Jeffrey Shen

3/1/2020

Foundations of Programming (Python)

Assignment 06

Functions and Classes with Python

Introduction

In this document, I will provide an overview of using functions and classes. This report will include discussion of the CDInventory.py script and what challenges I came across.

CD Inventory Script

From previous week's assignment, a python script was created that manages a CD inventory based on user input. This week focused on implementing functions and classes. The initial pseudocode was provided by the instructor. The menu and structure of the code is as follows:

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

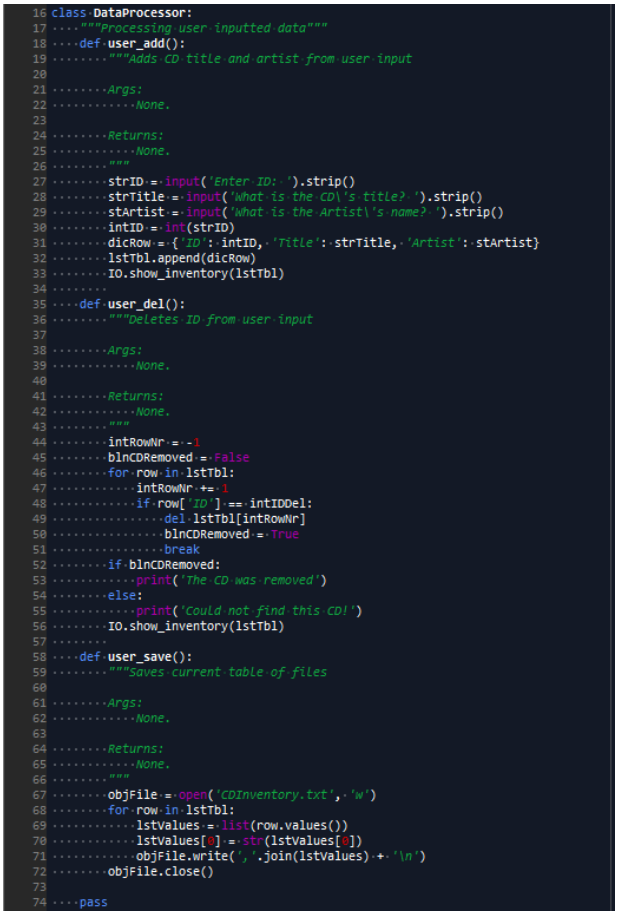
```
if strChoice == 'x'
    do this
elif strChoice == 'l'
    do this
elif strChoice == 'a'
    do this
elif strChoice == 'i'
    do this
elif strChoice == 'd'
    do this
elif strChoice == 's'
    do this
```

In the beginning of the script, I initialize the variables needed. These are the table/dictionaries that data will be stored in.

```
'''
Title: Assignment06.py
Desc: Working with classes and functions.
DBiesinger, 2020-Jan-01, Created File
Jeffrey Shen, 2020-Feb-25, Created file and comments
'''

'''DATA'''
strChoice = ''# User input
lstTbl = []# list of lists to hold data
dicRow = {}# list of data row
strFileName = 'load.txt'# data storage file
objFile = None# file object
```

Figure 1 Initialize Variables

The first class is called "DataProcessor" – this holds a grouping of user defined functions. There is user adding, user deleting, and user saving to manage their CD inventory.



```
class DataProcessor:
    """Processing user inputted data"""
    def user_add():
        """Adds CD title and artist from user input

        Args:
            None.

        Returns:
            None.
        """
        strID = input('Enter ID: ').strip()
        strTitle = input('What is the CD\'s title? ').strip()
        stArtist = input('What is the Artist\'s name? ').strip()
        intID = int(strID)
        dicRow = {'ID': intID, 'Title': strTitle, 'Artist': stArtist}
        lstTbl.append(dicRow)
        IO.show_inventory(lstTbl)

    def user_del():
        """Deletes ID from user input

        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!')
        IO.show_inventory(lstTbl)

    def user_save():
        """Saves current table of files

        Args:
            None.

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

    pass
```

Figure 2 DataProcessor Class

Each of these functions in the class are called objects. They are function called in the main while loop. The functions have the same syntax as from previous week's assignment, however, the key difference is that they are organized into a class structure.

The next class is "FileProcessor" – this holds two functions: read from text file and create a list of dictionary, write to text file.

```python
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('Load.txt', '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(strFileName, table):
        """Writes file data to text file

        Args:
            file name and table data

        Returns:
            None.
        """
        with open('CDInventory.txt', 'a+') as f:
            for row in lstTbl:
                txt_line = ", ".join([str(value) for value in row.values()]) + '\n'
                f.write(txt_line)
        f.close()

        pass
```

Figure 3 FileProcessor Class

The last class included is “IO” for input and output. There are three functions: print the menu options to user, determine menu input choice, show CD inventory from table. The class is used for presentation and is associated with print commands that are shown to the user for input/output.

```python
'''PRESENTATION (Input/Output)'''
class IO:

    @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\tCD Title (by: Artist)\n')
        for row in table:
            print('{}\t{} (by:{})'.format(*row.values()))
        print('======================================')

    pass
```

Figure 4 IO Class

In the main while loop, the structure was outlined above but follows an if-elif-elif-elif-elif-elif-else: structure. However, the key aspect in this aspect was to function call from the while loop. This way, the code structure is organized into presentation, processing, and data. For example, in line 199 and 200, if the user selects “a” to add a CD, DataProcessor and IO classes are called. These are then followed by the function associated in that class. The rest of the code is formatted as such so that functions are called respectively.

```
# 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
    elif 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
        # 3.3.2 Add item to the table
        DataProcessor.user_add()
        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 User input 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.user_del()
        # show updated table
        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
            DataProcessor.user_save()
        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')
```

Figure 5 Main While Loop

In the appendix, an example scenario of the script is shown, depicting each of the menu operations (both in Spyder and in the Terminal).

Questions

• What is a function?

- A function is a way of grouping statements that execute particular set of code via defined name.

• What are parameters?

- Allow values to be passed for processing, generally called arguments.

• What are arguments?

- Generally, arguments are values provided to a function. For example, “def Function(argument)”

• What is the difference between parameters and arguments?

- Parameters are variables in a method definition whereas arguments are the actual value passed into the function.

• What are return values?

- They are what variables are outputted by a function (“returns” the results)

• What is the difference between a global and a local variable?

- Global variables are reached anywhere in the code whereas local is only within that scope.

• What is shadowing?

- Variable declared within a certain scope (method, class) has the same name as a variable declared in an outer scope.

• How do you use functions to organize your code?

- Functions help clearly separate tasks, so it is more readable and modifiable. In this assignment, functions were defined by user input actions.

• What is the difference between a function and a class?

- Class is a definition of an object which contains variables (attributes). A function is a method in an object and is a prescribed set of code that can be called.

• How do functions help you program using the “Separations of Concerns" pattern?

- Code is separated into data, processing, and presentation. There can be classes for each and functions within so that when calling in the main code, it’s easier to follow.

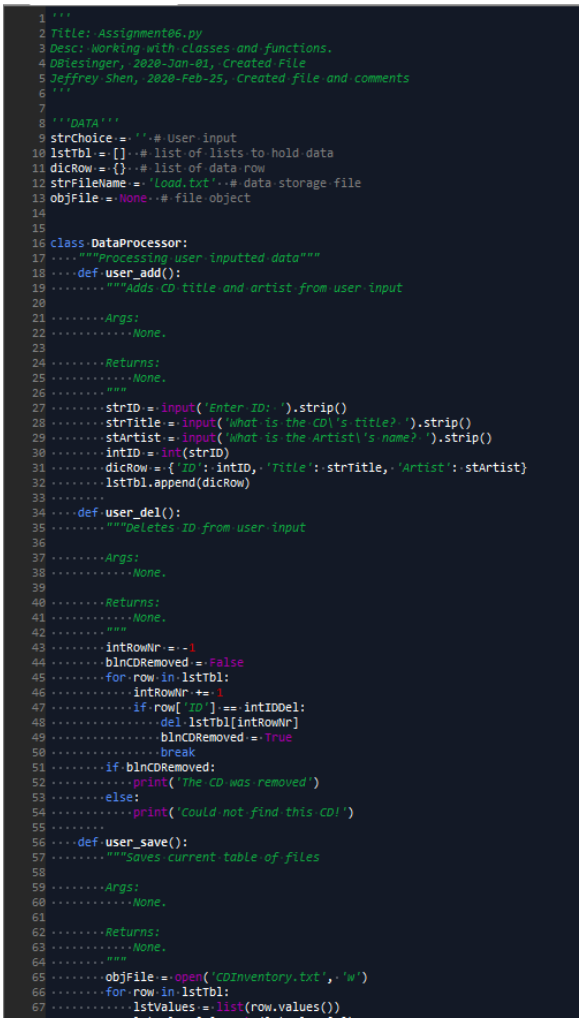
Summary

In this lab, I explored using classes and functions in Python to manage a CD inventory. The idea of using functions and classes stems from Object Oriented Programming[1] and organizing your code into a Separations of Concerns (SOC). This way the code is more readable to viewers and yourself. The challenge in this assignment was learning how to accurately function call from the while loop. The syntax and formatting was confusing at first since there were so multiple functions and classes. I could see why SOC is used for more complex programs.[2]

[1] https://www.w3schools.com/python/python_classes.asp
[2] https://docs.python.org/3/tutorial/classes.html

Appendix

Complete Code for AddressBook.py



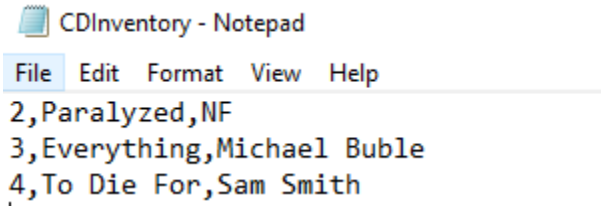
```
'''
Title: Assignment06.py
Desc: Working with classes and functions.
DBiesinger, 2020-Jan-01, Created File
Jeffrey Shen, 2020-Feb-25, Created file and comments
'''

'''DATA'''
strChoice = ''  # User input
lstTbl = []  # list of lists to hold data
dicRow = {}  # list of data row
strFileName = 'Load.txt'  # data storage file
objFile = None  # file object


class DataProcessor:
    """Processing user inputted data"""
    def user_add():
        """Adds CD title and artist from user input

        Args:
            None.

        Returns:
            None.
        """
        strID = input('Enter ID: ').strip()
        strTitle = input('What is the CD\'s title? ').strip()
        strArtist = input('What is the Artist\'s name? ').strip()
        intID = int(strID)
        dicRow = {'ID': intID, 'Title': strTitle, 'Artist': strArtist}
        lstTbl.append(dicRow)

    def user_del():
        """Deletes ID from user input

        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!')

    def user_save():
        """Saves current table of files

        Args:
            None.

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

    pass


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('Load.txt', '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(strFileName, table):
        """Writes file data to text file

        Args:
            file name and table data

        Returns:
            None.
        """
        with open('CDInventory.txt', 'a+') as f:
            for row in lstTbl:
                txt_line = '', ''.join([str(value) for value in row.values()]) + '\n'
                f.write(txt_line)
        f.close()

        pass


'''PRESENTATION (Input/Output)'''
class IO:

    @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\tCD Title (by: Artist)\n')
        for row in table:
            print('{}\t{} (by:{})'.format(*row.values()))
        print('====================================')

        pass


# Main While loop
# 1. When program starts, read in the currently saved Inventory
FileProcessor.read_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
    elif 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
        # 3.3.2 Add item to the table
        DataProcessor.user_add()
        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 user input 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.user_del()
        # show updated table
        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
            DataProcessor.user_save()
        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')
```

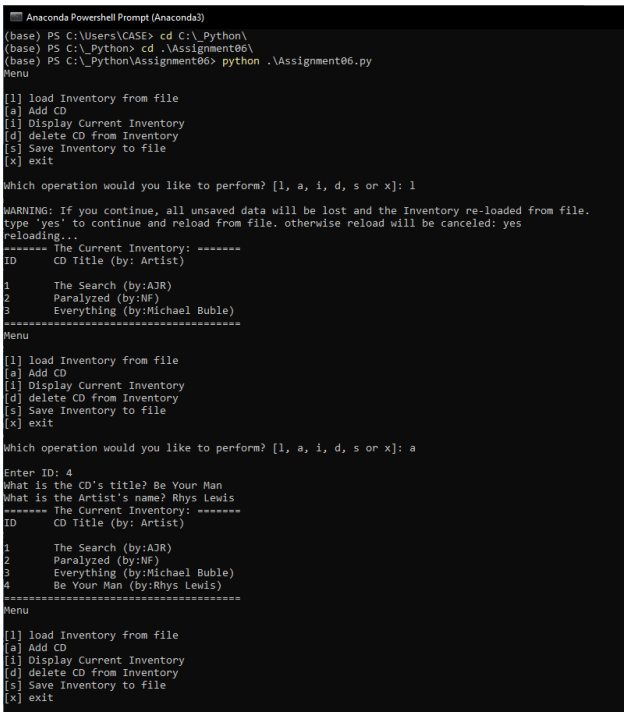
Example Run from Spyder

```
In [3]: runfile('C:/_Python/Assignment06/Assignment06.py', wdir='C:/_Python/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-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       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
======================================
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? To Die For

What is the Artist's name? Sam Smith
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
4       To Die For (by:Sam Smith)
======================================
```

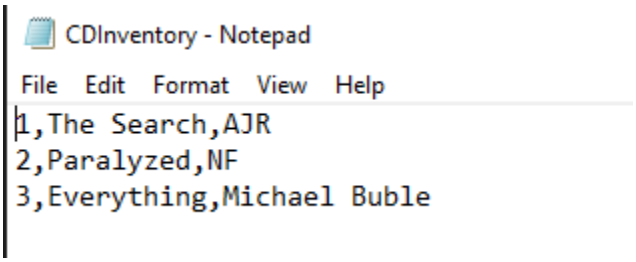
```
Menu

[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       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
4       To Die For (by:Sam Smith)
======================================

Which ID would you like to delete? 1
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
4       To Die For (by:Sam Smith)
======================================
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)

2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
4       To Die For (by:Sam Smith)
======================================

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



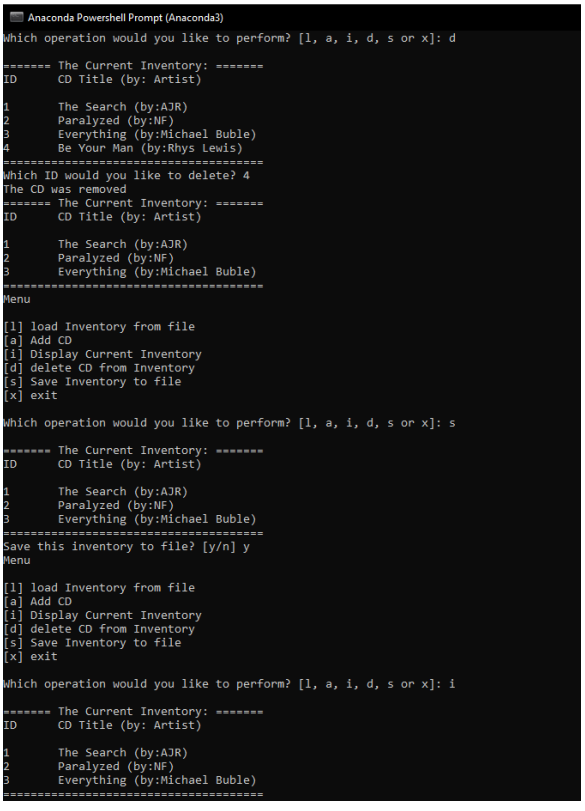
Example Run from Terminal



```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\CASE> cd C:\_Python\
(base) PS C:\_Python> cd .\Assignment06\
(base) PS C:\_Python\Assignment06> python .\Assignment06.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 canceled: yes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
======================================
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? Be Your Man
What is the Artist's name? Rhys Lewis
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
4       Be Your Man (by:Rhys Lewis)
======================================
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
```



```
CDInventory - Notepad
File  Edit  Format  View  Help
1,The Search,AJR
2,Paralyzed,NF
3,Everything,Michael Buble
```



```
Anaconda Powershell Prompt (Anaconda3)
Which operation would you like to perform? [l, a, i, d, s or x]: d

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

1       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
4       Be Your Man (by:Rhys Lewis)
======================================
Which ID would you like to delete? 4
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
======================================
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       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
======================================
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]: i

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

1       The Search (by:AJR)
2       Paralyzed (by:NF)
3       Everything (by:Michael Buble)
======================================
```