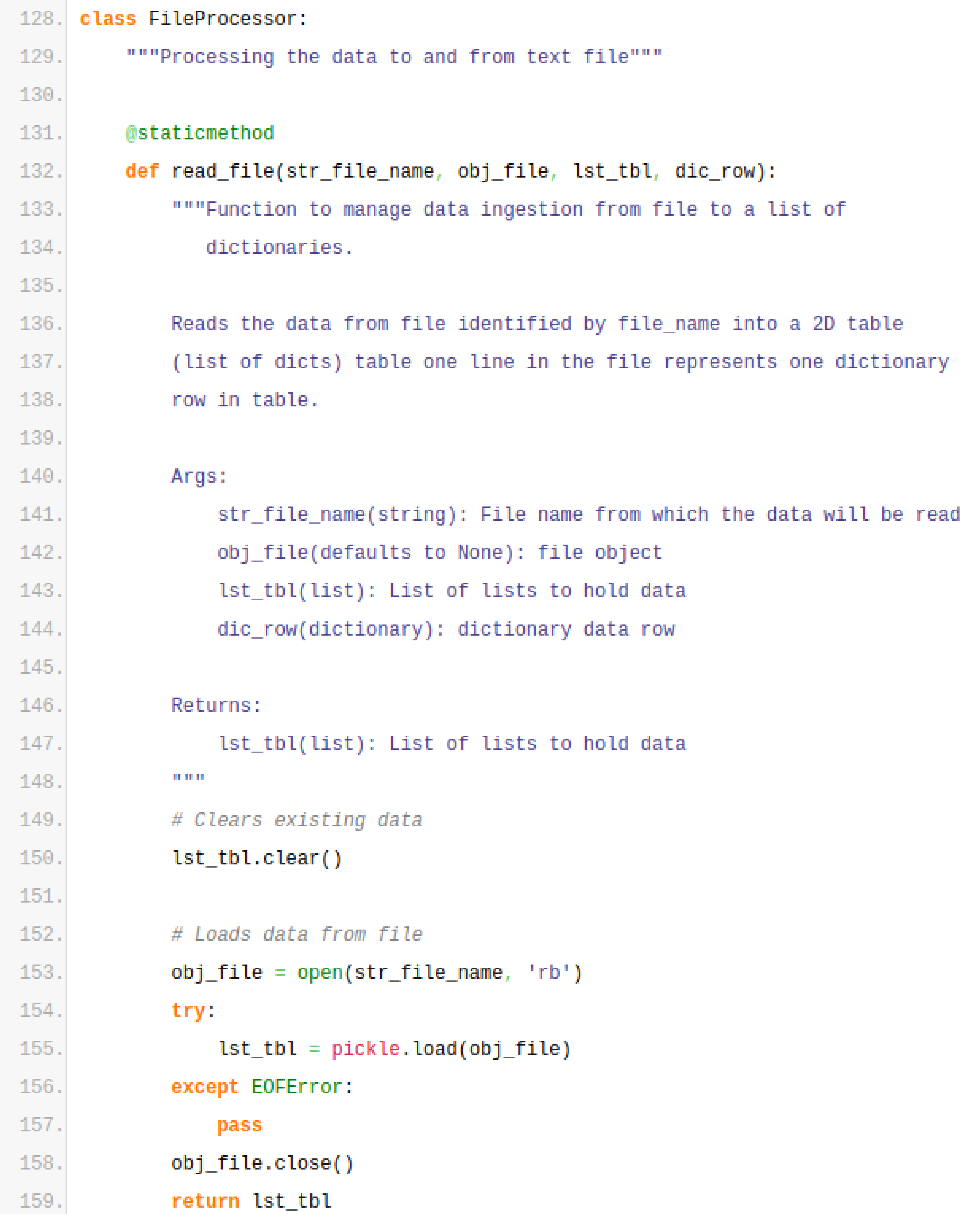
Charles Hodges(hodges11@uw.edu)
Aug 22, 2021
IT FDN 110: Introduction to Programming (Python)
Assignment 07
GitHub Link:

Error Handling, Pickling, & Shelves

Introduction

In this assignment, we were instructed to modify our last assignment, Assignment 06, and add error handling(try:except), make our file a binary file, and use either pickling or shelves for the read and Write to/from that file. To accomplish this, I added back all of the arguments and returns that I had removed during Assignment 6, so as to not rely on the global variables.

Arguments and Returns

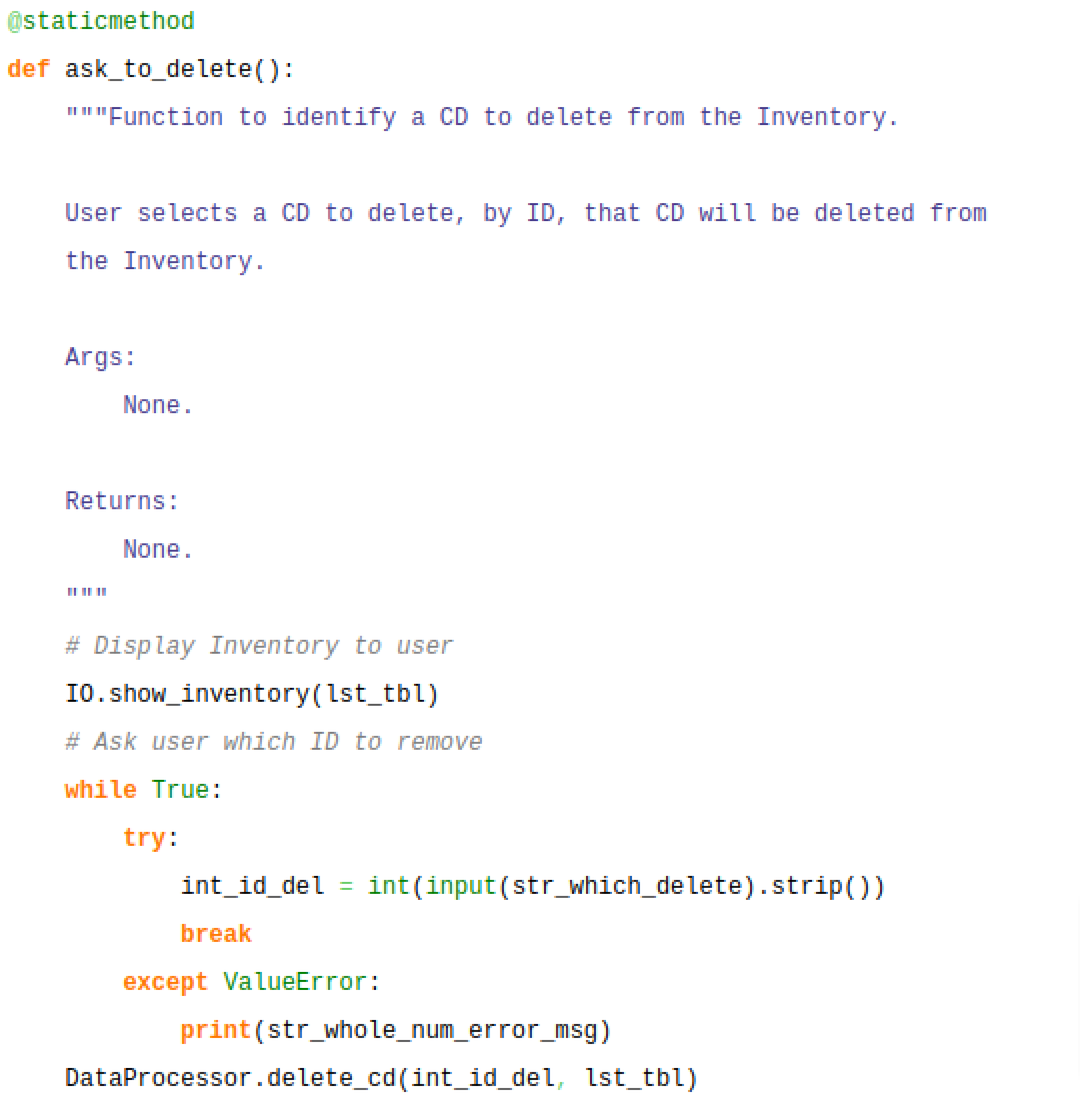


```
class FileProcessor:
    """Processing the data to and from text file"""

    @staticmethod
    def read_file(str_file_name, obj_file, lst_tbl, dic_row):
        """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:
            str_file_name(string): File name from which the data will be read
            obj_file(defaults to None): file object
            lst_tbl(list): List of lists to hold data
            dic_row(dictionary): dictionary data row

        Returns:
            lst_tbl(list): List of lists to hold data
        """
        # Clears existing data
        lst_tbl.clear()

        # Loads data from file
        obj_file = open(str_file_name, 'rb')
        try:
            lst_tbl = pickle.load(obj_file)
        except EOFError:
            pass
        obj_file.close()
        return lst_tbl
```

Figure 1 - Arguments and returns

Here is an example of adding back in the arguments and the returns. This function now takes 4 arguments, and returns one. In the previous assignment, we could get away with relying on the global variables, due to performing changes in a serial fashion, without the chance for a conflict, like a race condition. But, after being instructed that it was not proper style, I added them all back into all of the functions where appropriate. In this function, it returns the "*lst_tbl*" after loading it from the file. Eight functions now take arguments now, but this was only one example

Error Handling

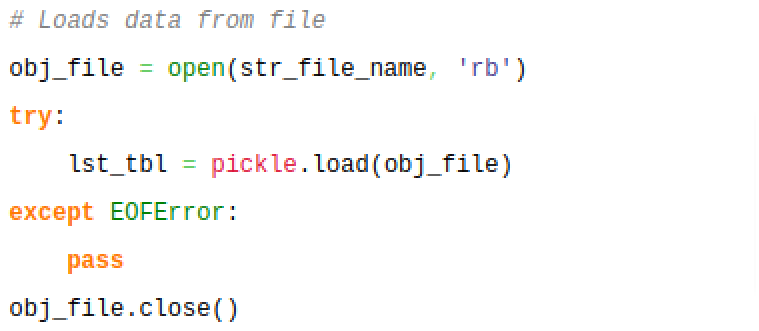


```
@staticmethod
def ask_to_delete():
    """Function to identify a CD to delete from the Inventory.

    User selects a CD to delete, by ID, that CD will be deleted from
    the Inventory.

    Args:
        None.

    Returns:
        None.
    """
    # Display Inventory to user
    IO.show_inventory(lst_tbl)
    # Ask user which ID to remove
    while True:
        try:
            int_id_del = int(input(str_which_delete).strip())
            break
        except ValueError:
            print(str_whole_num_error_msg)
    DataProcessor.delete_cd(int_id_del, lst_tbl)
```

Figure 2 - Error Handling

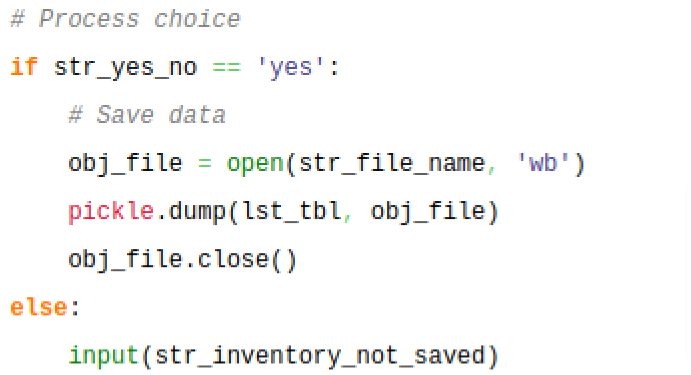
This was one of four examples of Error handling in this program. Another is the screenshot above. IN this version of the program, it handled ValueError twice, EOFError, and FileNotFoundError. I had used the "*try:except*" to catch Errors in previous assignments, here and there, so this was not new. Your suggestion in class to just invoke the error, and then you will know the type, was spot on. It’s always been the easiest way.

Pickling



```
# Loads data from file
obj_file = open(str_file_name, 'rb')
try:
    lst_tbl = pickle.load(obj_file)
except EOFError:
    pass
obj_file.close()
```

Figure 3 - Loading pickle file data



```
# Process choice
if str_yes_no == 'yes':
    # Save data
    obj_file = open(str_file_name, 'wb')
    pickle.dump(lst_tbl, obj_file)
    obj_file.close()
else:
    input(str_inventory_not_saved)
```

Figure 4 - Saving to a file with pickle

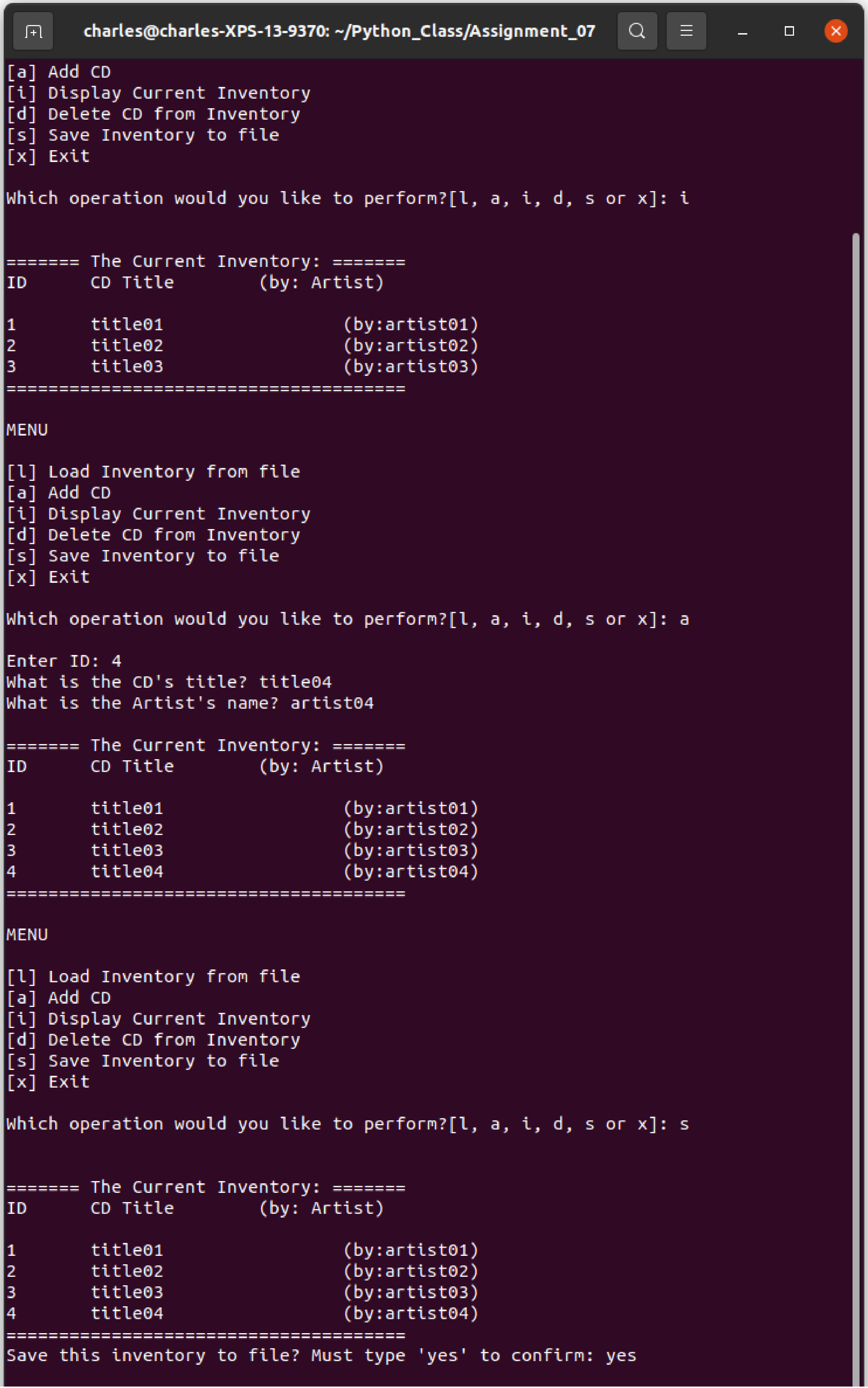
I was able to use Pickle twice in this program. I started out by using Shelve, but found pickle to be a more straightforward method instead. After a few variations and attempts to write and read data correctly, I determined that simple pickle statements were best. One issue I had was that since the files were binary, and not text, I had to add a bunch of print statements to see what was happening where, and in what order. In the previous versions, I could open and inspect the text files, and use Python Debugger. This week was a bit trickier as a result.Loading

Summary

In this assignment we were to use the program we created in Assignment 06, and modify it within certain parameters. Firstly, I added back in the Arguments and returns, so as to not rely on global variables. Secondly, I added Error handling for Input and File processing. Finally, I used Pickling to write the CD data to a binary file, and then load it from the file as well.

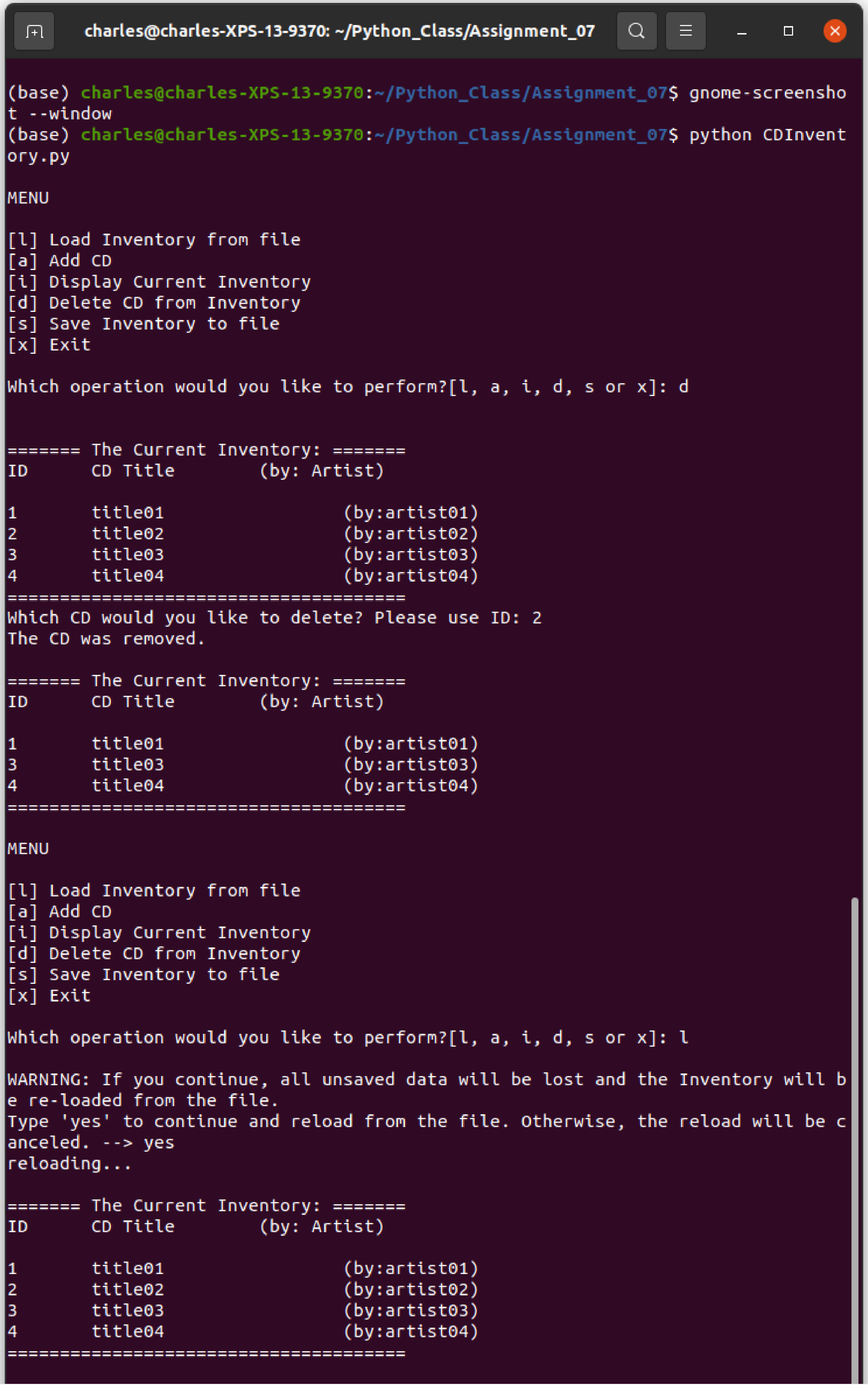
Screenshots

Terminal



```
[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       title01                 (by:artist01)
2       title02                 (by:artist02)
3       title03                 (by:artist03)
======================================

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: 4
What is the CD's title? title04
What is the Artist's name? artist04

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

1       title01                 (by:artist01)
2       title02                 (by:artist02)
3       title03                 (by:artist03)
4       title04                 (by:artist04)
======================================

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       title01                 (by:artist01)
2       title02                 (by:artist02)
3       title03                 (by:artist03)
4       title04                 (by:artist04)
======================================
Save this inventory to file? Must type 'yes' to confirm: yes
```

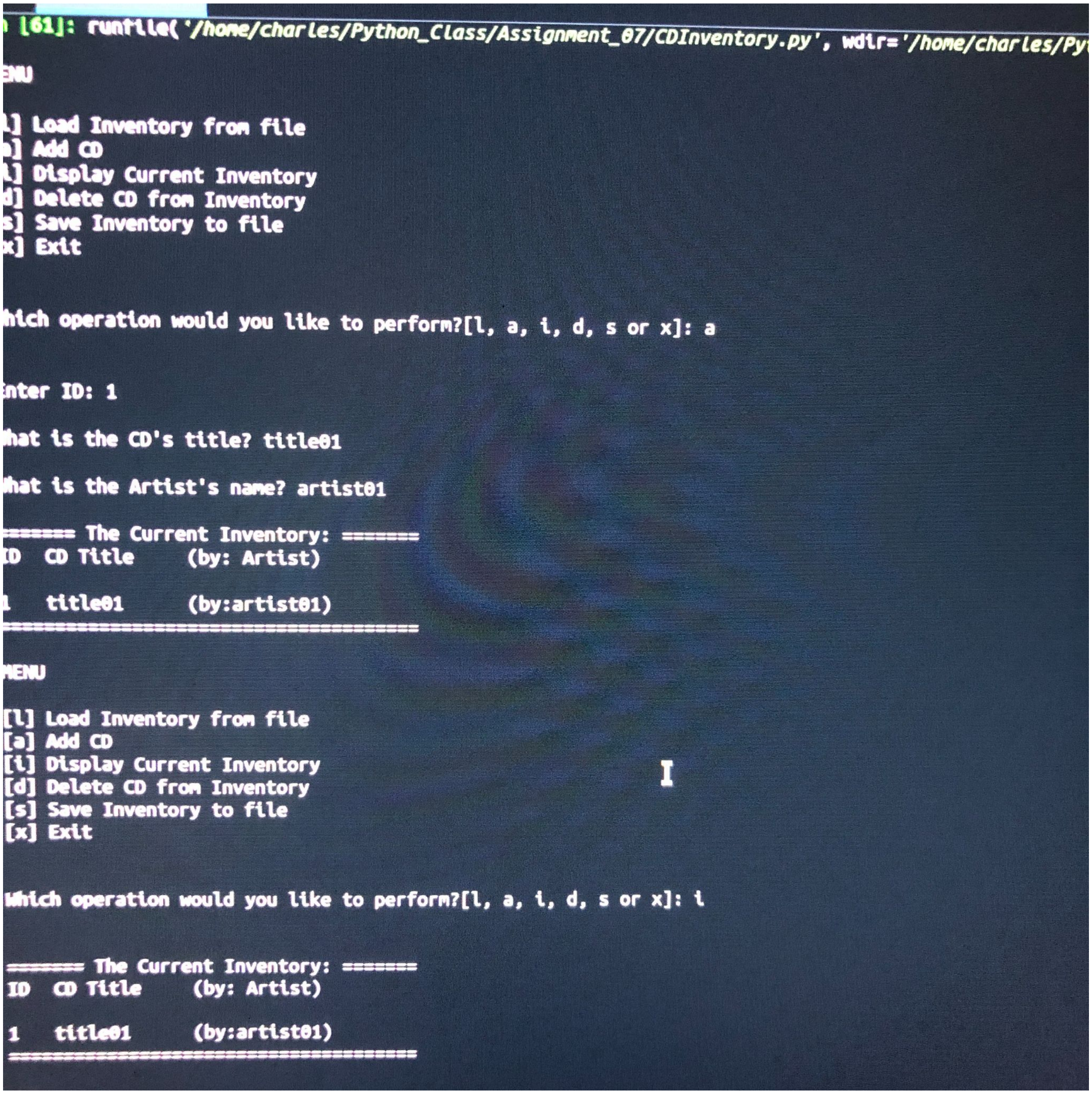
Figure 5 - Running in the Terminal



```
(base) charles@charles-XPS-13-9370:~/Python_Class/Assignment_07$ gnome-screensho
t --window
(base) charles@charles-XPS-13-9370:~/Python_Class/Assignment_07$ python CDInvent
ory.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]: d


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

1       title01                 (by:artist01)
2       title02                 (by:artist02)
3       title03                 (by:artist03)
4       title04                 (by:artist04)
======================================
Which CD would you like to delete? Please use ID: 2
The CD was removed.

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

1       title01                 (by:artist01)
3       title03                 (by:artist03)
4       title04                 (by:artist04)
======================================

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 will b
e re-loaded from the file.
Type 'yes' to continue and reload from the file. Otherwise, the reload will be c
anceled. --> yes
reloading...

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

1       title01                 (by:artist01)
2       title02                 (by:artist02)
3       title03                 (by:artist03)
4       title04                 (by:artist04)
======================================
```

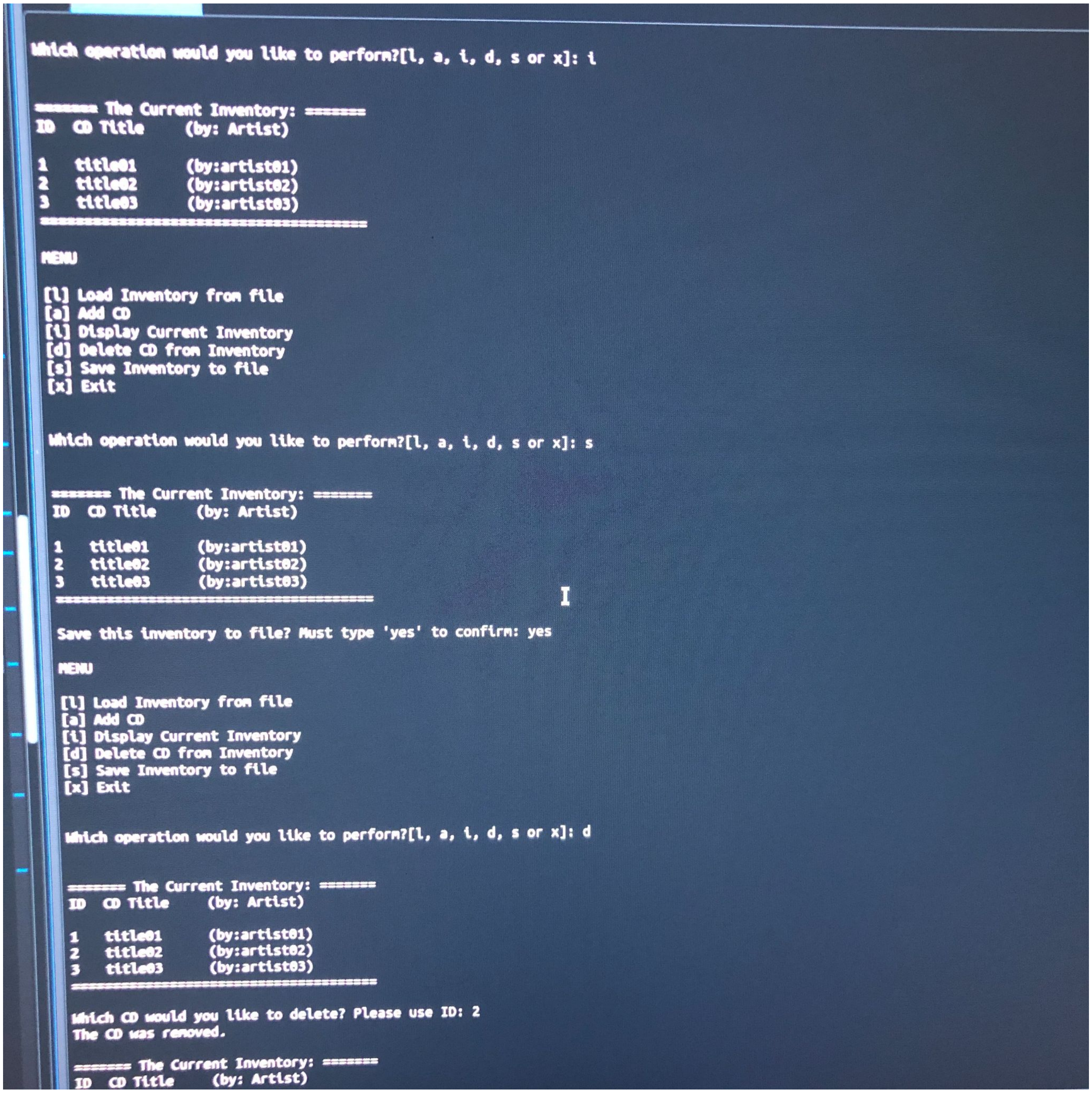
Figure 6 - Running in the Terminal

Spyder



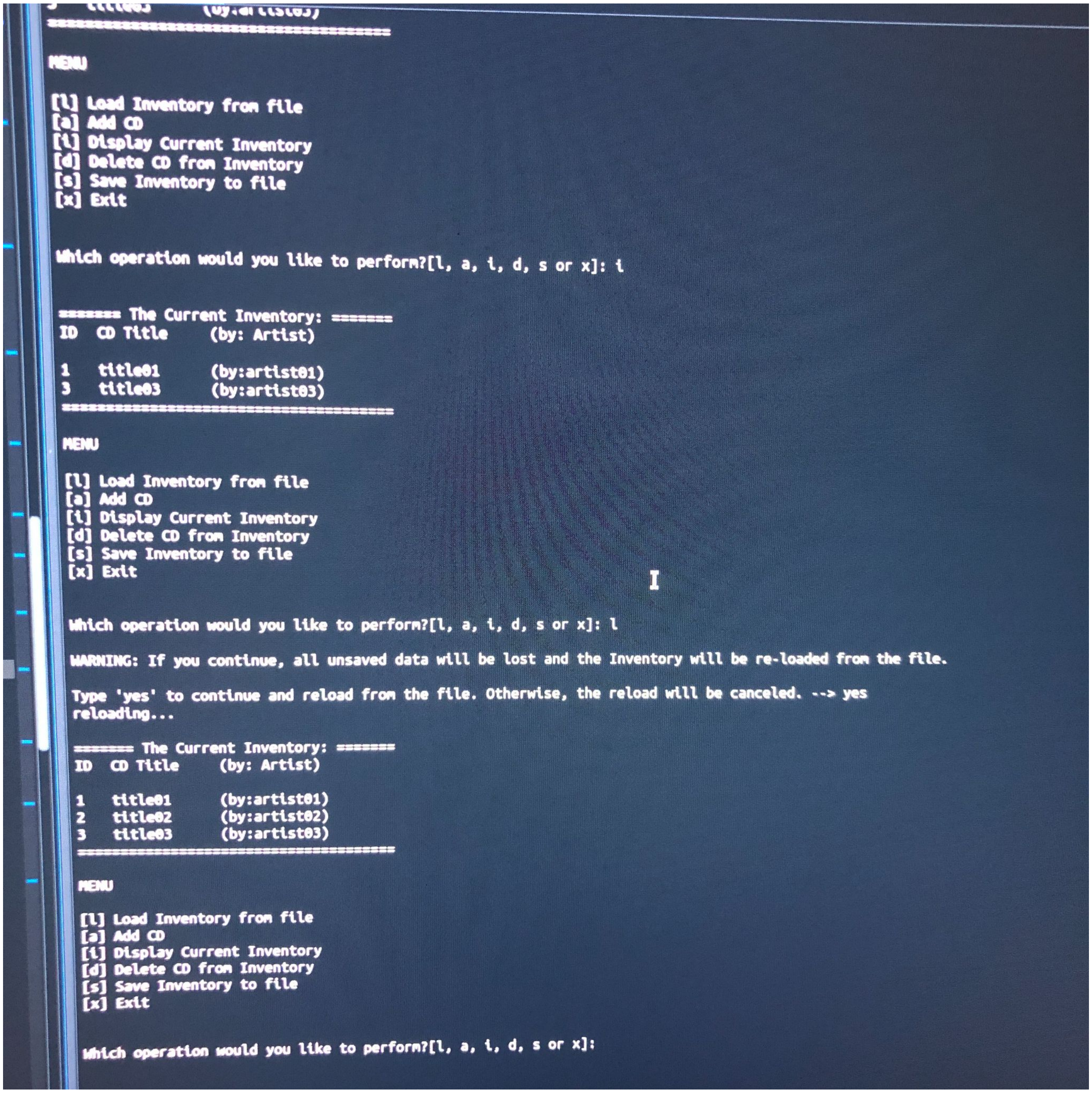
```
n [61]: runfile('/home/charles/Python_Class/Assignment_07/CDInventory.py', wdir='/home/charles/Py

ENU

l] Load Inventory from file
a] Add CD
i] Display Current Inventory
d] Delete CD from Inventory
s] Save Inventory to file
x] Exit


hich operation would you like to perform?[l, a, i, d, s or x]: a


nter ID: 1

hat is the CD's title? title01

hat is the Artist's name? artist01

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

1   title01      (by:artist01)
======================================

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   title01      (by:artist01)
======================================
```

Figure 7 - Running in Spyder



```
Which operation would you like to perform?[l, a, i, d, s or x]: i

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

1   title01      (by:artist01)
2   title02      (by:artist02)
3   title03      (by:artist03)
======================================

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   title01      (by:artist01)
2   title02      (by:artist02)
3   title03      (by:artist03)
======================================

Save this inventory to file? Must type 'yes' to confirm: yes

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   title01      (by:artist01)
2   title02      (by:artist02)
3   title03      (by:artist03)
======================================

Which CD would you like to delete? Please use ID: 2
The CD was removed.

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

Figure 8 - Running in Spyder



```
3   title03      (by:artist03)
======================================

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   title01      (by:artist01)
3   title03      (by:artist03)
======================================

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 will be re-loaded from the file.

Type 'yes' to continue and reload from the file. Otherwise, the reload will be canceled. --> yes
reloading...

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

1   title01      (by:artist01)
2   title02      (by:artist02)
3   title03      (by:artist03)
======================================

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]:
```

Figure 9 - Running in Spyder