last two assignments for this week’s assignment. I dropped the use of pickling since this assignment required the use of a .txt file again. The biggest change might have been dropping the option of deleting a record from the menu. I spent a lot of time setting that up previously. The code is still laid out by Data, Processing, IO, and main body.

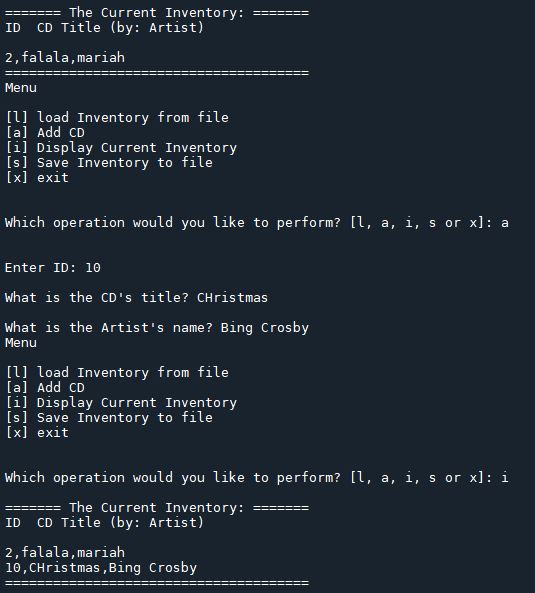


Figure : Working code.

The Data section contains a CD class and two definitions. The \_\_init\_\_ def shows the three fields associated with the CD inventory. The display def is used when the user wants to see the inventory, returning a string with the properties of the CD.

The Processing section processes the data to and from a text file. The definitions within the FileIO class are for loading or saving the inventory. I added exception handling around loading the inventory, so that if a record in the text file contained a string for the ID it would not load, and the user would be notified about it.

The Presentation section is represented by the IO class and several staticmethod definitions. The print menu definition displays menu choices upon opening the script. The menu\_choice method takes the users input, choosing from l (load), a (add), i (show inventory), s (save), or x (end program). Show\_inventory shows the user what is in the current inventory. Add\_cd gathers inventory inputs from the user.

The main body is extremely streamlined at this point. It begins with the loading the inventory from the FileIO class. A while loop prints the menu and breaks the script if the user selects x. Selecting i shows the current inventory via the for loop of the display def which is nested inside the show\_inventory method.

To add data the IO object runs the add\_cd method. The s (save) and l (load) options call the FileIO object and run either save\_inventory or load\_inventory, respectively.

# Parent and Child Classes

The show\_inventory method caused the most trouble for me this week. I couldn’t get the row.display to work. I was trying to get it to pull the print string from the display in the CD class. I was trying to emulate what I had seen on the RealPython website about Parent and Child classes. I tried to use \_\_str\_\_(self): with a return that would display the CD ID, artist, and title. I didn’t have clear instructions on how to call that from the show\_inventory definition. At first I tried to use CD.display() but it was missing “self.” However when I added “self” it gave me an error saying that “self” wasn’t defined. In the end I used row.display() in the for loop and it worked perfectly.

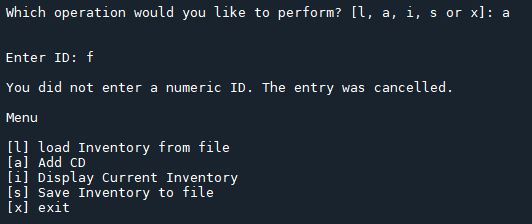


Figure 2 – Exception handling – integers.

At this point I am most confused by how many definitions there are, and how everything fits together. As the script was built out, becoming exponentially more complicated, I had real trouble keeping track of what I had named everything and how all the pieces work in tandem. It feels like I am juggling too many balls and starting to lose the things I thought I knew. I wish I had more time to slow down and re-apply knowledge in different ways.

# Summary

Another week, another CD inventory. I’m not sure how it can continue to be so difficult to build iterations of the same thing every week. Once again, the final code works but it was a real struggle to get there.

\*See python and knowledge document for assignment 8 at <https://github.com/sheenahar/Assignment_08>

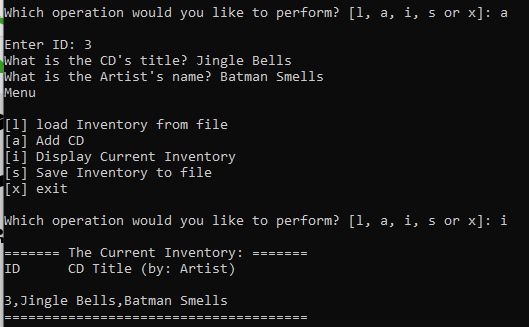


Figure 3: Assignment\_08 script in terminal.

# Appendix

## Listing Assignment\_08.py

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183 | *#------------------------------------------#*  *# Title: Assignmen08.py*  *# Desc: Assignnment 08 - Working with classes*  *# Change Log: (Who, When, What)*  *# DBiesinger, 2030-Jan-01, created file*  *# DBiesinger, 2030-Jan-01, added pseudocode to complete assignment 08*  *# SHarms, 2022-Nov-30, Added code*  *# SHarms, 2022-Dec-04, Fixed inv display issue. Added pseudocode.*  *#------------------------------------------#*  *# -- 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:*  *\_\_init\_\_(self, id, title, artist):*  *display(self): prints each data row to user.*  *"""*  **def** \_\_init\_\_(self, id, title, artist):  *"""Schema for how CD data is stored*  *Args: None*  *Returns: None"""*    self.cd\_id = id  self.cd\_title = title  self.cd\_artist = artist  **def** display(self):  *"""Displays CD data*  *Args: None*  *Returns: prints a string"""*    print (str(self.cd\_id) + ',' + self.cd\_title + ',' + self.cd\_artist)  *# -- PROCESSING -- #*  **class** **FileIO**:  *"""Processes data to and from file:*  *properties:*  *methods:*  *load\_inventory(file\_name): -> (a list of CD objects)*  *save\_inventory(file\_name, lst\_Inventory): -> None*  *"""*  @staticmethod  **def** load\_inventory(file\_name):  *"""Function to manage data ingestion from file to a list of dictionaries*  *Reads the data from file identified by file\_name into a list of objects.*  *Args:*  *file\_name (string): name of file used to read the data from*  *Returns:*  *None.*  *"""*  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(',')  **try**:  objCD = CD(int(data[0]), data[1], data[2])  **except** **ValueError**:  print ("An ID was found to not be an integer. Inventory failed to load past that record.")  **else**:  lstOfCDObjects.append(objCD)  objFile.close()  @staticmethod  **def** save\_inventory(file\_name, lst\_Inventory):  *"""Writes from list to a file.*  *Args:*  *file\_name: file name*  *lst\_Inventory: list of CD objects*  *Returns:*  *None"""*  objFile = open(file\_name, 'w')  **for** row **in** lst\_Inventory:  strCD = str(row.cd\_id) + ',' + row.cd\_title + ',' + row.cd\_artist  objFile.write(strCD + '**\n**')  objFile.close()  *# -- PRESENTATION (Input/Output) -- #*  **class** **IO**:  *"""Menu and user choices:*  *properties:*  *methods:*  *print\_menu(): -> None*  *menu\_choice(): -> choice string*  *show\_inventory(table):-> None*  *add\_CD():-> None*  *"""*  @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 that holds the data during runtime.*  *Returns:*  *None.*  *"""*  print('======= The Current Inventory: =======')  print('ID**\t**CD Title (by: Artist)**\n**')  **for** row **in** table:  row.display()  print('======================================')  @staticmethod  **def** add\_CD():  *"""Asks user for cd data inputs*  *Args:*  *None.*  *Returns:*  *None.*  *"""*  **try**:  intID = int(input('Enter ID: ').strip())  **except** **ValueError**:  print("**\n**You did not enter a numeric ID. The entry was cancelled.**\n**")  **else**:  strTitle = input('What is the CD**\'**s title? ').strip()  strArtist = input('What is the Artist**\'**s name? ').strip()  objCD = CD(intID, strTitle, strArtist)  lstOfCDObjects.append(objCD)  *# -- Main Body of Script -- #*  FileIO.load\_inventory(strFileName)  *# Display menu to user*  **while** **True**:  IO.print\_menu()  strChoice = IO.menu\_choice()  **if** strChoice == 'x':  **break**  *# show user current inventory*  **elif** strChoice == 'i':  IO.show\_inventory(lstOfCDObjects)  **continue** *# start loop back at top.*  *# let user add data to the inventory*  **elif** strChoice == 'a':  IO.add\_CD()  **continue**  *# let user save inventory to file*  **elif** strChoice == 's':  FileIO.save\_inventory(strFileName, lstOfCDObjects)  **continue**  *# let user load inventory from file*  **elif** strChoice == 'l':  FileIO.load\_inventory(strFileName)  **continue** |