Greg Eisele

Nov 19, 2022

IT FDN 110 B

Assignment 06

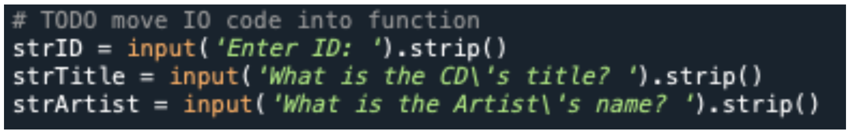
Introduction

Week 6 introduced functions and classes. Functions are a way to pre-package code that can be called at a later point in the program. Its useful to make code more efficient when a program may need to utilize the same block of code several times.

Assignment 06

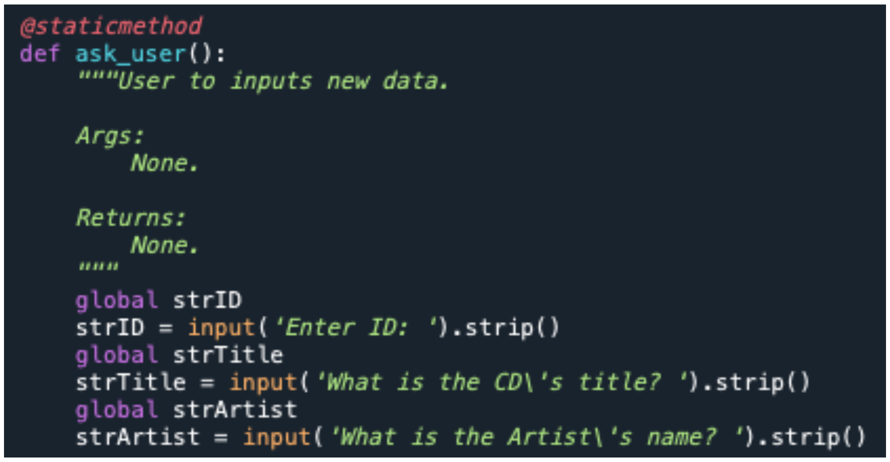
For assignment 6, we were provided with a functioning CD Inventory python file, and asked to restructure the code, relocating specific code blocks into functions, and calling the functions where applicable. There are four instances where we must accomplish this, and i'll walk through each instance.

The first instance was the code block (figure 1) that intakes a new entry row (ID, Title, Artist). This code was cut and pasted into a new function, 'ask_user()', which is entered in the IO class. See figure 2 for 'ask_user' function.



```
# TODO move IO code into function
strID = input('Enter ID: ').strip()
strTitle = input('What is the CD\'s title? ').strip()
strArtist = input('What is the Artist\'s name? ').strip()
```

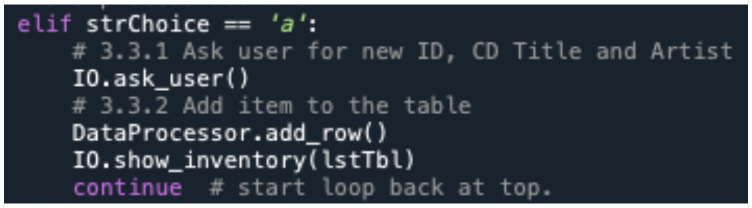
Figure 1 - New User Entry Code Block



```
@staticmethod
def ask_user():
    """User to inputs new data.

    Args:
        None.

    Returns:
        None.
    """
    global strID
    strID = input('Enter ID: ').strip()
    global strTitle
    strTitle = input('What is the CD\'s title? ').strip()
    global strArtist
    strArtist = input('What is the Artist\'s name? ').strip()
```

Figure 2 - ask_user() Function Definition

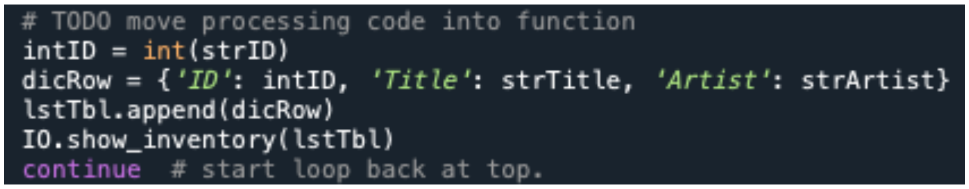
The 'global' prefix had to be used on each of the variables in order for the variable to be recognized outside the function. In the 'a' (add) block of the main loop, the function 'IO.ask_user()' gets called to accomplish the code that was previously there (figure 3).



```
elif strChoice == 'a':
    # 3.3.1 Ask user for new ID, CD Title and Artist
    IO.ask_user()
    # 3.3.2 Add item to the table
    DataProcessor.add_row()
    IO.show_inventory(lstTbl)
    continue  # start loop back at top.
```

Figure 3 - ask_user() Function Call

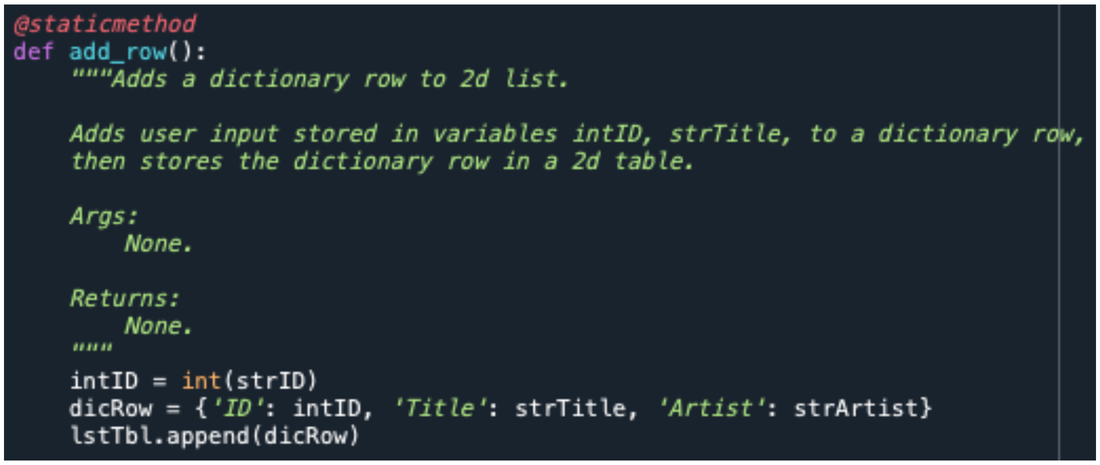
The second instance was the code that appends the list of albums (see figure 4).



```python
# TODO move processing code into function
intID = int(strID)
dicRow = {'ID': intID, 'Title': strTitle, 'Artist': strArtist}
lstTbl.append(dicRow)
IO.show_inventory(lstTbl)
continue  # start loop back at top.
```

Figure 4 - Append List Code Block

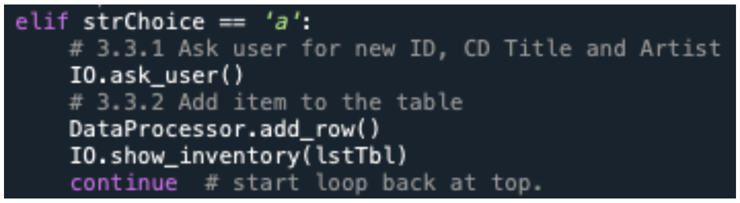
This code was cut and pasted into a new function, 'add_row()', inside the 'DataProcessor' class. See figure 5 for function definition.



```python
@staticmethod
def add_row():
    """Adds a dictionary row to 2d list.

    Adds user input stored in variables intID, strTitle, to a dictionary row,
    then stores the dictionary row in a 2d table.

    Args:
        None.

    Returns:
        None.
    """
    intID = int(strID)
    dicRow = {'ID': intID, 'Title': strTitle, 'Artist': strArtist}
    lstTbl.append(dicRow)
```

Figure 5 - add_row() Function Definition

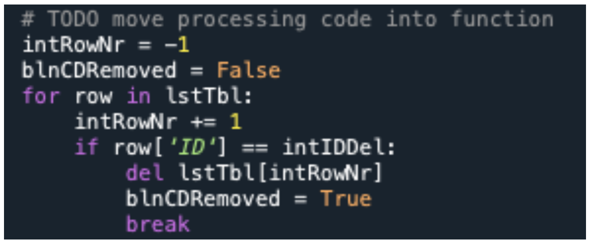
In the 'a' (add) block of the main loop, the function 'DataProcessor.add_row()' gets called to accomplish the code that was previously there (figure 6).



```python
elif strChoice == 'a':
    # 3.3.1 Ask user for new ID, CD Title and Artist
    IO.ask_user()
    # 3.3.2 Add item to the table
    DataProcessor.add_row()
    IO.show_inventory(lstTbl)
    continue  # start loop back at top.
```

Figure 6 - add_row() Function Call

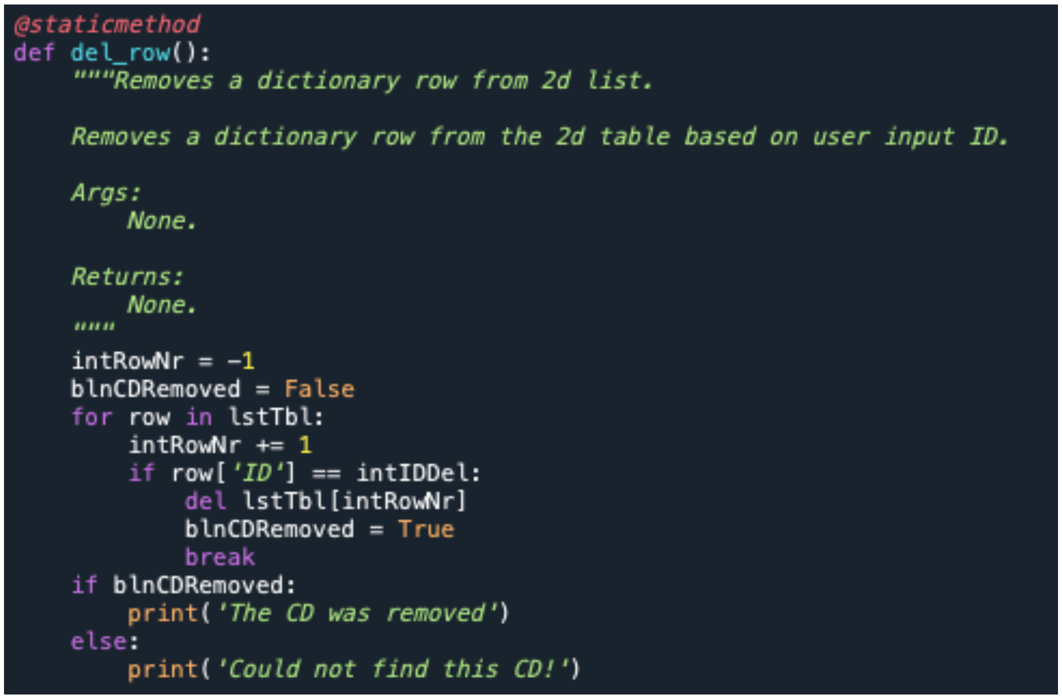
The third instance was the code that deletes a row from the list of albums (see figure 7).



```
# TODO move processing code into function
intRowNr = -1
blnCDRemoved = False
for row in lstTbl:
    intRowNr += 1
    if row['ID'] == intIDDel:
        del lstTbl[intRowNr]
        blnCDRemoved = True
        break
```

Figure 7 - Delete Row Code Block

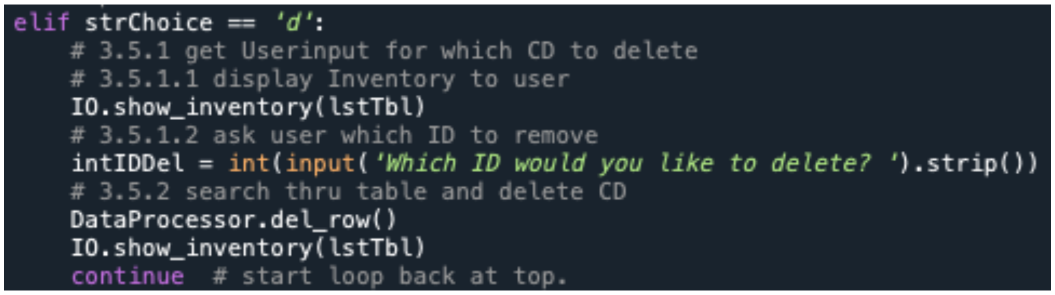
This code was cut and pasted into a new function ‘del_row()’, inside the ‘DataProcessor’ class. See figure 8 for function definition.



```
@staticmethod
def del_row():
    """Removes a dictionary row from 2d list.

    Removes a dictionary row from the 2d table based on user input ID.

    Args:
        None.

    Returns:
        None.
    """
    intRowNr = -1
    blnCDRemoved = False
    for row in lstTbl:
        intRowNr += 1
        if row['ID'] == intIDDel:
            del lstTbl[intRowNr]
            blnCDRemoved = True
            break
    if blnCDRemoved:
        print('The CD was removed')
    else:
        print('Could not find this CD!')
```

Figure 8 - del_row() Function Definition

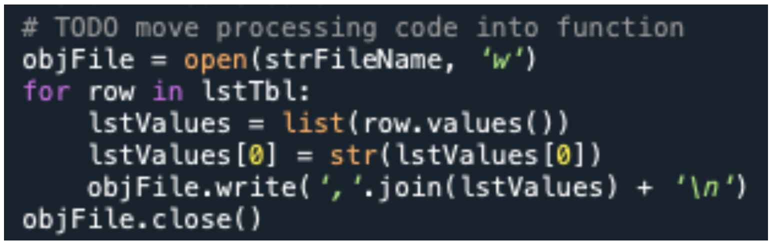
In the delete (‘d’) block of code, the function ‘DataProcessor.del_row()’ gets called to accomplish the code that was previously there (figure 9).



```
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.del_row()
    IO.show_inventory(lstTbl)
    continue  # start loop back at top.
```

Figure 9 - del_row() Function Call

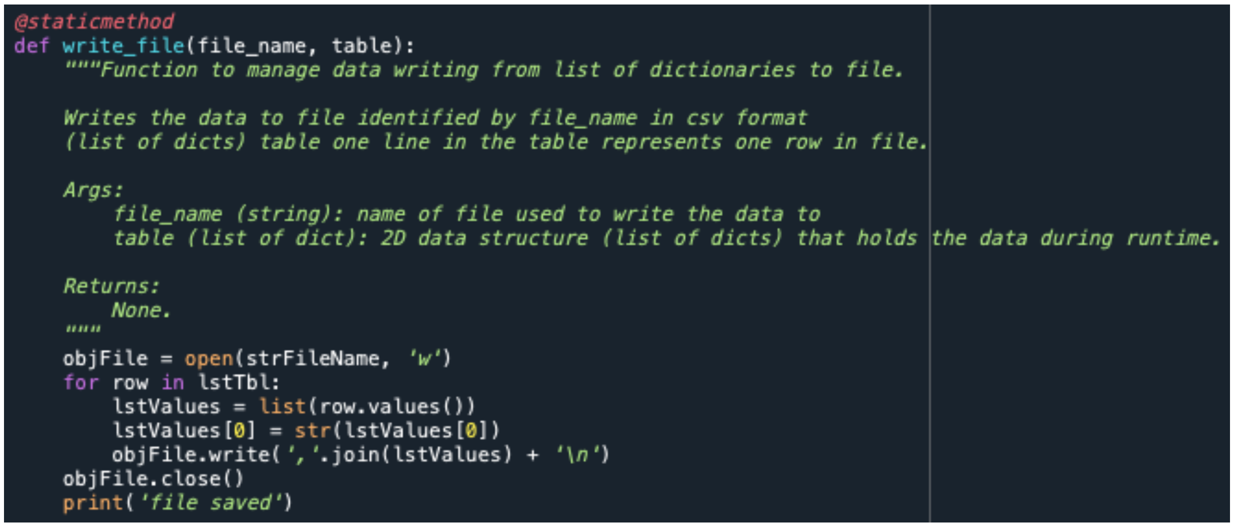
The fourth (and final) instance was the code that writes the list from memory to file (figure 10).



```
# TODO move processing code into function
objFile = open(strFileName, 'w')
for row in lstTbl:
    lstValues = list(row.values())
    lstValues[0] = str(lstValues[0])
    objFile.write(','.join(lstValues) + '\n')
objFile.close()
```

Figure 10 - Write to File Code Block

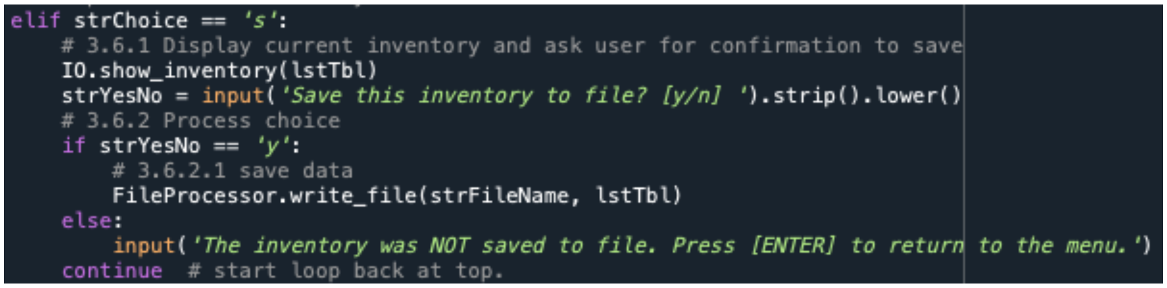
This code was cut and pasted into a new function, 'write_file()' (figure 11). Two parameters had to be used (file_name and table) in order to get the program to function correctly.



```
@staticmethod
def write_file(file_name, table):
    """Function to manage data writing from list of dictionaries to file.

    Writes the data to file identified by file_name in csv format
    (list of dicts) table one line in the table represents one row in file.

    Args:
        file_name (string): name of file used to write the data to
        table (list of dict): 2D data structure (list of dicts) that holds the data during runtime.

    Returns:
        None.
    """
    objFile = open(strFileName, 'w')
    for row in lstTbl:
        lstValues = list(row.values())
        lstValues[0] = str(lstValues[0])
        objFile.write(','.join(lstValues) + '\n')
    objFile.close()
    print('file saved')
```

Figure 11 - write_file() Function Definition.

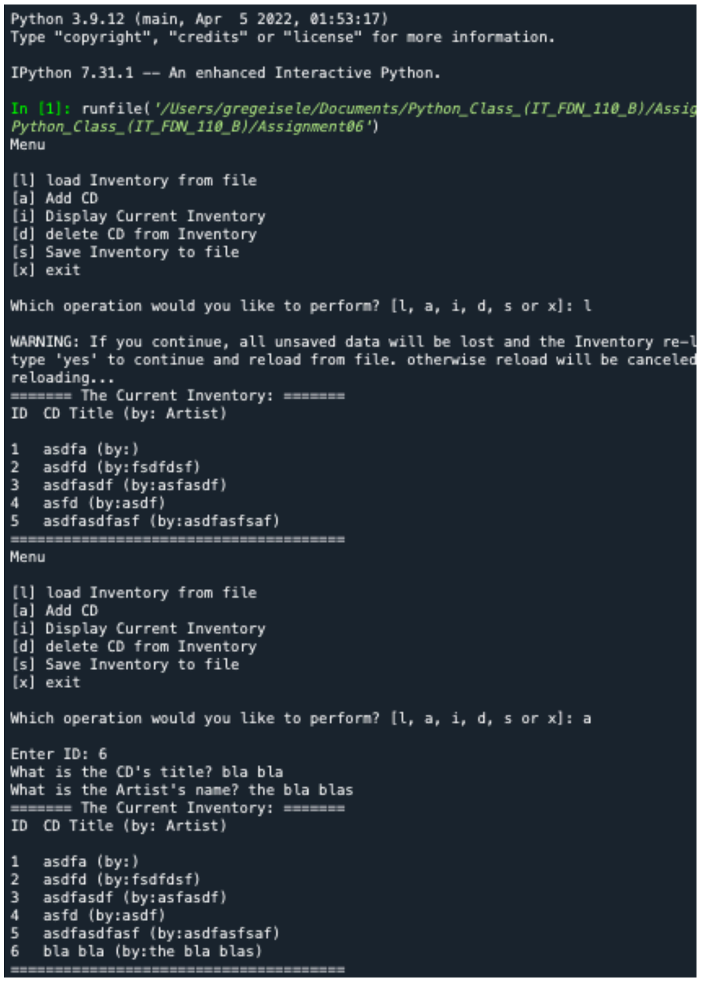
In the save ('s') block of code, the function FileProcessor.write_file() was called, using 'strFileName, and 'lstTbl' variables. See figure 12 for code execution.



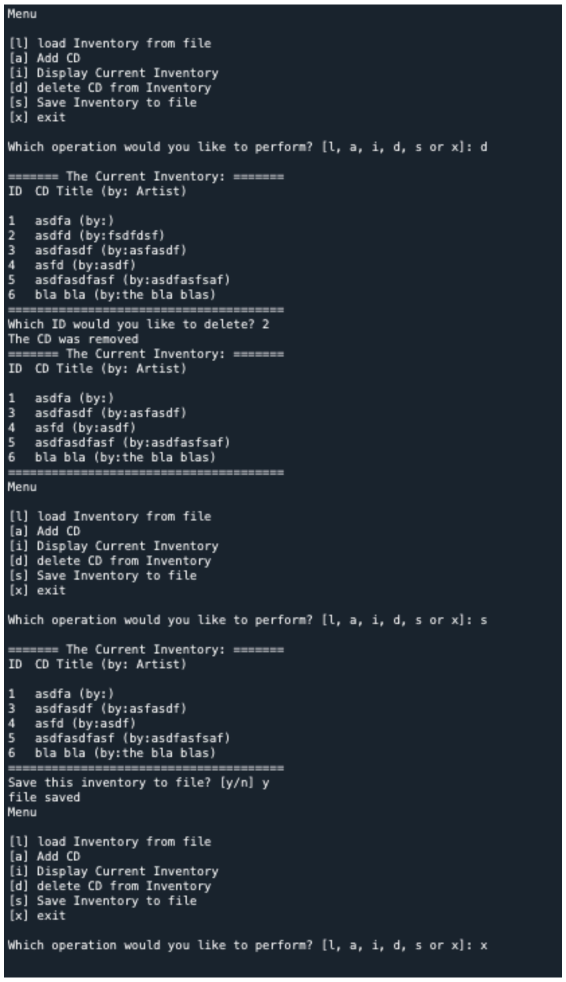
```
elif strChoice == 's':
    # 3.6.1 Display current inventory and ask user for confirmation to save
    IO.show_inventory(lstTbl)
    strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
    # 3.6.2 Process choice
    if strYesNo == 'y':
        # 3.6.2.1 save data
        FileProcessor.write_file(strFileName, lstTbl)
    else:
        input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
    continue  # start loop back at top.
```

Figure 12 - write_file() Function Call

Figure 13 show different option selections functioning correctly in Spyder.



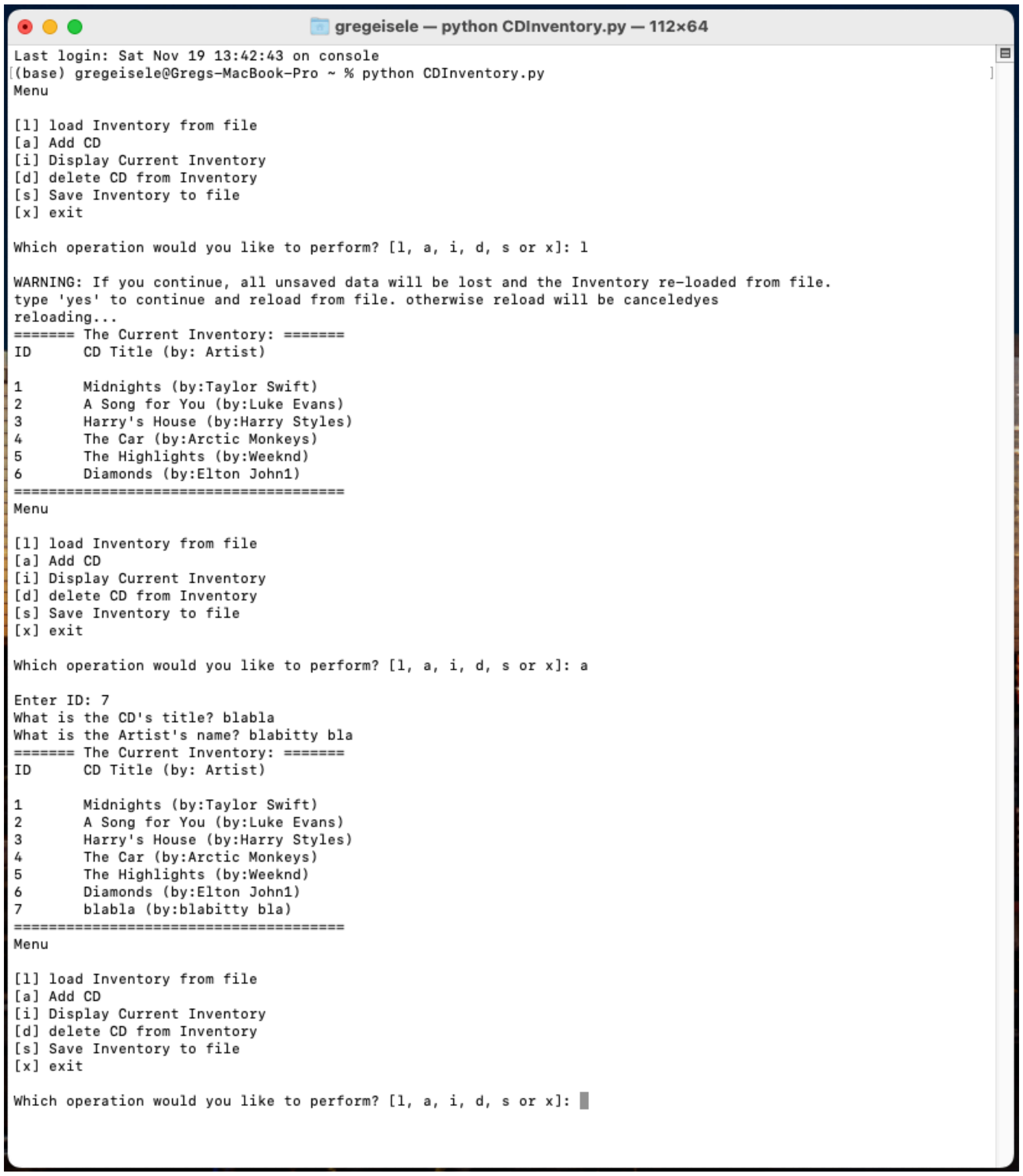
```
Python 3.9.12 (main, Apr  5 2022, 01:53:17)
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('/Users/gregeisele/Documents/Python_Class_(IT_FDN_110_B)/Assig
Python_Class_(IT_FDN_110_B)/Assignment06')
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-l
type 'yes' to continue and reload from file. otherwise reload will be canceled
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   asdfa (by:)
2   asdfd (by:fsdfdsf)
3   asdfasdf (by:asfasdf)
4   asfd (by:asdf)
5   asdfasdfasf (by:asdfasfsaf)
======================================
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: 6
What is the CD's title? bla bla
What is the Artist's name? the bla blas
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   asdfa (by:)
2   asdfd (by:fsdfdsf)
3   asdfasdf (by:asfasdf)
4   asfd (by:asdf)
5   asdfasdfasf (by:asdfasfsaf)
6   bla bla (by:the bla blas)
======================================
```



```
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   asdfa (by:)
2   asdfd (by:fsdfdsf)
3   asdfasdf (by:asfasdf)
4   asfd (by:asdf)
5   asdfasdfasf (by:asdfasfsaf)
6   bla bla (by:the bla blas)
======================================
Which ID would you like to delete? 2
The CD was removed
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   asdfa (by:)
3   asdfasdf (by:asfasdf)
4   asfd (by:asdf)
5   asdfasdfasf (by:asdfasfsaf)
6   bla bla (by:the bla blas)
======================================
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   asdfa (by:)
3   asdfasdf (by:asfasdf)
4   asfd (by:asdf)
5   asdfasdfasf (by:asdfasfsaf)
6   bla bla (by:the bla blas)
======================================
Save this inventory to file? [y/n] y
file saved
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
```

Figure 13 - Program Working in Spyder

Figure 14 shows the program functioning in Terminal.



gregeisele — python CDInventory.py — 112×64

```
Last login: Sat Nov 19 13:42:43 on console
(base) gregeisele@Gregs-MacBook-Pro ~ % python CDInventory.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]: 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 canceledyes
reloading...
======= The Current Inventory: =======
ID	CD Title (by: Artist)

1	Midnights (by:Taylor Swift)
2	A Song for You (by:Luke Evans)
3	Harry's House (by:Harry Styles)
4	The Car (by:Arctic Monkeys)
5	The Highlights (by:Weeknd)
6	Diamonds (by:Elton John1)
======================================
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: 7
What is the CD's title? blabla
What is the Artist's name? blabitty bla
======= The Current Inventory: =======
ID	CD Title (by: Artist)

1	Midnights (by:Taylor Swift)
2	A Song for You (by:Luke Evans)
3	Harry's House (by:Harry Styles)
4	The Car (by:Arctic Monkeys)
5	The Highlights (by:Weeknd)
6	Diamonds (by:Elton John1)
7	blabla (by:blabitty bla)
======================================
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 14 - Program Working in Terminal

Summary

Of the 4 functions we needed to create, the 3 processing functions were pretty straightforward for me. However, the 1 IO function provided the greatest difficulty. For several different attempts at creating the IO function ‘ask_user()’, I would just get some sort of new error. The errors always revolved around the variable not being defined. When I would define it outside the function, in the data section, the errors would go away, but the code wouldn’t work properly. The user input would never overwrite the variable data, because one was only defined in the function and the other was defined globally - so they wouldn’t talk. When I learned about the ‘global’ pretext, it fixed everything. I’m not sure if this was the correct way to solve the code for this assignment, but it seems to have worked….

Git Hub Link

https://github.com/Greg6874/Assignment_06