Daniel Fredin

November 18, 2022

Foundations of Programming: Python

Assignment 06

CD Inventory Script – Modified with classes and functions

# Introduction

The purpose of this week’s assignment was to again modify the CD Inventory script but this time to implement classes and functions instead of having the main loop containing all the executable code. Once we were able to execute the modified, we had to make use of Github and upload our script and this knowledge document to our new repository, Assignement\_06.

# Starting code

When I first began to execute the code before changing any of the lines of code into functions, I realized that step one was to read in the currently saved CDInventory.txt file. However, since I did not have any text file created it was throwing a FileNotFoundError: [Errno 2] No such file or directory. This reminded from last weeks assignment that I needed to check if a file existed or not before reading in the file data. So, recycling some script from last weeks assignment, I imported the os.path module and created a test as shown in Fig. 1. If the file existed, then the program would continue to open the file in ‘read’ mode and then read the data into the program. However, if the file did not exist, then I had the program open the file in ‘write’ mode so that it would create an empty file named CDInventory.txt in the directory where the python script was executed. This prevented the FileNotFoundError from being thrown and I was able to continue running the program.

Text

Description automatically generated

Figure - Check to confirm or deny that a text file exists. If confirmed, then program reads from file while if denied then the program creates a blank file.

# Functions

## *I/O*

In the design of the functions, I chose to attempt the lines of code that did not manipulate the data. So therefore, I started with IO function. Since we were not passing any parameters into the function, only returning what the user inputted, I defined the function user\_input() with no arguments. I did have the function return three values in the form of a tuple, ID, title, and artist shown in Fig. 2. Then I passed these values into the script’s variables strID, strTitle, and strArtist as shown in Fig. 3.

Text

Description automatically generated

Figure - Definition of user\_input function which gathered the input of ID, title, and artist from the user.



Figure - Passing the tuple of return values of function user\_input to the variables that are used throughout the script: strID, strTitle, and strArtist.

## *Data processing*

The functions for adding data and deleting data were straightforward. I used the same code that was given in the starter script but instead of the user\_input function, I passed arguments into the function for processing. For example, in the add\_data function, I passed the local variable arguments ID, title, and artist which correspond to the global variables strID, strTitle, and strArtist, shown in Fig. 4. After that, I had the function continue with the original code of assigning each variable to a dicRow and then appended that dicRow to the global lstTbl as shown in Fig. 5.



Figure - Passing three global variables into the add\_data function.

Text

Description automatically generated

Figure - Function add\_data with passed arguments and append the global 2D data structure lstTbl.

## *File processing*

Again, the function for writing to file was straightforward. I used the provided code for writing data to a file but within the function write\_file, I passed the arguments the local variables file\_name and table which correspond to the global variables strFileName and lstTbl as shown in Fig. 6. Once the function had the global variables passed to its arguments, it opened the file in ‘write’ mode and continued row by row saving the data into a text file.



Figure - Passing global variables into write\_file function.

# Execution of script

I executed the script in both Spyder and a python terminal window as shown in below in the following pages.

# *Running in Spyder*

Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]

Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

Restarting kernel...

|  |
| --- |
|  |

In [**1**]: runfile('C:/\_FDProgramming/Assignment\_06/CDInventory.py', wdir='C:/\_FDProgramming/Assignment\_06')

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? Thriller

What is the Artist's name? Michael Jackson

======= The Current Inventory: =======

ID CD Title (by: Artist)

1 Thriller (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: 2

What is the CD's title? Back in Black

What is the Artist's name? AC/DC

======= The Current Inventory: =======

ID CD Title (by: Artist)

1 Thriller (by: Michael Jackson)

2 Back in Black (by: AC/DC)

======================================

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 Thriller (by: Michael Jackson)

2 Back in Black (by: AC/DC)

======================================

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]: a

Enter ID: 33

What is the CD's title? Nevermind

What is the Artist's name? Nirvana

======= The Current Inventory: =======

ID CD Title (by: Artist)

1 Thriller (by: Michael Jackson)

2 Back in Black (by: AC/DC)

33 Nevermind (by: Nirvana)

======================================

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 Thriller (by: Michael Jackson)

2 Back in Black (by: AC/DC)

33 Nevermind (by: Nirvana)

======================================

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

The inventory was NOT saved to file. Press [ENTER] to return to the menu.

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 Inventory re-loaded from file.

type 'yes' to continue and reload from file. otherwise reload will be canceled. yes

reloading...

======= The Current Inventory: =======

ID CD Title (by: Artist)

1 Thriller (by: Michael Jackson)

2 Back in Black (by: AC/DC)

======================================

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? Nevermind

What is the Artist's name? Nirvana

======= The Current Inventory: =======

ID CD Title (by: Artist)

1 Thriller (by: Michael Jackson)

2 Back in Black (by: AC/DC)

3 Nevermind (by: Nirvana)

======================================

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 Thriller (by: Michael Jackson)

2 Back in Black (by: AC/DC)

3 Nevermind (by: Nirvana)

======================================

Which ID would you like to delete? 2

The CD was removed

======= The Current Inventory: =======

ID CD Title (by: Artist)

1 Thriller (by: Michael Jackson)

3 Nevermind (by: Nirvana)

======================================

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 Thriller (by: Michael Jackson)

3 Nevermind (by: Nirvana)

======================================

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 Thriller (by: Michael Jackson)

3 Nevermind (by: Nirvana)

======================================

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

In [**2**]:

# *Running in python terminal window*

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

# *CD Inventory text file*

Graphical user interface, text

Description automatically generated

# Github

My Github address is located at this [website](https://github.com/dfredin/Assignment_06) (external reference).[[1]](#footnote-1)

# Summary

The goal of this week’s assignment was to implement the use of classes and functions in a premade CD Inventory script in stead of having all the executable code contained with in the main loop. I found that using classes and functions made the script very nice and easy to read. In stead of having to meticulously go line by line through the main loop to understand what was happening, with the use of functions, I was able to see a more defined and understandable structure. While it was difficult at first to figure out how to pass parameters into the functions with out passing the global variables, I found that functions definitely simplified the complexity of my script and am looking forward to implementing them in future programming scripts.

# Appendix

Using [saravji’s website](https://www.saravjishut.org/syntax) (external reference) [[2]](#footnote-2)to properly highlight my source code for CDInventory.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  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235 | *#------------------------------------------#*  *# Title: Assignment06\_Starter.py*  *# Desc: Working with classes and functions.*  *# Change Log: (Who, When, What)*  *# DBiesinger, 2030-Jan-01, Created File*  *# Dfredin, 2022-Nov-18, Modified file by adding specific functions*  *#------------------------------------------#*  *# -- DATA -- #*  strChoice = '' *# User input*  lstTbl = [] *# list of lists to hold data*  dicRow = {} *# list of data row*  strFileName = 'CDInventory.txt' *# data storage file*  objFile = **None** *# file object*  flag = bool() *# boolean flag for file check*  **import** **os.path** *# checks for if filepath exists*  *# -- PROCESSING -- #*  **class** **DataProcessor**:    @staticmethod  **def** add\_data(ID, title, artist):  *"""Takes the user's input for ID, Title, and artist and appends to table lstTbl.*    *Args:*  *ID (int): integer number that describes the ID for the CD entry.*  *title (string): string of the title of the CD.*  *artist (string): string of the name of the artist of the CD.*  *Returns:*  *None.*  *"""*  intID = int(ID)  dicRow = {'ID': intID, 'Title': title, 'Artist': artist}  lstTbl.append(dicRow)    @staticmethod  **def** delete\_data(IDdel, table):  *"""Deletes the row that the user selcted with IDdel*  *Args:*  *IDdel (int): ID of CD row that the user wants deleted.*  *table (list of dicts): 2D data structure that holds the rows of CDs in a list.*    *Returns:*  *None.*  *"""*  intRowNr = -1  blnCDRemoved = **False**  **for** row **in** table:  intRowNr += 1  **if** row['ID'] == IDdel:  **del** table[intRowNr]  blnCDRemoved = **True**  **break**  **if** blnCDRemoved:  print('The CD was removed')  **else**:  print('Could not find this CD!')    **class** **FileProcessor**:  *"""Processing the data to and from text file"""*  @staticmethod  **def** read\_file(file\_name, table):  *"""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.*  *"""*  table.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]}  table.append(dicRow)  objFile.close()  @staticmethod  **def** write\_file(file\_name, table):  *"""Function that overwrites the new data input by the user into the named txt file*  *Args:*  *file\_name (string): name of file used to write the data to.*  *table (list): data structure that holds the data during runtime.*  *Returns:*  *None.*  *"""*  objFile = open(file\_name, 'w')  **for** row **in** table:  lstValues = list(row.values())  lstValues[0] = str(lstValues[0])  objFile.write(','.join(lstValues) + '**\n**')  objFile.close()    *# -- PRESENTATION (Input/Output) -- #*  **class** **IO**:  *"""Handling Input / Output"""*  @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('[d] Delete CD from Inventory**\n**[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, d, s or x*  *"""*  choice = ' '  **while** choice **not** **in** ['l', 'a', 'i', 'd', 's', 'x']:  choice = input('Which operation would you like to perform? [l, a, i, d, s or x]: ').lower().strip()  print() *# Add extra space for layout*  **return** choice  @staticmethod  **def** show\_inventory(table):  *"""Displays current inventory table*  *Args:*  *table (list of dict): 2D data structure (list of dicts) that holds the data during runtime.*  *Returns:*  *None.*  *"""*  print('======= The Current Inventory: =======')  print('ID**\t**CD Title (by: Artist)**\n**')  **for** row **in** table:  print('**{}\t{}** (by: **{}**)'.format(\*row.values()))  print('======================================')    @staticmethod  **def** user\_input():  *"""Gets user input for adding CDs to inventory (ID, title, and artist)*    *Args:*  *None.*  *Returns:*  *ID (int): an integer the user inputs for CD ID.*  *title (string): a string of the CD title name.*  *artist (string): a string of the artist's name.*  *"""*  ID = input('Enter ID: ').strip()  title = input('What is the CD**\'**s title? ').strip()  artist = input('What is the Artist**\'**s name? ').strip()  **return** ID, title, artist  *# 1. When program starts, read in the currently saved Inventory*  flag = os.path.isfile(strFileName) *# Checks to see if file exists*  **if** flag: *# if the file exists, then it reads and loads the data*  FileProcessor.read\_file(strFileName, lstTbl)  **else**: *# if the file does not exist, it creates an empty file*  FileProcessor.write\_file(strFileName, lstTbl)  *# 2. start main loop*  **while** **True**:  *# 2.1 Display Menu to user and get choice*  IO.print\_menu()  strChoice = IO.menu\_choice()  *# 3. Process menu selection*  *# 3.1 process exit first*  **if** strChoice == 'x':  **break**  *# 3.2 process load inventory*  **if** strChoice == 'l':  print('WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.')  strYesNo = input('type **\'**yes**\'** to continue and reload from file. otherwise reload will be canceled. ')  **if** strYesNo.lower() == 'yes':  print('reloading...')  FileProcessor.read\_file(strFileName, lstTbl)  IO.show\_inventory(lstTbl)  **else**:  input('canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')  IO.show\_inventory(lstTbl)  **continue** *# start loop back at top.*  *# 3.3 process add a CD*  **elif** strChoice == 'a':  *# 3.3.1 Ask user for new ID, CD Title and Artist*  strID, strTitle, stArtist = IO.user\_input()  *# 3.3.2 Add item to the table*  DataProcessor.add\_data(strID, strTitle, stArtist)  IO.show\_inventory(lstTbl)  **continue** *# start loop back at top.*  *# 3.4 process display current inventory*  **elif** strChoice == 'i':  IO.show\_inventory(lstTbl)  **continue** *# start loop back at top.*  *# 3.5 process delete a CD*  **elif** strChoice == 'd':  *# 3.5.1 get Userinput for which CD to delete*  *# 3.5.1.1 display Inventory to user*  IO.show\_inventory(lstTbl)  *# 3.5.1.2 ask user which ID to remove*  intIDDel = int(input('Which ID would you like to delete? ').strip())  *# 3.5.2 search thru table and delete CD*  DataProcessor.delete\_data(intIDDel, lstTbl)  IO.show\_inventory(lstTbl)  **continue** *# start loop back at top.*  *# 3.6 process save inventory to file*  **elif** strChoice == 's':  *# 3.6.1 Display current inventory and ask user for confirmation to save*  IO.show\_inventory(lstTbl)  strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()  *# 3.6.2 Process choice*  **if** strYesNo == 'y':  *# 3.6.2.1 save data*  FileProcessor.write\_file(strFileName, lstTbl)  **else**:  input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')  **continue** *# start loop back at top.*  *# 3.7 catch-all should not be possible, as user choice gets vetted in IO, but to be save:*  **else**:  print('General Error') |

1. Retrieved 18 Nov 2022 (https://github.com/dfredin/Assigment\_06) [↑](#footnote-ref-1)
2. Retrieved 18 Nov 2022 [↑](#footnote-ref-2)