Sravani Musunuri

August 20th , 2022

IT FDN 110 A Su 22: Foundations of Programming: Python

Assignment 07

Error Handling and Pickling to binary data

# Introduction

This assignment provides all the knowledge required to modify the CD Inventory Python script from last week to convert permanent data store to user binary data i.e., pickling and add structured error handling. By watching module videos and reading book chapter gave clear understanding of working with text files, reading data options, pickling, structured error handling and creating custom GitHub readme files. I used Spyder to execute all my lab programs.

# Course Work

I started with importing pickle module to the python script for converting permanent data store to use binary data format. Pickling is mainly used for serializing and deserializing the data and used for file transfer or importing bulk data for faster turnaround time. In this program two basic methods of Python pickle module are used.

* pickle.dump(obj, file) – This method is used during the pickling process. I used this method to write 2D list of dictionaries to a file in write\_file() function.
* pickle.load(obj) – This method is used during the unpickling process. I used this method to load data from binary file to an object in read\_file() function.

Extending this program to add structured error handling in areas where there is user interaction, type casting and file access operations.

* User Interaction/type casting: I added error handling when user input’s the CD Inventory details and ID to delete the CD inventory.
  + Inventory ID (strID) should be an integer and value cannot be 0 or less.
  + strTitle and strArtist cannot be NULL.
  + ID to delete the Inventory (intIDDel) should be an integer
* File access operations: Added error handling during reading and writing to the file in read\_file() and write\_file() functions.
* Error handling is also added when List (lstTbl) is empty while displaying the CD Inventory.

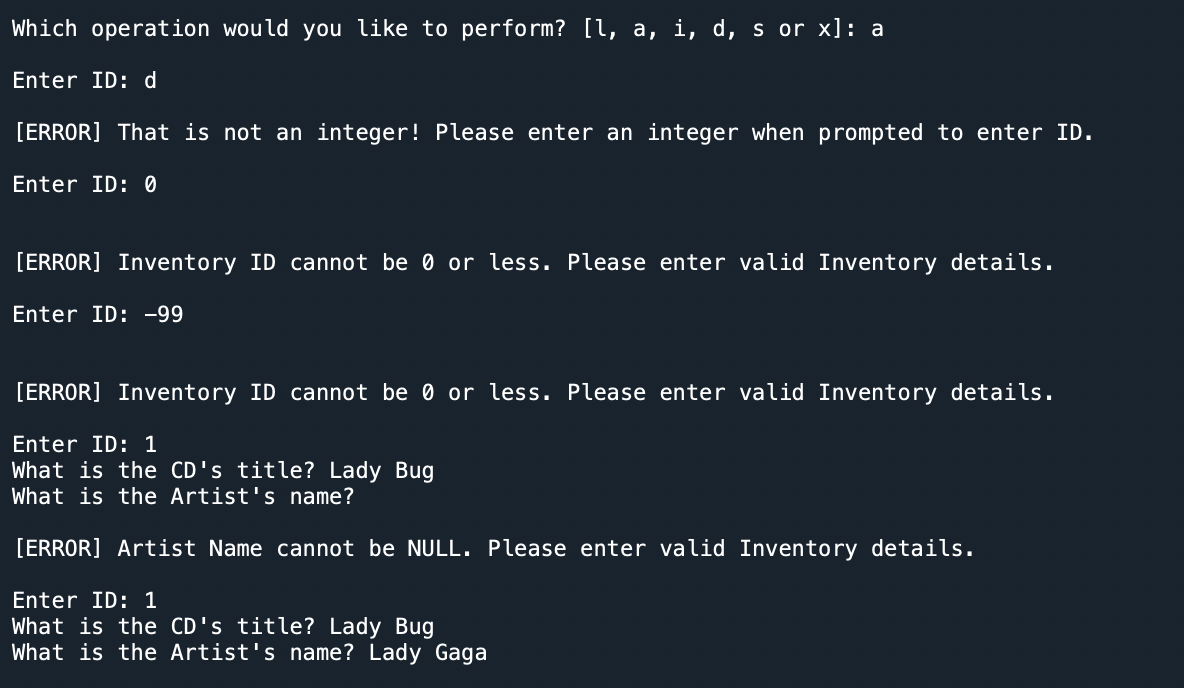
Adding few lines of code here:

1. @staticmethod
2. **def** user\_input():
3. """Gets user input for CD Inventory
5. Args:
6. None.
8. Returns:
9. strID (string): ID of CD Inventory
10. strTitle (string): Title of CD Inventory
11. strArtist (string): Artist of the CD Inventory
13. """
14. **while** True:
15. **try**:
16. strID **=** int(input('Enter ID: ').strip())
18. **if** strID <**=** 0:
19. print()
20. **raise** Exception('[ERROR] Inventory ID cannot be 0 or less. Please enter valid Inventory details.')

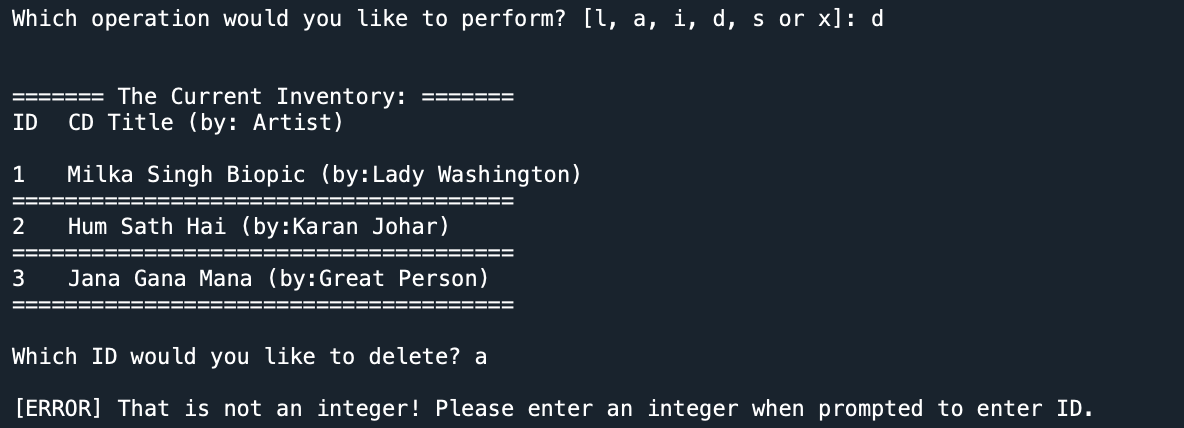
23. strTitle **=** input('What is the CD\'s title? ').strip()
25. **if** strTitle **==** "":
26. print()
27. **raise** Exception('[ERROR] CD Title cannot be NULL. Please enter valid Inventory details.')
29. strArtist **=** input('What is the Artist\'s name? ').strip()

32. **if** strArtist **==** "":
33. **raise** Exception('[ERROR] Artist Name cannot be NULL. Please enter valid Inventory details.')
35. print()
37. **return** strID,strTitle,strArtist
39. **except** ValueError:
40. print()
41. print("[ERROR] That is not an integer! Please enter an integer when prompted to enter ID.")
42. print()
44. **except** Exception as e:
45. print()
46. print(e)
47. print()

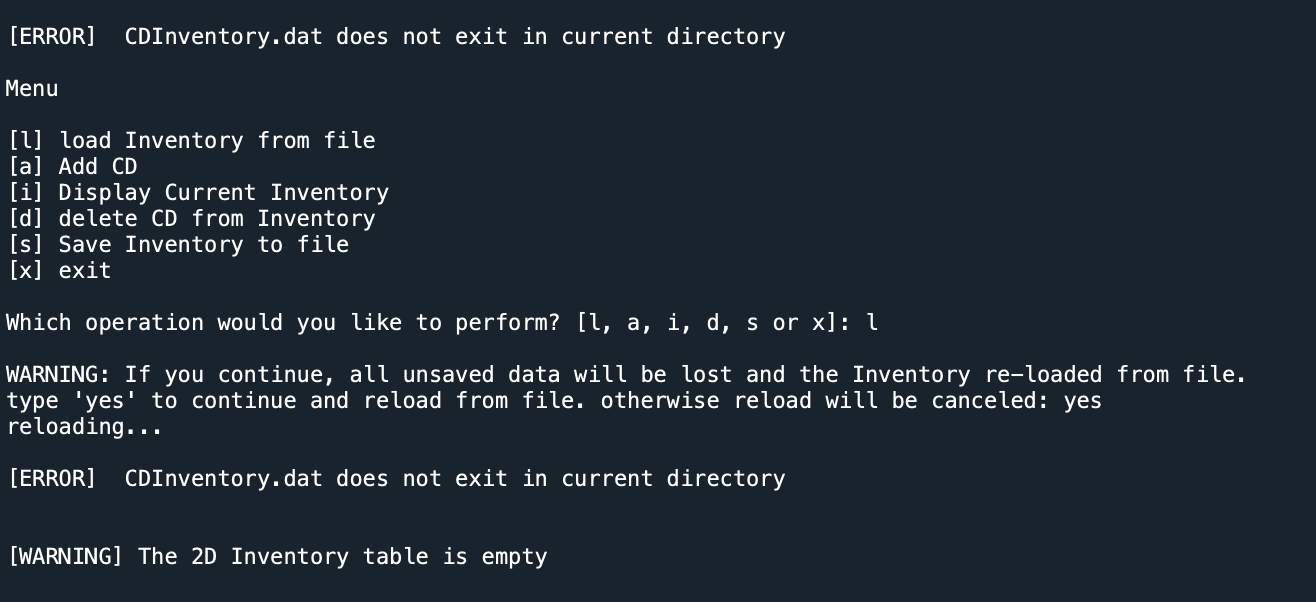
*Listing 1 - Error Handling for User Input*



*Figure 1 - Error Handling for ValueError and Custom Errors*



*Figure 2 - Error handling for ValueError*



*Figure 3 - Error Handling for FileNotFoundError and Empty List*

# Running python script on terminal

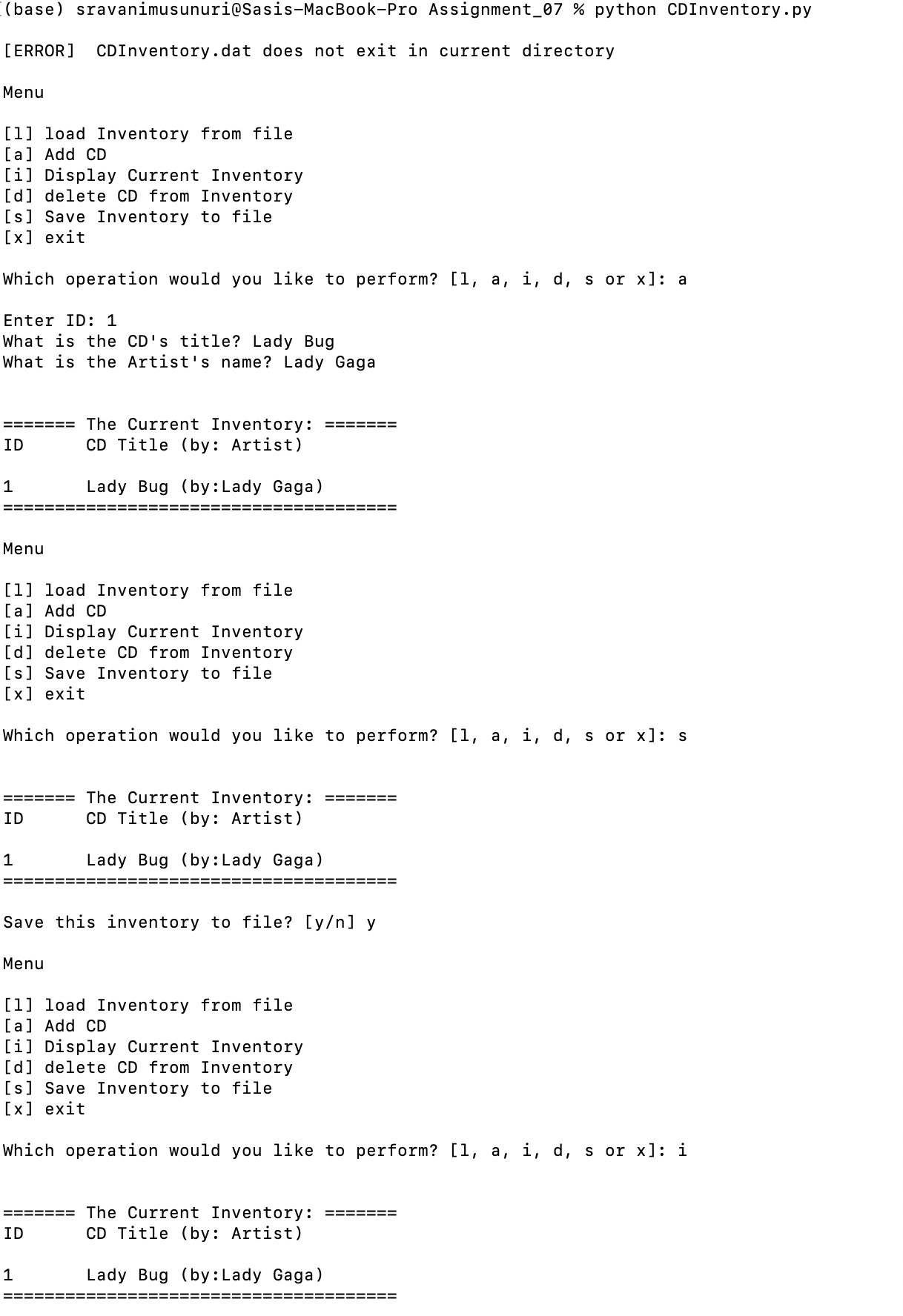


Figure 4 - Screenshot of output of python script running on terminal

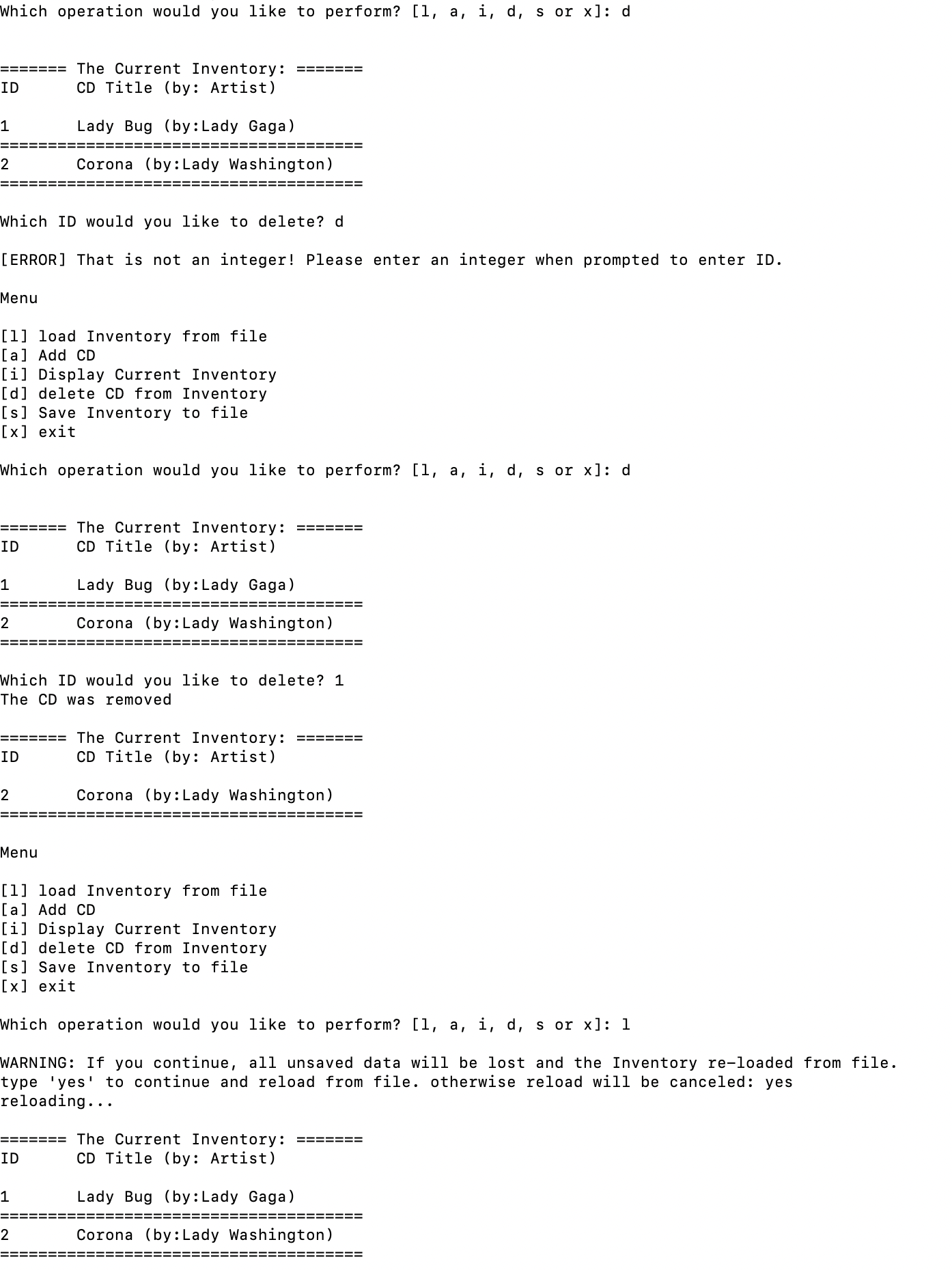
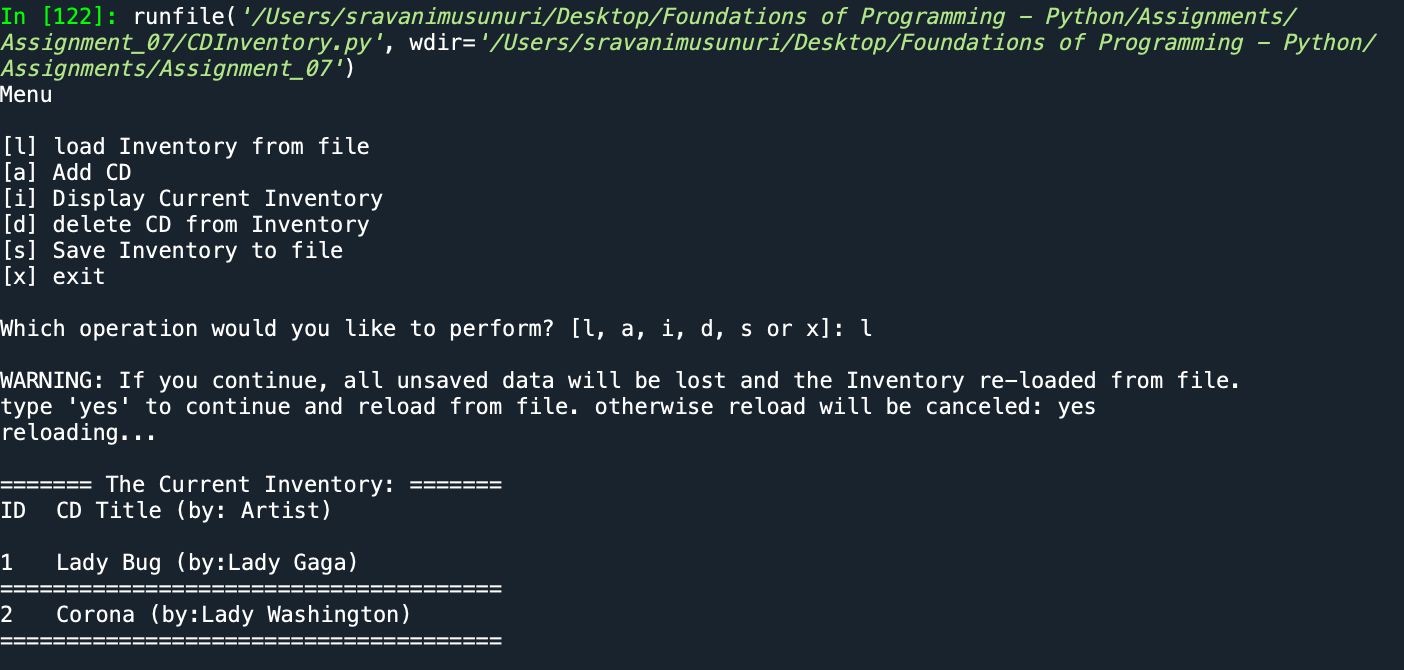
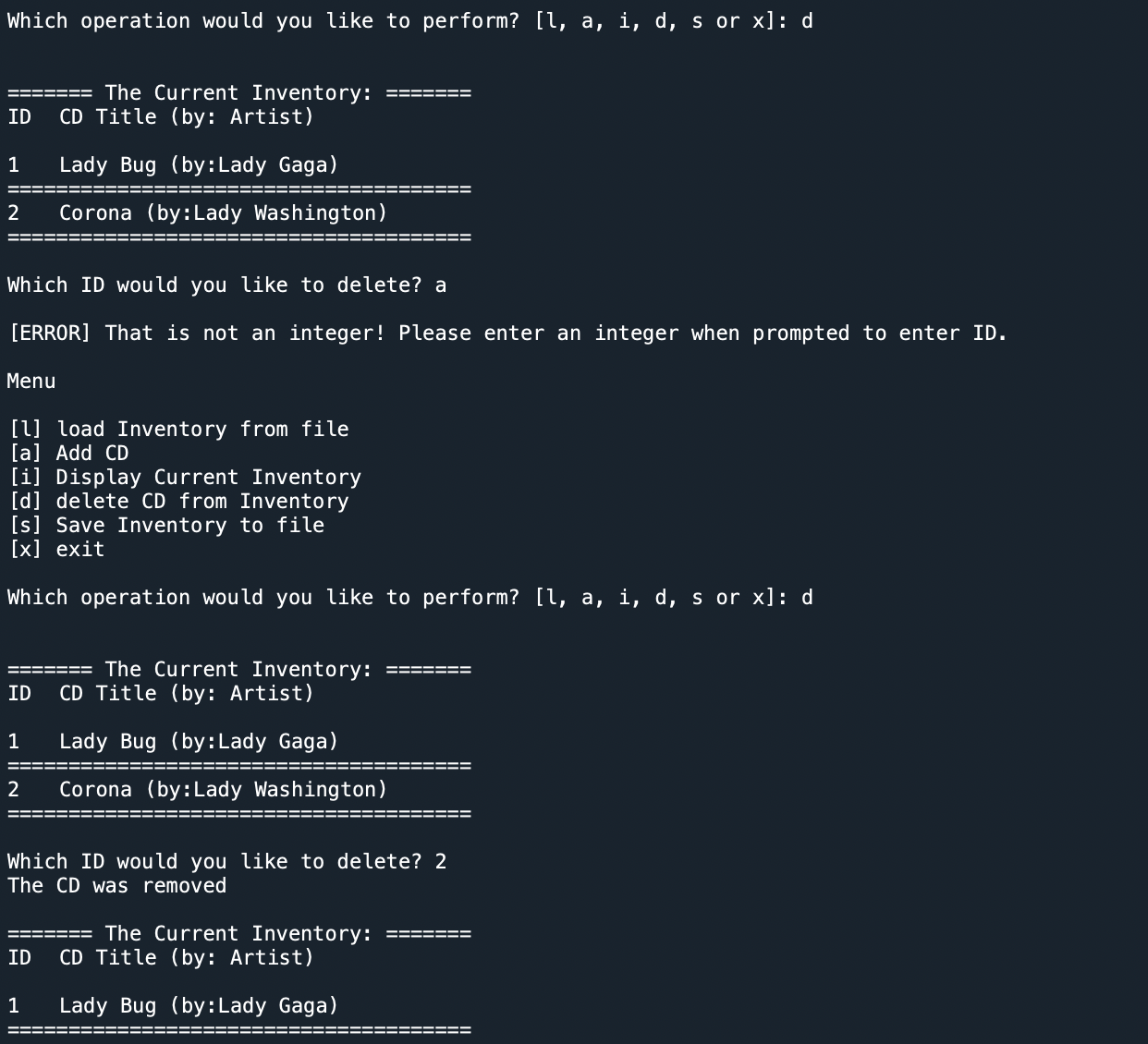


Figure 5 - Screenshot of output of python script running on terminal

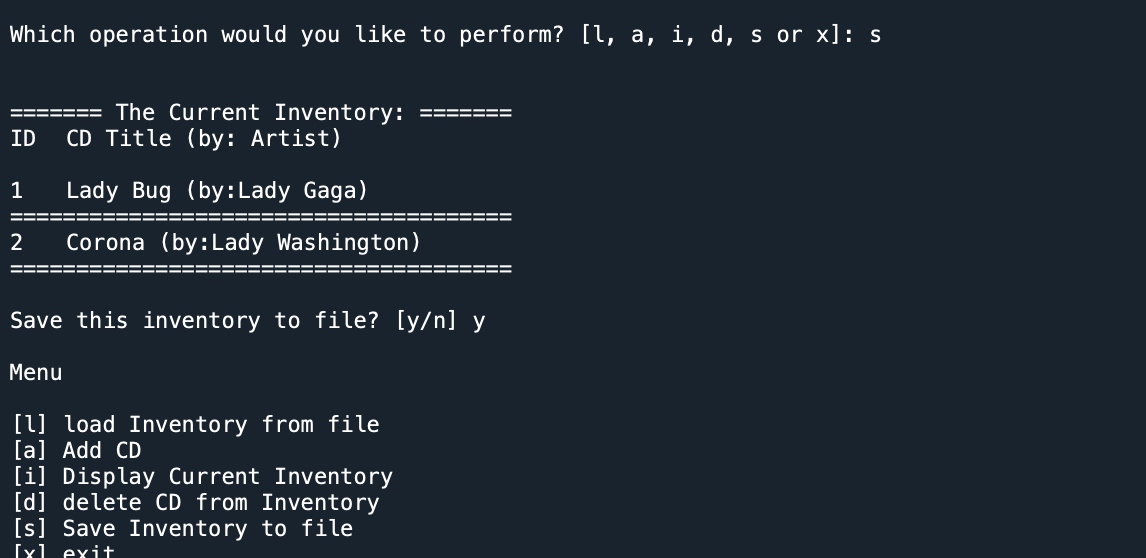
# Running the python script on Spyder



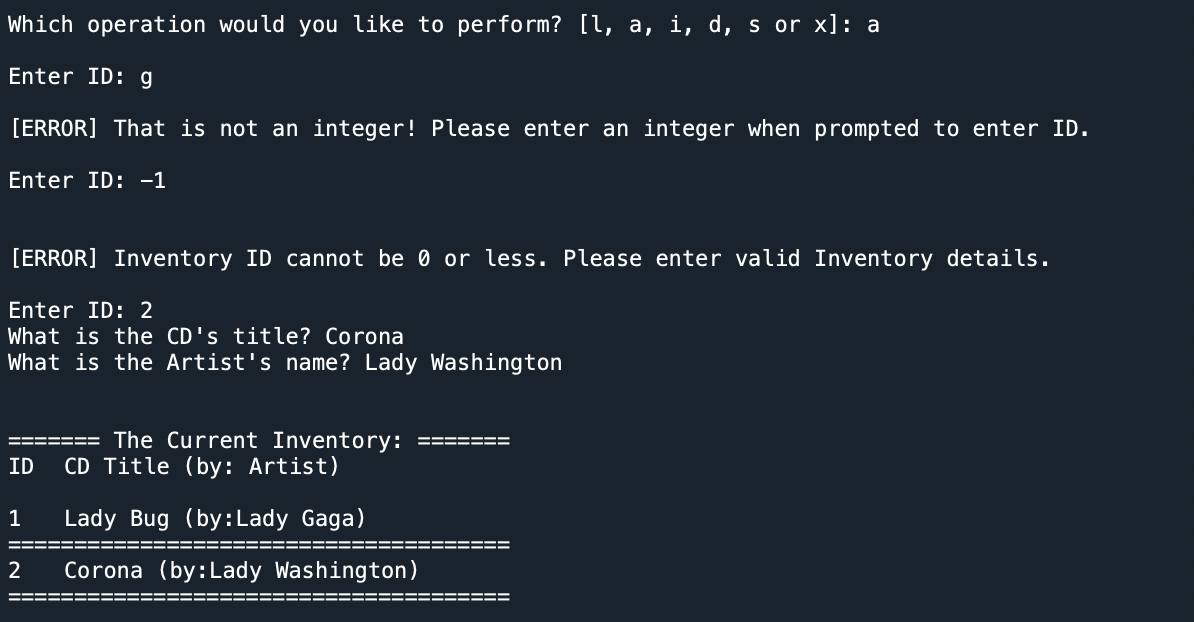
*Figure 6 - Load Inventory from file*

*.*

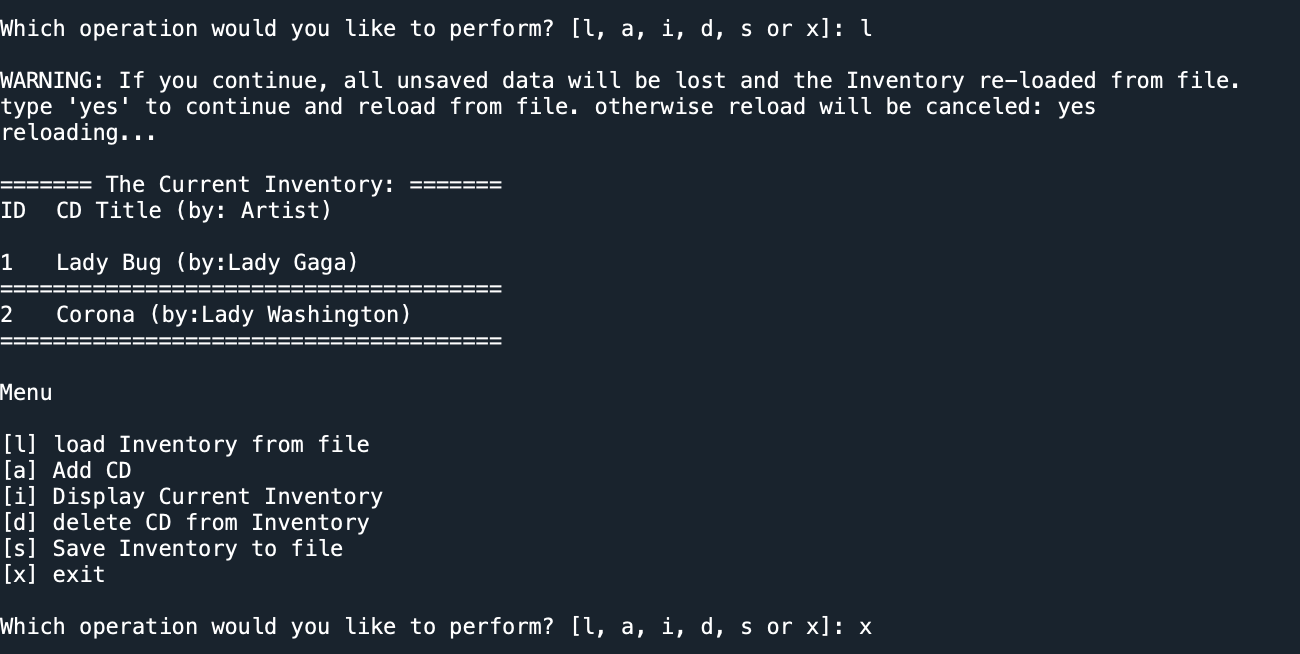
*Figure 7 - Delete CD from Inventory*

**

*Figure 8 - Save Inventory to File*

**

*Figure 9 - Add CD Inventory*

**

*Figure 10 - Display Current Inventory and Exit*

# Research Exception Handling and Pickling in Python

As recommended to search for some examples on Web, I found below links interesting:

1. <https://pythonprogramming.net/python-pickle-module-save-objects-serialization/>

2. <https://www.youtube.com/watch?v=6wSFWOleZlc>

3. <https://realpython.com/python-pickle-module/>

4. <https://www.geeksforgeeks.org/python-exception-handling/>

# GitHub Link

# <https://github.com/sravanimusunuri6/Assignment_07>

# Summary

In this module, I got chance to learn:

* Benefits for using structured error handling without crashing the program.
* Difference between text file and binary file, pickling concept
* Exception class and derive new class from the existing.