Daniel Fredin

November 23, 2022

Foundations of Programming: Python

Assignment 07

CD Inventory Script – Modified with error handling

# Introduction

This week’s assignment involved modifying our previous weeks CD Inventory script with the addition of structured error handling and using binary data files instead of our usual text files. In addition to that we learned about the different options there are for reading data from files such as the functions readlines() and the readline() along with using those functions in different loops such as a for loop or a while loop.

# Binary files

I found working with binary files to be much easier to handle than working with text files. No longer did I need to implement a for/in loop while reading and saving data to a text file to ensure that the data was in a comma separated value format. By using binary files instead of text files, it reduced my code significantly. For example, in my write\_file() function for Assignment\_06 it was 6 lines however in this weeks assignment, it was reduced to only 2 lines, shown in Fig. 1.

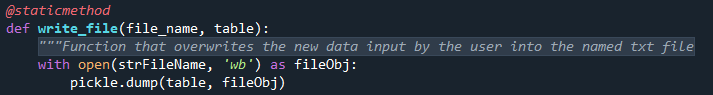


Figure - The reduction in the amount of lines of code for saving to files while using binary files.

## *Website for pickling*

I used the official python documentation for learning more about pickling and how to utilize the dump() and load() functions. The website I used is located [here](https://docs.python.org/3/library/pickle.html) (external reference)[[1]](#footnote-1).

# Error handling

I did not realize it but I was already trying to take care of error handling in the previous week’s assignment by importing the os.path module and executing the search for if the file exists or not with os.path.isfile() function but I found it much easier to understand the script when it was executed with the try/except blocks. I found that in some cases it was beneficial to implement error handling within the function definitions but other times it was more useful to implement them in the main loop of the code. For example, when implementing the ValueError I entered the try/except within the definition of the function for adding new CD’s since that is where the user input was taking place shown in Fig. 2. Also, for the deletion of CD’s the user input was taking place within the main loop of the code so that is where I implemented it’s try/except shown in Fig. 3.

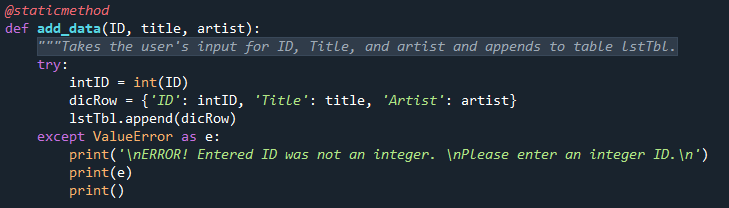


Figure - Try/Except block used within the definition of the add\_data function.

Text

Description automatically generated

Figure - Try/Except block used within the main loop of the code where the user enters the ID they want to delete.

## *Website for exception handling*

I also used the official python documentation for understanding more about error handling and how to implement the different types of exceptions. The website I used is located [here](https://docs.python.org/3.9/tutorial/errors.html) (external reference)[[2]](#footnote-2).

# 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.

In [**1**]: runfile('C:/\_FDProgramming/Assignment\_07/CDInventory.py', wdir='C:/\_FDProgramming/Assignment\_07')

ERROR! Data file not found!

An empty file has now been created.

[Errno 2] No such file or directory: 'CDInventory.dat'

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: e

What is the CD's title? Nevermind

What is the Artist's name? Nirvana

ERROR! Entered ID was not an integer.

Please enter an integer ID.

invalid literal for int() with base 10: 'e'

======= 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? w

ERROR! Entered ID was not an integer.

Please enter an integer ID for deletion.

invalid literal for int() with base 10: 'w'

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

# Github

My Github address is located at this [website](https://github.com/dfredin/Assignment_07) (external reference).[[3]](#footnote-3)

# Summary

For this week’s assignment, our goal was to modify our previous weeks CD Inventory to include the use of binary files as well as undertaking structured error handling. I found that I am becoming much better at diagnosing errors throughout my code as I had already found way to handling my FileNotFoundError from the previous week’s assignment. Its hard to believe that the scripts that I am writing and modifying are becoming this complex in just seven weeks compared to our introductory assignment script. I feel that the use of binary files is much more efficient way to save and access data since we do not have to go through the process of creating comma separated values in our text file as well as loading each value from a file with a loop. The pickle.dump and the pickle.load functions are a much easier way of utilizing data while keeping them in the data structure that we designed for them, in this case, the 2D list of dictionaries. I am glad that we started to use structed error handling because up until now I’ve been trying to figure out ways to keep my program running if the user did not follow the directions of if the data file was not already created. I am looking forward to implementing these techniques in further detail with our future assignments.

# Appendix

Using [saravji’s website](https://www.saravjishut.org/syntax) (external reference) [[4]](#footnote-4)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  236  237  238  239  240  241  242  243  244 | *#------------------------------------------#*  *# 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*  *# Dfredin, 2022-Nov-23, Modified file for error handling and use of binary files*  *#------------------------------------------#*  *# -- DATA -- #*  strChoice = '' *# User input*  lstTbl = [] *# list of lists to hold data*  dicRow = {} *# list of data row*  strFileName = 'CDInventory.dat' *# data storage file*  objFile = **None** *# file object*  flag = bool() *# boolean flag for file check*  **import** **pickle** *# allows storage to .dat files*  *# -- 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.*  *"""*  **try**:  intID = int(ID)  dicRow = {'ID': intID, 'Title': title, 'Artist': artist}  lstTbl.append(dicRow)  **except** **ValueError** **as** e:  print('**\n**ERROR! Entered ID was not an integer. **\n**Please enter an integer ID.**\n**')  print(e)  print()    @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:*  *table (list of dict): 2D data structure that contains the file data.*  *"""*  **with** open(strFileName, 'rb') **as** fileObj:  table = pickle.load(fileObj)  **return** table    @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.*  *"""*  **with** open(strFileName, 'wb') **as** fileObj:  pickle.dump(table, fileObj)    *# -- 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*  **try**:  lstTbl = FileProcessor.read\_file(strFileName, lstTbl) *# File not found error handling*  **except** **FileNotFoundError** **as** e:  print('**\n**ERROR! Data file not found! **\n**An empty file has now been created.**\n**')  print(e)  print()  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...')  **try**:  lstTbl = FileProcessor.read\_file(strFileName, lstTbl) *# File not found error handling*  IO.show\_inventory(lstTbl)  **except** **FileNotFoundError** **as** e:  print('**\n**ERROR! Data file not found! **\n**An empty file has now been created.**\n**')  print(e)  FileProcessor.write\_file(strFileName, 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*  **try**:  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.*  **except** **ValueError** **as** e:  print('**\n**ERROR! Entered ID was not an integer. **\n**Please enter an integer ID for deletion.**\n**')  print(e)  print()  *# 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 23 Nov 2022 [↑](#footnote-ref-1)
2. Retrieved 23 Nov 2022 [↑](#footnote-ref-2)
3. Retrieved 23 Nov 2022 (https://github.com/dfredin/Assigment\_07) [↑](#footnote-ref-3)
4. Retrieved 23 Nov 2022 [↑](#footnote-ref-4)