Ben Streck

04 Dec 2022

Foundations of Programming (Python)

Assignment 08

Object Oriented Programming

Introduction

In this assignment, I outline the process of creating a CD inventory script in python. This script differs from Assignment06 and Assignment07 because it uses object-oriented programming. The script presents users with a menu that gives them five options: 'Load Inventory from File', 'Add CD', 'Display Current Inventory', 'Save Inventory to File' and 'Exit.' Each menu option has different functionality and is accompanied by code that loads, appends, displays, or saves data to the cdInventory.txt file. Completing this assignment requires an understanding of object-oriented programming, exception handling, classes, functions, and more. In CD_Inventory.py, I organized my script into four main sections: Data, Processing, Presentation (Input/ Output), and Main Body. I used classes to organize my functions into logical groups such as data, file processing, and input/output. I also used classes as the blueprint for creating CD objects. Finally, I used a while loop with conditional statements to display the available options and run the correct functions/ methods based on user inputs.

Writing the Script

Creating the CD Object Blueprint

Listing 1 shows how I created the class for CD objects. First, I initialized the three fields (cd_id, cd_title, and cd_artist). Then, I defined the init constructor followed by the three attributes. Finally, I included all the important properties and methods to ensure that the CD objects behave correctly.

```
class CD:
    # Fields
    cd_id = 0
    cd_title = ''
    cd_artist = ''
    # Constructor
    def __init__(self, num, cd, art):
    # Attributes
        self.__cd_id = num
        self.__cd_title = cd
        self.__cd_artist = art
    # Properties
    @property
    def cd_id(self):
        return self.__cd_id
    @cd_id.setter
    def cd_id(self, value):
        if str(value).isnumeric():
            self.__cd_id = value
        else:
            raise Exception('The cd_id must be a positive integer')
    @property
    def cd_title(self):
        return self.__cd_title
    @cd_title.setter
    def cd_title(self, value):
        self.__cd_title = value
    @property
    def cd_artist(self):
        return self.__cd_artist
    @cd_artist.setter
    def cd_artist(self, value):
            self.__cd_artist = value
    # Methods
    def noAnswer(self):
        return """
# ------------------------------ #
This object has three properties:
  - cd_id
  - cd_title
  - cd_artist
# ------------------------------ #
                """
    def __str__(self):
        return self.noAnswer()
```

Listing 1 – CD Object Blueprint

Loading and Saving the Data

Listing 2 shows the class for saving and loading data to and from the text file. This code is similar to that of Assignment06, but it required some slight adjustments to accommodate the change from a list of dictionaries to a list of CD objects.

```
class FileIO:
    @staticmethod
    def save_inventory(file_name, table):
        with open(file_name, 'w') as file:
            for row in table:
                [d1, d2, d3] = str(row.cd_id), row.cd_title, row.cd_artist
                file.write(','.join([d1, d2, d3]) + '\n')

    @staticmethod
    def load_inventory(file_name, table):
        table.clear()  # this clears existing data and allows to load data from file
        try:
            with open(file_name, 'r') as file:
                for line in file:
                    data = line.strip().split(',')
                    dum_obj = CD(int(data[0]), data[1], data[2])
                    table.append(dum_obj)
        except FileNotFoundError:
            print('\n{} does not exist...'.format(file_name))
            print('\nCreating the file...')
            file = open(file_name, 'w')
            file.close()
            print('\nThe file, {}, has now been created!'.format(file_name))
        except Exception:
            print('\nThere was a general error...')
```

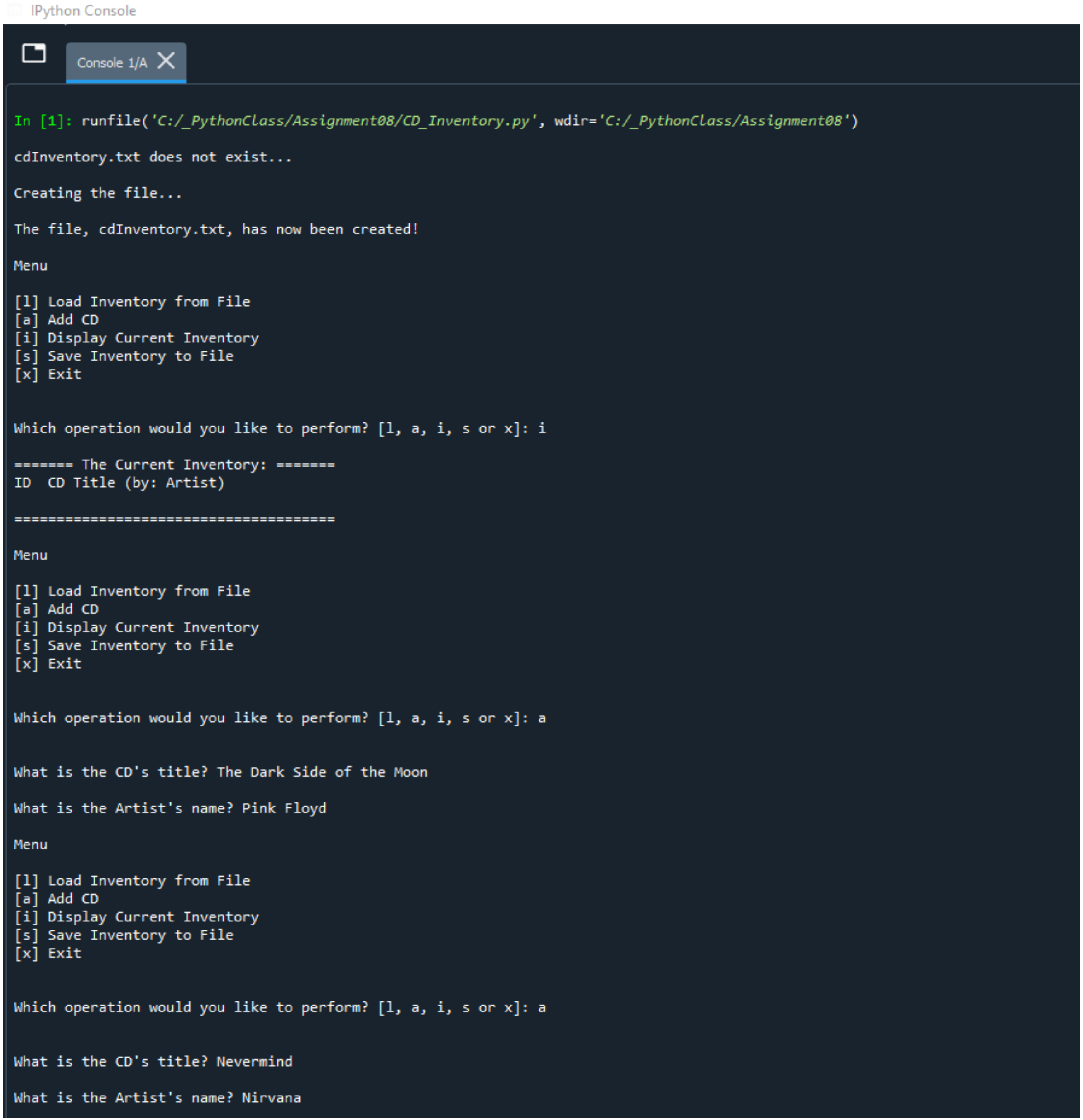
Listing 2 – Loading and Saving the Data

Input/ Output and Main Body

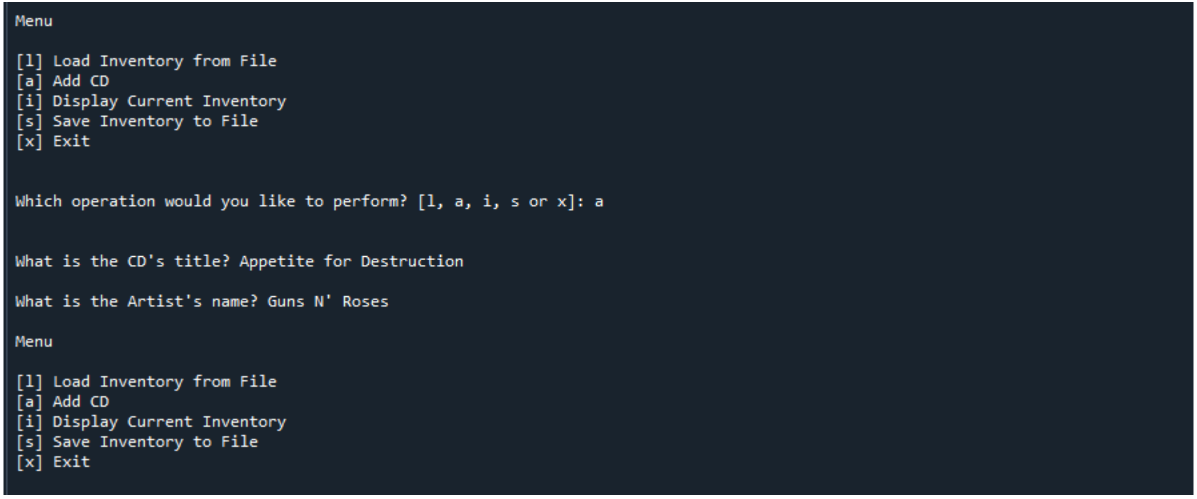
The input/ output class remained very similar the code from Assignment06 and Assignment07. The only changes required were removing the 'Delete CD' option from the menu and adjusting the IO.show_inventory method to accommodate CD objects instead of CD dictionaries.

Running the Script

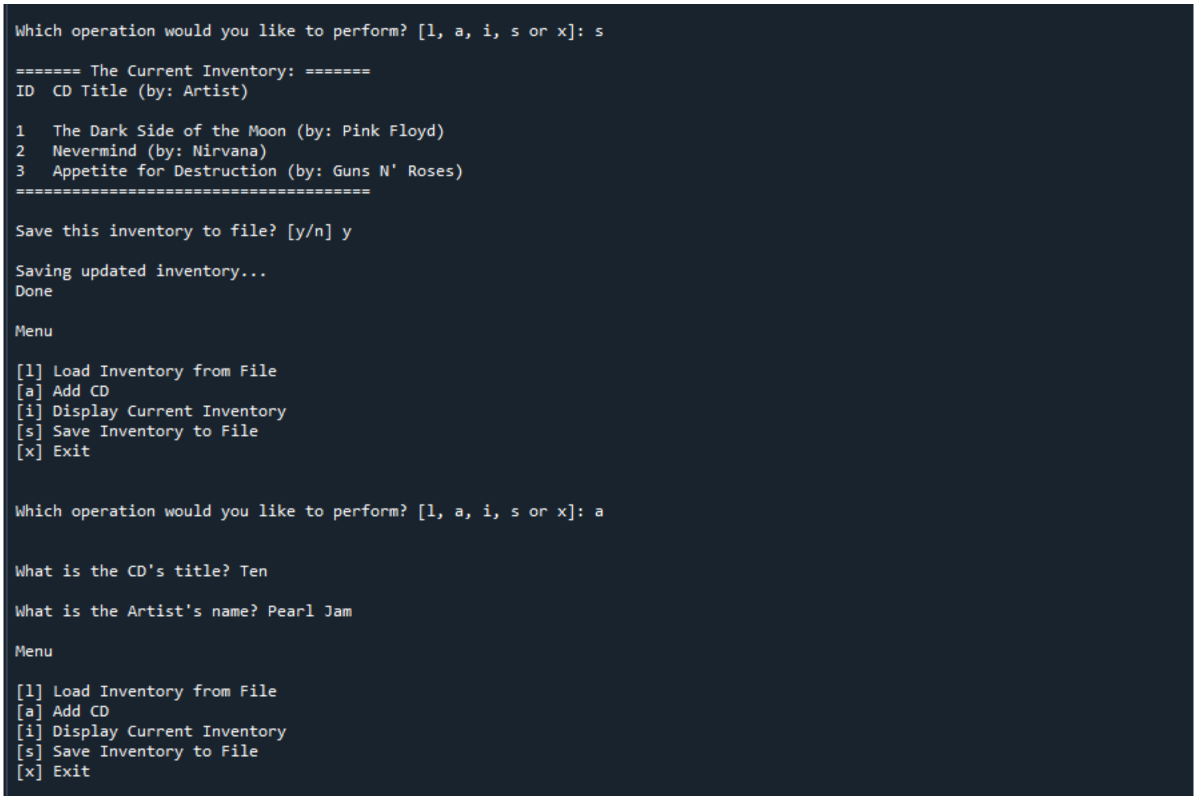
Spyder



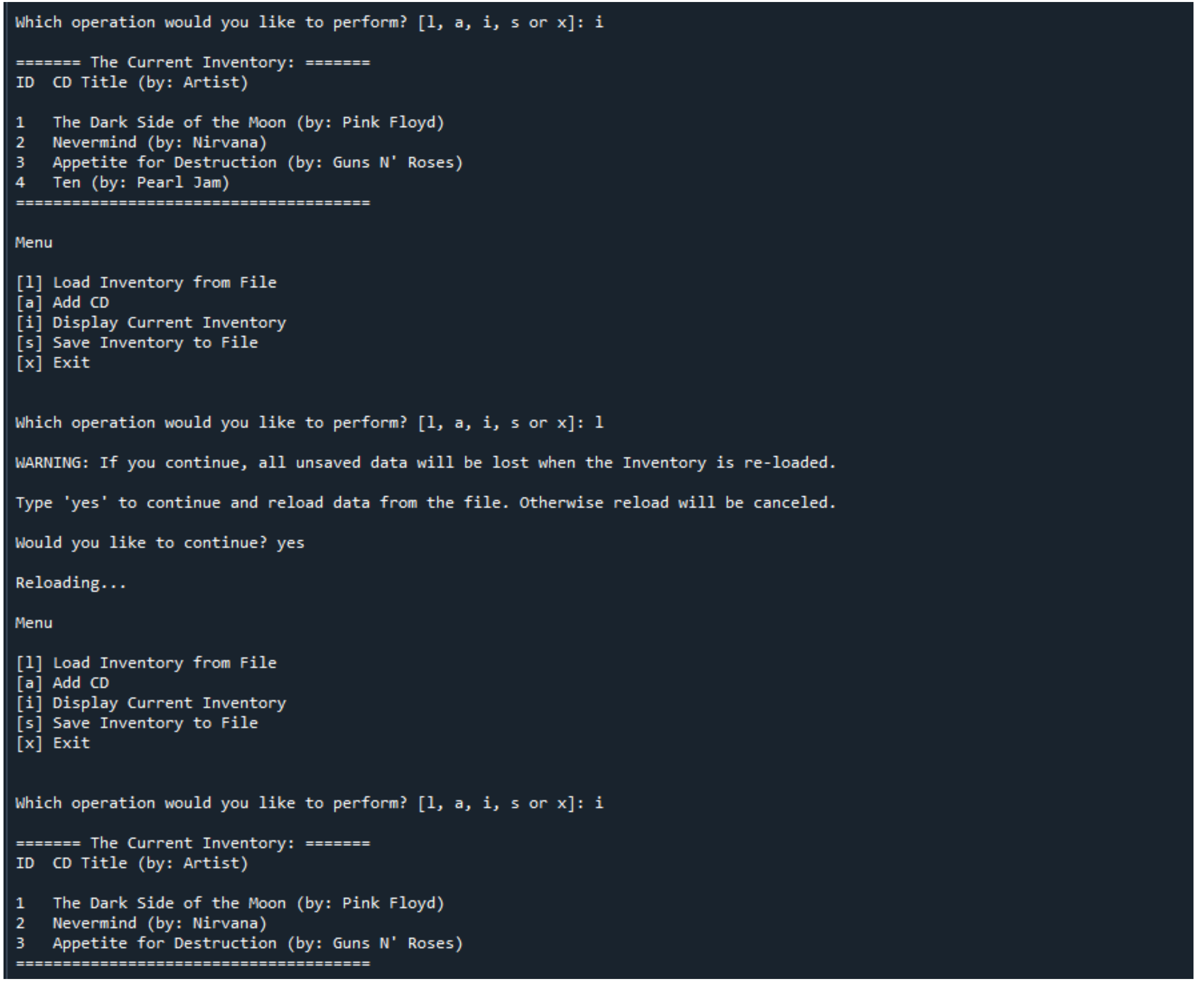
```
In [1]: runfile('C:/_PythonClass/Assignment08/CD_Inventory.py', wdir='C:/_PythonClass/Assignment08')

cdInventory.txt does not exist...

Creating the file...

The file, cdInventory.txt, has now been created!

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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

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

======================================

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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


What is the CD's title? The Dark Side of the Moon

What is the Artist's name? Pink Floyd

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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


What is the CD's title? Nevermind

What is the Artist's name? Nirvana
```



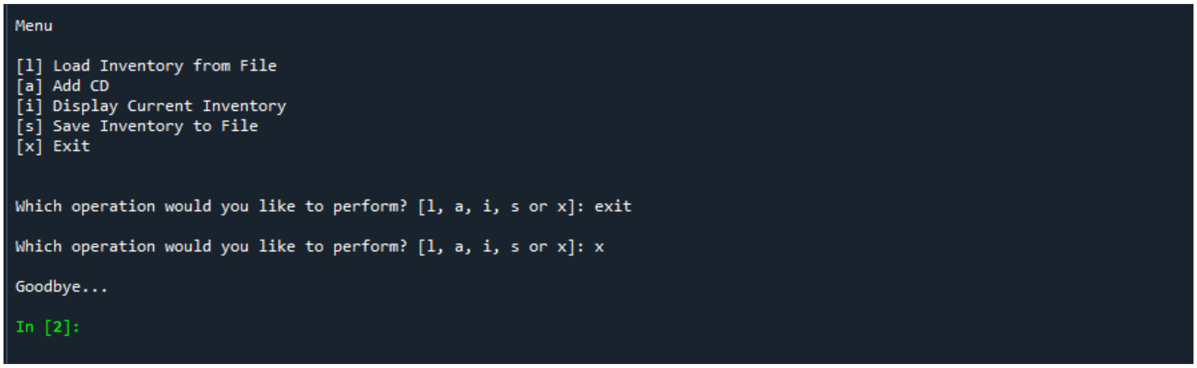
```
Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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


What is the CD's title? Appetite for Destruction

What is the Artist's name? Guns N' Roses

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit
```



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

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

1   The Dark Side of the Moon (by: Pink Floyd)
2   Nevermind (by: Nirvana)
3   Appetite for Destruction (by: Guns N' Roses)
======================================

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

Saving updated inventory...
Done

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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


What is the CD's title? Ten

What is the Artist's name? Pearl Jam

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit
```



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

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

1   The Dark Side of the Moon (by: Pink Floyd)
2   Nevermind (by: Nirvana)
3   Appetite for Destruction (by: Guns N' Roses)
4   Ten (by: Pearl Jam)
======================================

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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

WARNING: If you continue, all unsaved data will be lost when the Inventory is re-loaded.

Type 'yes' to continue and reload data from the file. Otherwise reload will be canceled.

Would you like to continue? yes

Reloading...

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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

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

1   The Dark Side of the Moon (by: Pink Floyd)
2   Nevermind (by: Nirvana)
3   Appetite for Destruction (by: Guns N' Roses)
======================================
```

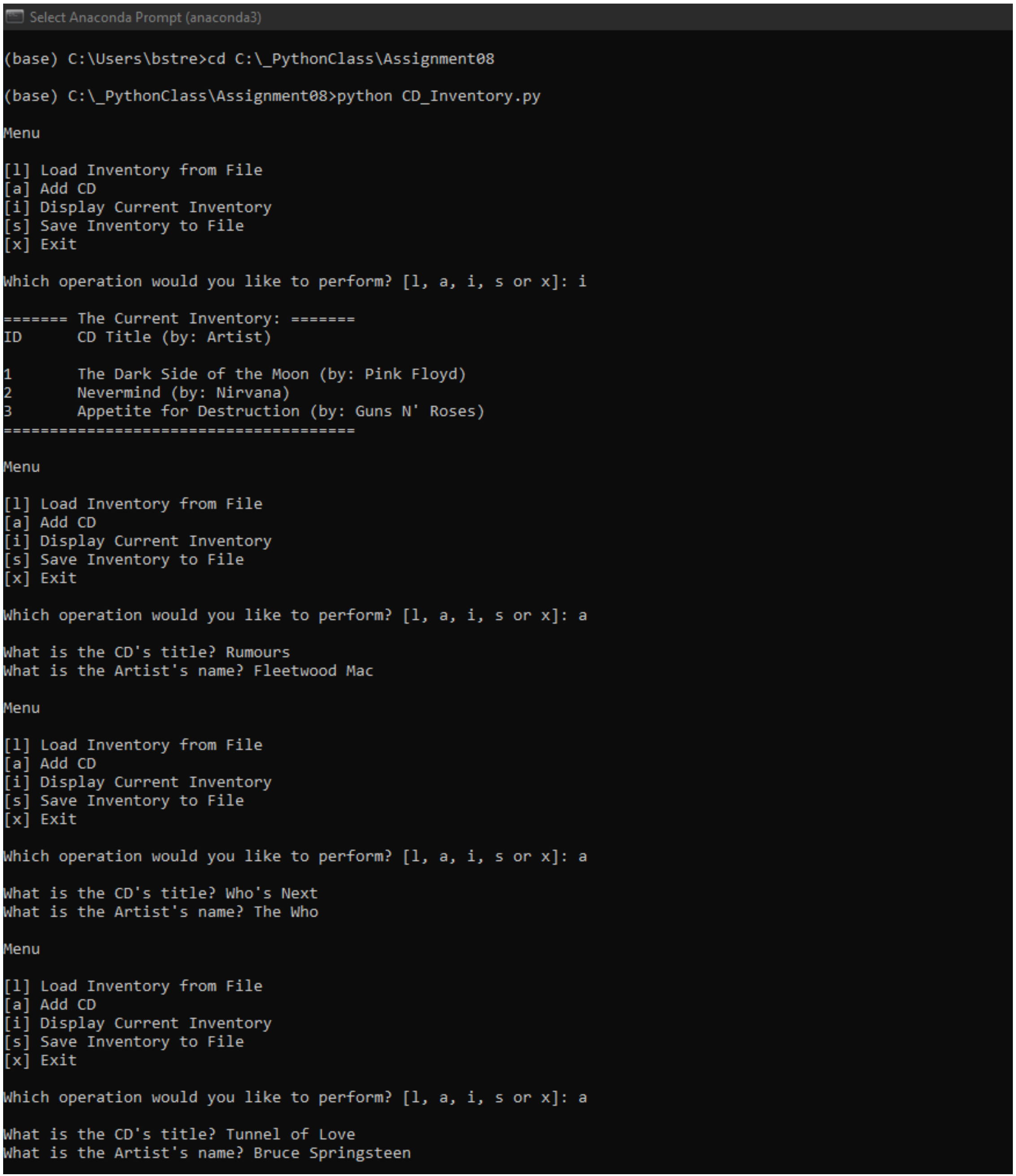


```
Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit


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

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

Goodbye...

In [2]:
```

Figure 1 – Running CD_Inventory.py in Spyder

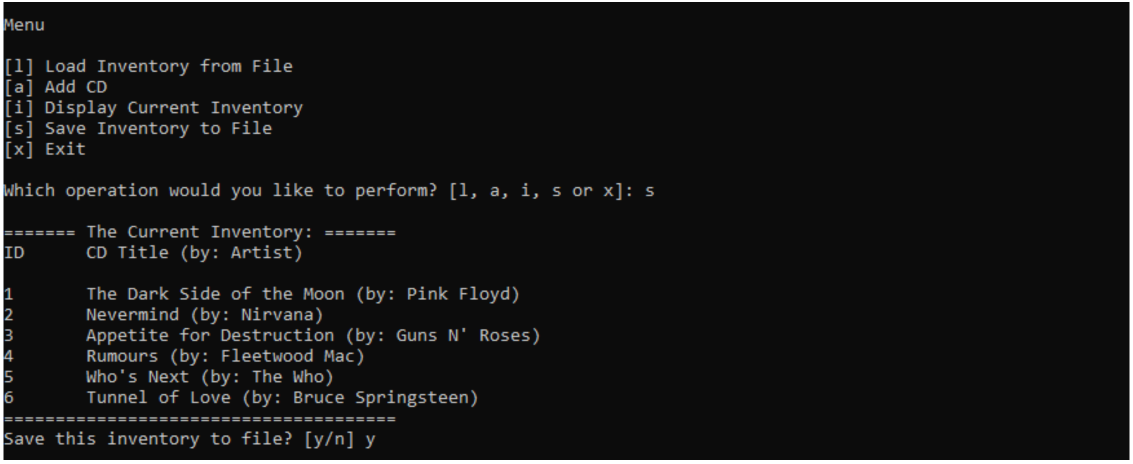
I opened Spyder on my Windows computer, opened CD_Inventory.py, and clicked F5 to execute the script. The file, cdInventory.txt did not exist yet and the program behaved correctly. It notified me that the file didn't exist and proceeded to create the file for use later in the script. I followed the user prompts and entered 'i' to display the current CD inventory. It was empty and the program behaved correctly. Next, I used 'a' three times to add three CDs and entered the necessary information when prompted. I followed that up with the 's' command to save the updated inventory. Afterwards, I used 'a' to add a fourth CD to the inventory. I used 'l' next to load the last save from cdInventory.txt. This undid my addition for the fourth CD. Next, I entered 'i' again to display the most recent CD inventory. Finally, I entered 'exit' to show the script's response to an invalid input followed by 'x' to exit the program. CD_Inventory.py ran correctly all the way through. It accepted user inputs, read data, modified data, displayed data, and saved data as intended. Figure 1 shows that the script functions correctly while running in the Spyder IDE.

Terminal

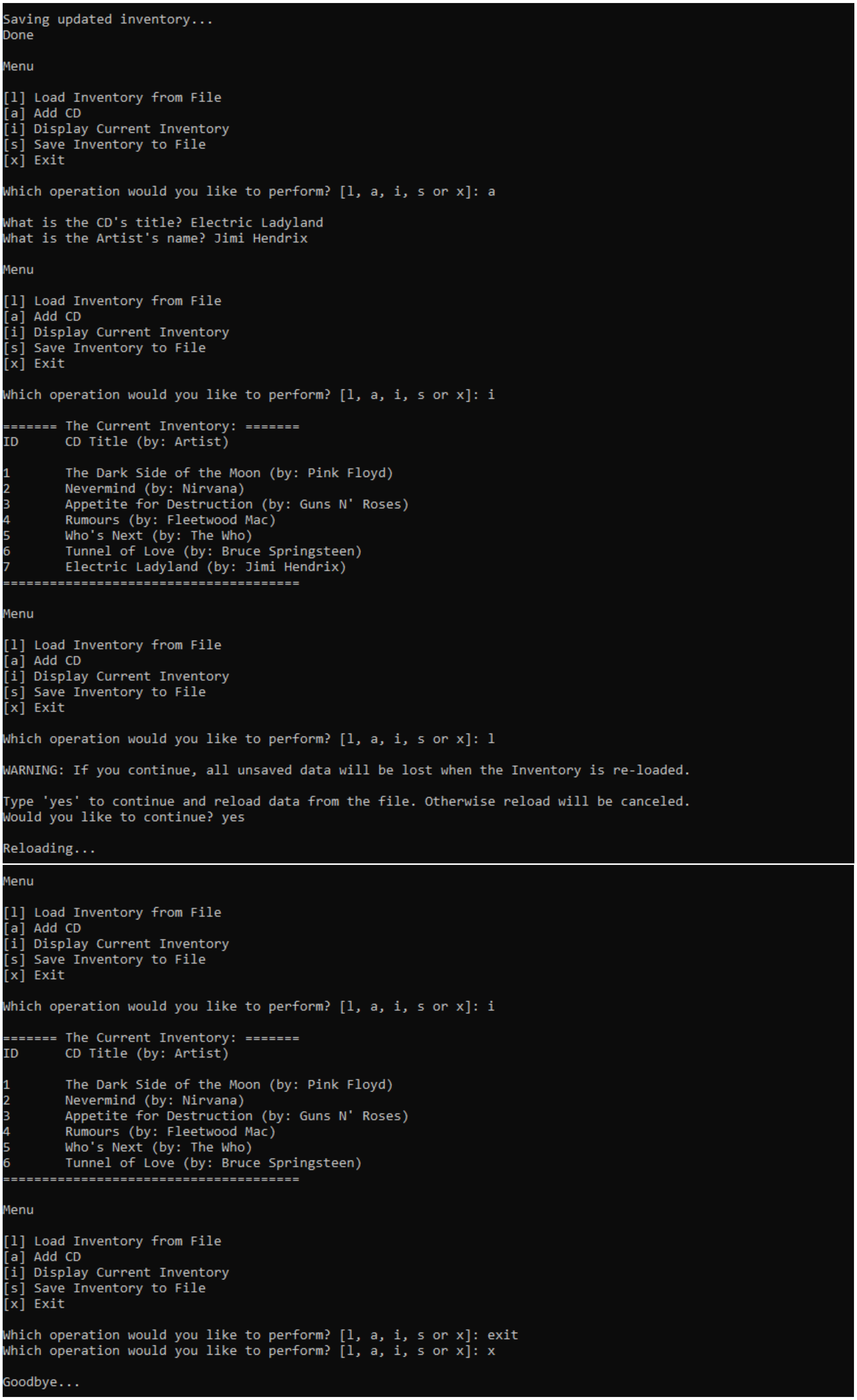


Select Anaconda Prompt (anaconda3)

```
(base) C:\Users\bstre>cd C:\_PythonClass\Assignment08

(base) C:\_PythonClass\Assignment08>python CD_Inventory.py

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

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

1       The Dark Side of the Moon (by: Pink Floyd)
2       Nevermind (by: Nirvana)
3       Appetite for Destruction (by: Guns N' Roses)
======================================

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

What is the CD's title? Rumours
What is the Artist's name? Fleetwood Mac

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

What is the CD's title? Who's Next
What is the Artist's name? The Who

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

What is the CD's title? Tunnel of Love
What is the Artist's name? Bruce Springsteen
```



```
Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

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

1       The Dark Side of the Moon (by: Pink Floyd)
2       Nevermind (by: Nirvana)
3       Appetite for Destruction (by: Guns N' Roses)
4       Rumours (by: Fleetwood Mac)
5       Who's Next (by: The Who)
6       Tunnel of Love (by: Bruce Springsteen)
======================================
Save this inventory to file? [y/n] y
```



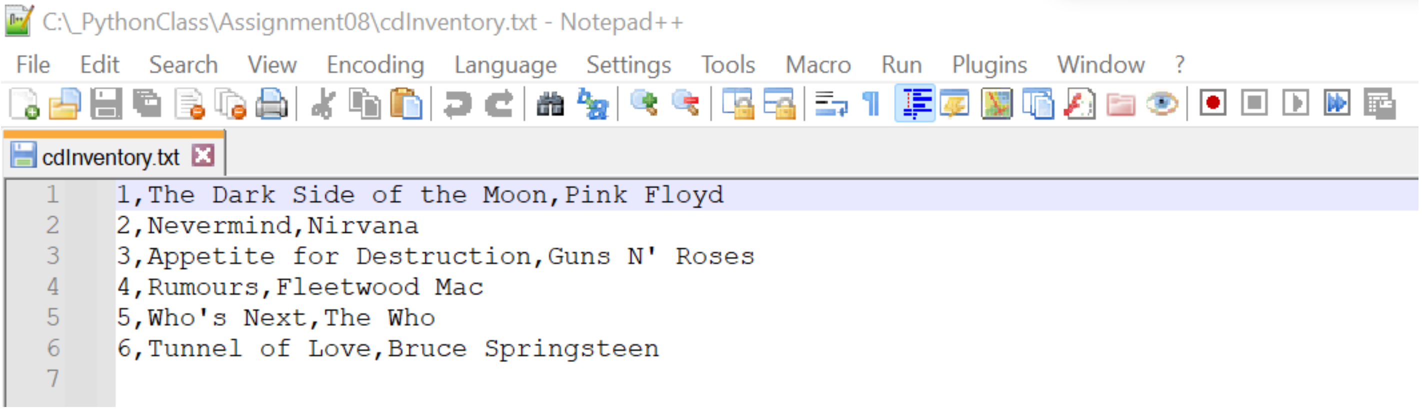
```
Saving updated inventory...
Done

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

What is the CD's title? Electric Ladyland
What is the Artist's name? Jimi Hendrix

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

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

1       The Dark Side of the Moon (by: Pink Floyd)
2       Nevermind (by: Nirvana)
3       Appetite for Destruction (by: Guns N' Roses)
4       Rumours (by: Fleetwood Mac)
5       Who's Next (by: The Who)
6       Tunnel of Love (by: Bruce Springsteen)
7       Electric Ladyland (by: Jimi Hendrix)
======================================

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

WARNING: If you continue, all unsaved data will be lost when the Inventory is re-loaded.

Type 'yes' to continue and reload data from the file. Otherwise reload will be canceled.
Would you like to continue? yes

Reloading...
Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

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

1       The Dark Side of the Moon (by: Pink Floyd)
2       Nevermind (by: Nirvana)
3       Appetite for Destruction (by: Guns N' Roses)
4       Rumours (by: Fleetwood Mac)
5       Who's Next (by: The Who)
6       Tunnel of Love (by: Bruce Springsteen)
======================================

Menu

[l] Load Inventory from File
[a] Add CD
[i] Display Current Inventory
[s] Save Inventory to File
[x] Exit

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

Goodbye...
```

Figure 2 – Running CD_Inventory.py in Terminal

I opened terminal on my Windows computer and navigated to the correct folder using the cd (change directory) command. Then, I ran the command ‘python CD_Inventory.py’ to execute the script. I followed the user prompts and entered ‘i’ to show that the program loaded the data correctly. I ran the script in terminal after I had already run it in Spyder, so there were three CDs in the inventory. Next, I used ‘a’ three times to add three CDs and entered the necessary information when prompted. I followed that up with the ‘s’ command to save the updated inventory. Afterwards, I used ‘a’ to add a seventh CD to the inventory. I used ‘l’ next to load the last save from cdInventory.txt. This undid my addition for the seventh CD. Next, I entered ‘i’ again to display the most recent CD inventory. Finally, I entered ‘exit’ to show the script’s response to an invalid input followed by ‘x’ to exit the program. CD_Inventory.py ran correctly all the way through. It accepted user inputs, read data, modified data, displayed data, and saved data as intended. Figure 2 shows that the script functions correctly while running in terminal.

Checking the Text File

After running the script in Spyder and then in a terminal window, I opened the text file to verify that it had been created, read, edited, and saved properly. As shown in Figure 3, everything worked correctly and cdInventory.txt had all the right information.



```
1,The Dark Side of the Moon,Pink Floyd
2,Nevermind,Nirvana
3,Appetite for Destruction,Guns N' Roses
4,Rumours,Fleetwood Mac
5,Who's Next,The Who
6,Tunnel of Love,Bruce Springsteen

```

Figure 3 – Verifying Script Functionality

GitHub Repository

Link: https://github.com/BenStreck/Assignment_08

Summary

I successfully created a python script that fulfills the requirements listed in Assignment 08. I did so using information from the textbook, the Module 08 videos, and the supplemental learning documents. The script demonstrates my understanding of objects, classes, functions, error handling, and data storage.

I did not encounter any struggles in this week’s assignment, but I am excited to continue working with object-oriented programming methods in the coming weeks!

Appendix

Full Listing – CD_Inventory.py

```
#------------------------------------------#
# Title: CD_Inventory.py
# Desc: Assignnment 08 - Working with Classes
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, created file
# DBiesinger, 2030-Jan-01, added pseudocode to complete assignment 08
# BStreck, 2022-Dec-03, added functionality to the TODO sections
#------------------------------------------#

```

```python
# -- DATA -- #
strFileName = 'cdInventory.txt'
lstOfCDObjects = []

class CD:
    """Stores data about a CD

    Properties:
        cd_id: (int) with CD ID
        cd_title: (string) with the title of the CD
        cd_artist: (string) with the artist of the CD

    Methods:
        noAnswer(self) --> (a string)
        __str__(self) --> self.noAnswer (a string)
    """

    # Fields
    cd_id = 0
    cd_title = ''
    cd_artist = ''

    # Constructor
    def __init__(self, num, cd, art):

    # Attributes
        self.__cd_id = num
        self.__cd_title = cd
        self.__cd_artist = art

    # Properties
    @property
    def cd_id(self):
        return self.__cd_id

    @cd_id.setter
    def cd_id(self, value):
        if str(value).isnumeric():
            self.__cd_id = value
        else:
            raise Exception('The cd_id must be a positive integer')

    @property
    def cd_title(self):
        return self.__cd_title

    @cd_title.setter
    def cd_title(self, value):
        self.__cd_title = value

    @property
    def cd_artist(self):
        return self.__cd_artist

    @cd_artist.setter
    def cd_artist(self, value):
            self.__cd_artist = value

    # Methods
    def noAnswer(self):
        return """
# ------------------------------ #
This object has three properties:
  - cd_id
  - cd_title
  - cd_artist
# ------------------------------ #
                """

    def __str__(self):
        return self.noAnswer()


# -- PROCESSING -- #
class FileIO:
    """Processes data to and from file

    Methods:
        save_inventory(file_name, table): --> None
        load_inventory(file_name, table): --> None
    """

    @staticmethod
    def save_inventory(file_name, table):
        """
        Method to manage data writing from the list of dictionaries to a text file.
        It shows the current inventory prior to saving which allows users to verify they are saving the correct data.

        Args:
            file_name (string): name of file used to read the data from
            table (list of CD objects): 2D structure that holds the data during runtime

        Returns:
            None
        """
        with open(file_name, 'w') as file:
            for row in table:
                [d1, d2, d3] = str(row.cd_id), row.cd_title, row.cd_artist
                file.write(','.join([d1, d2, d3]) + '\n')

    @staticmethod
    def load_inventory(file_name, table):
        """
        Method to manage data intake from the text file to a list of CD objects.
        The method reads data from the file identified by 'file_name' into a 2D table (list of CD objects).
        One line in the file represents one object in the table.

        This also contains error handling if there is a FileNotFoundError.

        Args:
            file_name (string): name of file used to read the data from
            table (list of CD objects): 2D structure that holds the data during runtime

        Returns:
            None
        """
        table.clear()  # this clears existing data and allows to load data from file
        try:
```

```python
            with open(file_name, 'r') as file:
                for line in file:
                    data = line.strip().split(',')
                    dum_obj = CD(int(data[0]), data[1], data[2])
                    table.append(dum_obj)
        except FileNotFoundError:
            print('\n{} does not exist...'.format(file_name))
            print('\nCreating the file...')
            file = open(file_name, 'w')
            file.close()
            print('\nThe file, {}, has now been created!'.format(file_name))
        except Exception:
            print('\nThere was a general error...')


# -- PRESENTATION (Input/Output) -- #
class IO:
    """
    Handling Input / Output

    Methods:
        print_menu(): --> None
        menu_choice(): --> choice (a string)
        show_inventory(table): --> None
        new_CD_choice(table): --> ID (an integer), strTitle (a string), strArtist (a string)
    """

    @staticmethod
    def print_menu():
        """
        Displays a menu of choices to the user

        Args:
            None

        Returns:
            None
        """
        print('\nMenu\n\n[l] Load Inventory from File\n[a] Add CD\n[i] Display Current Inventory')
        print('[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 string of the users input out of the choices l, a, i, s or x
        """
        choice = ' '
        while choice not in ['l', 'a', 'i', 's', 'x']:
            choice = input('Which operation would you like to perform? [l, a, i, s or x]: ').lower().strip()
        print()  # Add extra space for layout
        return choice

    @staticmethod
    def show_inventory(table):
        """
        Displays the current inventory table

        Args:
            table (list of CD objects): 2D structure that holds the data during runtime

        Returns:
            None
        """
        print('======= The Current Inventory: =======')
        print('ID\tCD Title (by: Artist)\n')
        for obj in table:
            print('{}\t{} (by: {})'.format(obj.cd_id, obj.cd_title, obj.cd_artist))
        print('======================================')

    @staticmethod
    def new_CD_choice(table):
        """
        Method to accept user inputs for a new CD

        Args:
            table (list of CD objects): 2D structure that holds the data during runtime

        Returns:
            ID (integer): the ID number of the new CD being added to the inventory
            strTitle (string): the title of the new CD being added to the inventory
            strArtist (string): the artist of the new CD being added to the inventory
        """
        ID = len(table) + 1
        strTitle = input('What is the CD\'s title? ').strip()
        strArtist = input('What is the Artist\'s name? ').strip()
        return ID, strTitle, strArtist


# -- Main Body of Script -- #

# 1. Load data from the file into a list of CD objects when the script starts
FileIO.load_inventory(strFileName, lstOfCDObjects)

# 2. Start main loop
while True:

    # 3. Display menu to user and get choice
    IO.print_menu()
    strChoice = IO.menu_choice()

    # 4. Process menu selections

        # 4.1 Exit
    if strChoice == 'x':
        print('Goodbye...')
        break

        # 4.2 Load Inventory
    if strChoice == 'l':
        print('WARNING: If you continue, all unsaved data will be lost when the Inventory is re-loaded.\n')
        print('Type \'yes\' to continue and reload data from the file. Otherwise reload will be canceled.')
        strYesNo = input('Would you like to continue? ')
        if strYesNo.lower() == 'yes':
            print('\nReloading...')
            FileIO.load_inventory(strFileName, lstOfCDObjects)
        else:
            input('Canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')
        continue

        # 4.3 Add a CD
    elif strChoice == 'a':
        dum1 = None
        ID, strTitle, strArtist = IO.new_CD_choice(lstOfCDObjects)
        dum1 = CD(ID, strTitle, strArtist)
        lstOfCDObjects.append(dum1)
        continue  # start loop back at top.

        # 4.4 Display Current Inventory
    elif strChoice == 'i':
        IO.show_inventory(lstOfCDObjects)
        continue  # start loop back at top.

        # 4.5 Save Inventory to File
    elif strChoice == 's':
        IO.show_inventory(lstOfCDObjects)
        strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
        if strYesNo == 'y':
            print('\nSaving updated inventory...')
            FileIO.save_inventory(strFileName, lstOfCDObjects)
            print('Done')
        else:
            input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
        continue  # start loop back at top.

        # 4.6 Catch-All Error... Should not be possible because the user's choice gets vetted in IO
    else:
        print('Invalid Input...\n')
        print('Please choose one of the options listed\n')
        continue
```