Sonia Romo
2020-AugI-18
IT FDN 110 B - Foundations of Programming, Python
Assignment 06

# Using Functions

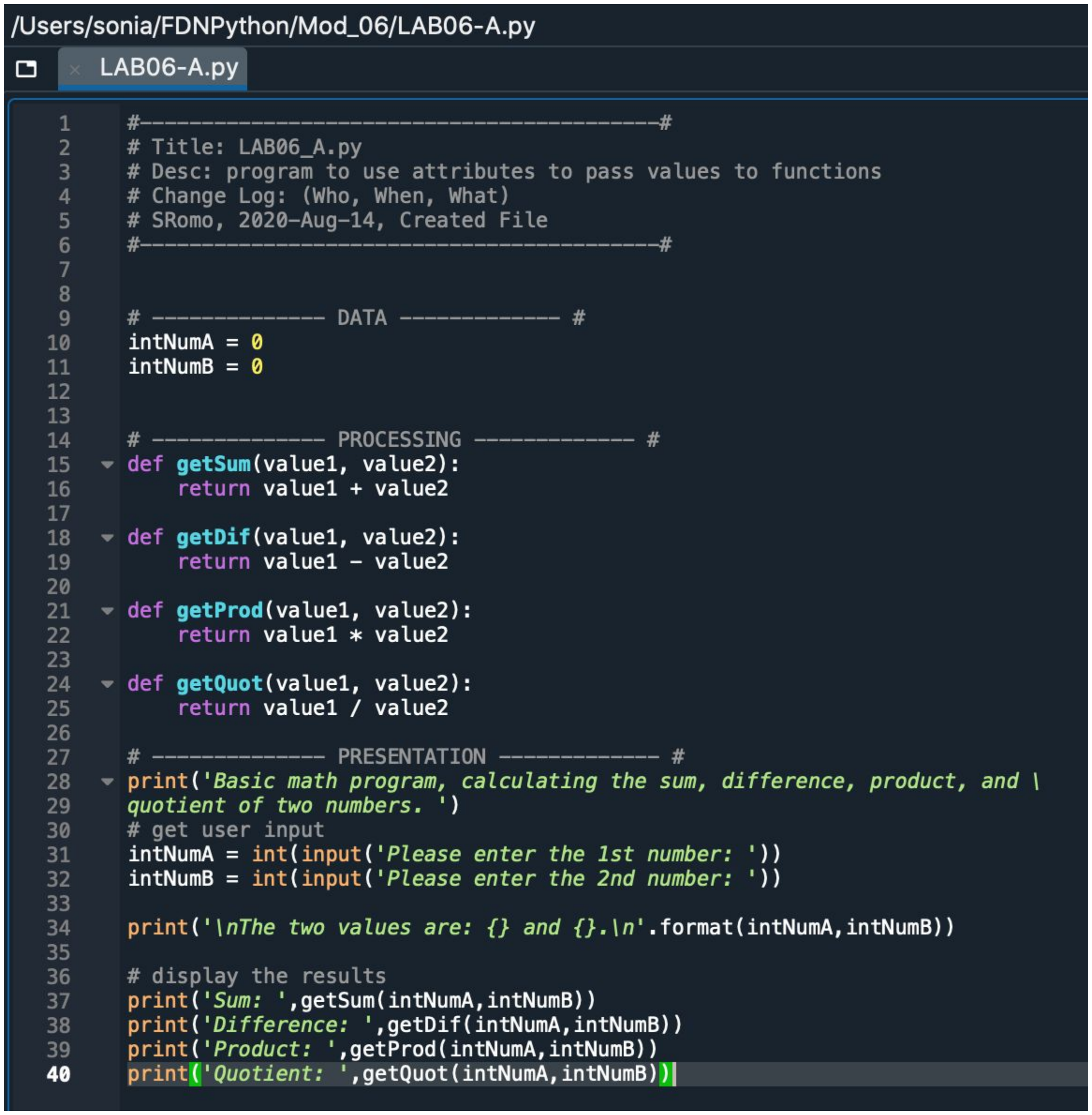
## Introduction

This week, we learned how to build and use functions, including how to input multiple values and variables using arguments and parameters, as well as how to return values. We also learned about global and local variables.

## Labs

### LAB06-A

In this lab, my code first declares two variables, then uses four different functions to perform basic math, then asks the user to input two numbers, and then uses the user input in the functions.

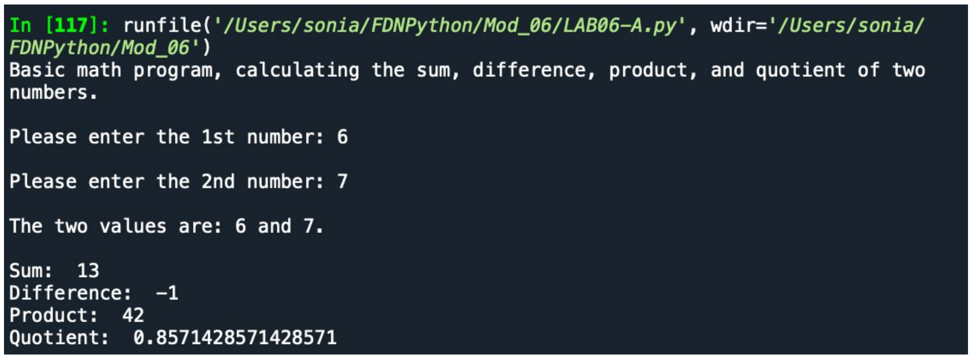


/Users/sonia/FDNPython/Mod_06/LAB06-A.py

LAB06-A.py

```python
#-----------------------------------------------#
# Title: LAB06_A.py
# Desc: program to use attributes to pass values to functions
# Change Log: (Who, When, What)
# SRomo, 2020-Aug-14, Created File
#-----------------------------------------------#


# -------------- DATA -------------- #
intNumA = 0
intNumB = 0


# -------------- PROCESSING -------------- #
def getSum(value1, value2):
    return value1 + value2

def getDif(value1, value2):
    return value1 - value2

def getProd(value1, value2):
    return value1 * value2

def getQuot(value1, value2):
    return value1 / value2

# -------------- PRESENTATION -------------- #
print('Basic math program, calculating the sum, difference, product, and \
quotient of two numbers. ')
# get user input
intNumA = int(input('Please enter the 1st number: '))
intNumB = int(input('Please enter the 2nd number: '))

print('\nThe two values are: {} and {}.\n'.format(intNumA,intNumB))

# display the results
print('Sum: ',getSum(intNumA,intNumB))
print('Difference: ',getDif(intNumA,intNumB))
print('Product: ',getProd(intNumA,intNumB))
print('Quotient: ',getQuot(intNumA,intNumB))
```

*Figure 01 - LAB06-A code*



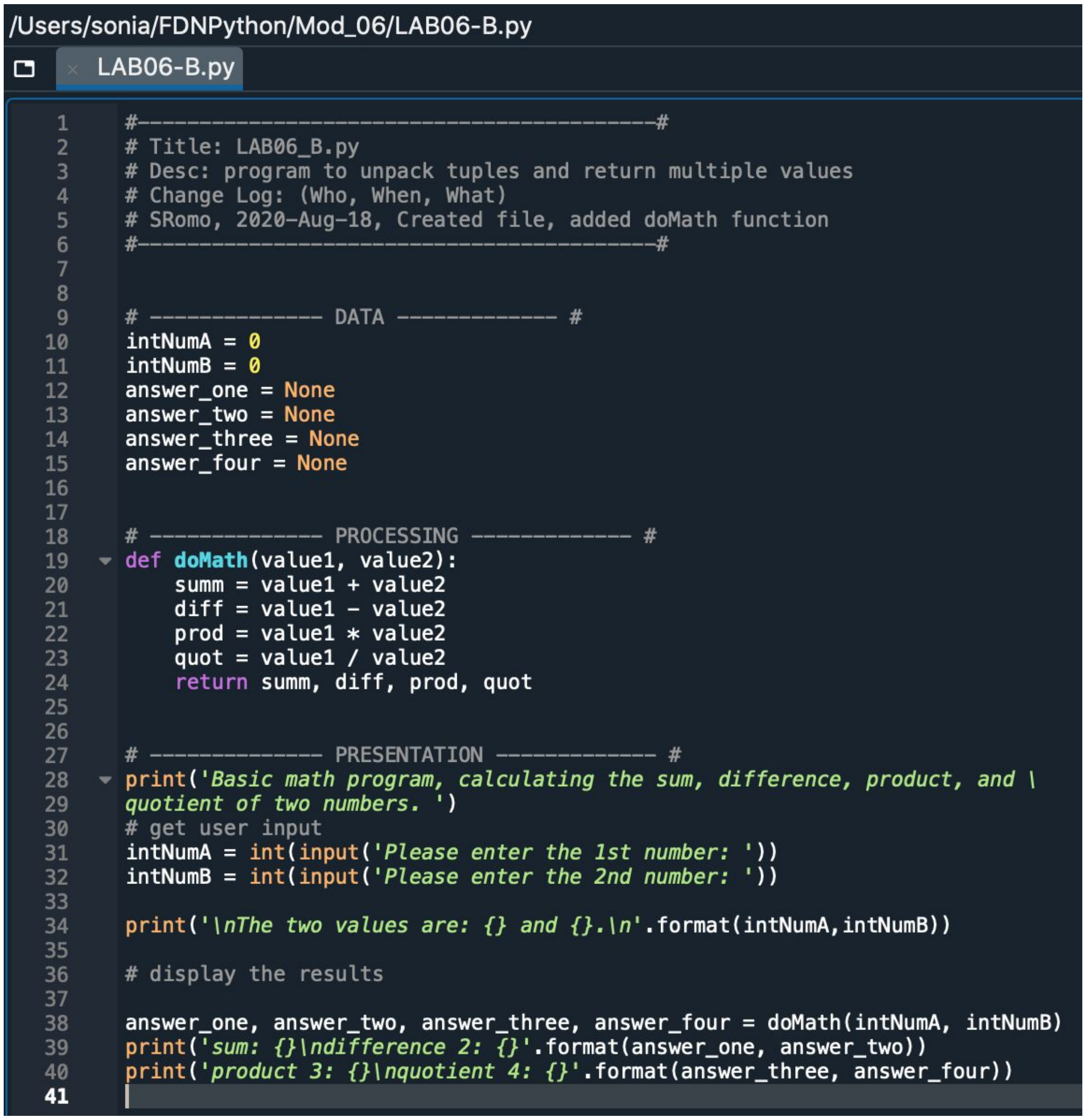
```
In [117]: runfile('/Users/sonia/FDNPython/Mod_06/LAB06-A.py', wdir='/Users/sonia/
FDNPython/Mod_06')
Basic math program, calculating the sum, difference, product, and quotient of two
numbers.

Please enter the 1st number: 6

Please enter the 2nd number: 7

The two values are: 6 and 7.

Sum:  13
Difference:  -1
Product:  42
Quotient:  0.8571428571428571
```

*Figure 02 - LAB06-A results*

## LAB06-B

In this lab, I modified Lab A to complete all four math equations in one function, using the same two values inputted by the user for each of the equations.

It first declares all variables, four additional variables. It then does all equations in one function, asks the user for input, unpacks the function results tuple into variables, and then includes the variables in print statements.

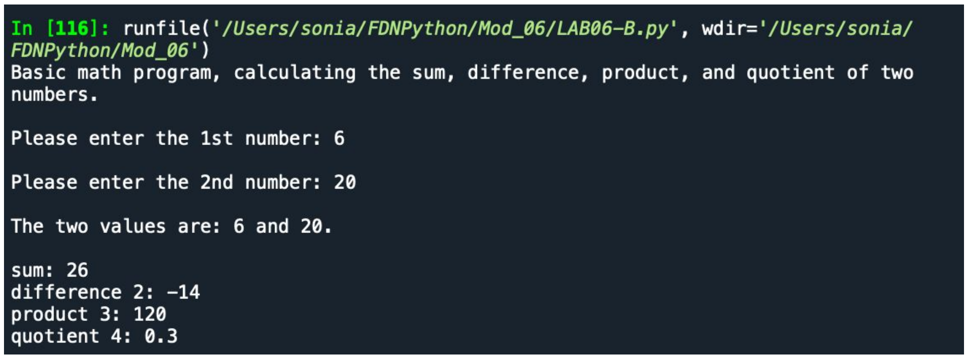


/Users/sonia/FDNPython/Mod_06/LAB06-B.py

LAB06-B.py

```python
#------------------------------------------#
# Title: LAB06_B.py
# Desc: program to unpack tuples and return multiple values
# Change Log: (Who, When, What)
# SRomo, 2020-Aug-18, Created file, added doMath function
#------------------------------------------#


# -------------- DATA ------------- #
intNumA = 0
intNumB = 0
answer_one = None
answer_two = None
answer_three = None
answer_four = None


# -------------- PROCESSING ------------- #
def doMath(value1, value2):
    summ = value1 + value2
    diff = value1 - value2
    prod = value1 * value2
    quot = value1 / value2
    return summ, diff, prod, quot


# -------------- PRESENTATION ------------- #
print('Basic math program, calculating the sum, difference, product, and \
quotient of two numbers. ')
# get user input
intNumA = int(input('Please enter the 1st number: '))
intNumB = int(input('Please enter the 2nd number: '))

print('\nThe two values are: {} and {}.\n'.format(intNumA,intNumB))

# display the results

answer_one, answer_two, answer_three, answer_four = doMath(intNumA, intNumB)
print('sum: {}\ndifference 2: {}'.format(answer_one, answer_two))
print('product 3: {}\nquotient 4: {}'.format(answer_three, answer_four))

```

*Figure 03 - LAB06-B code*



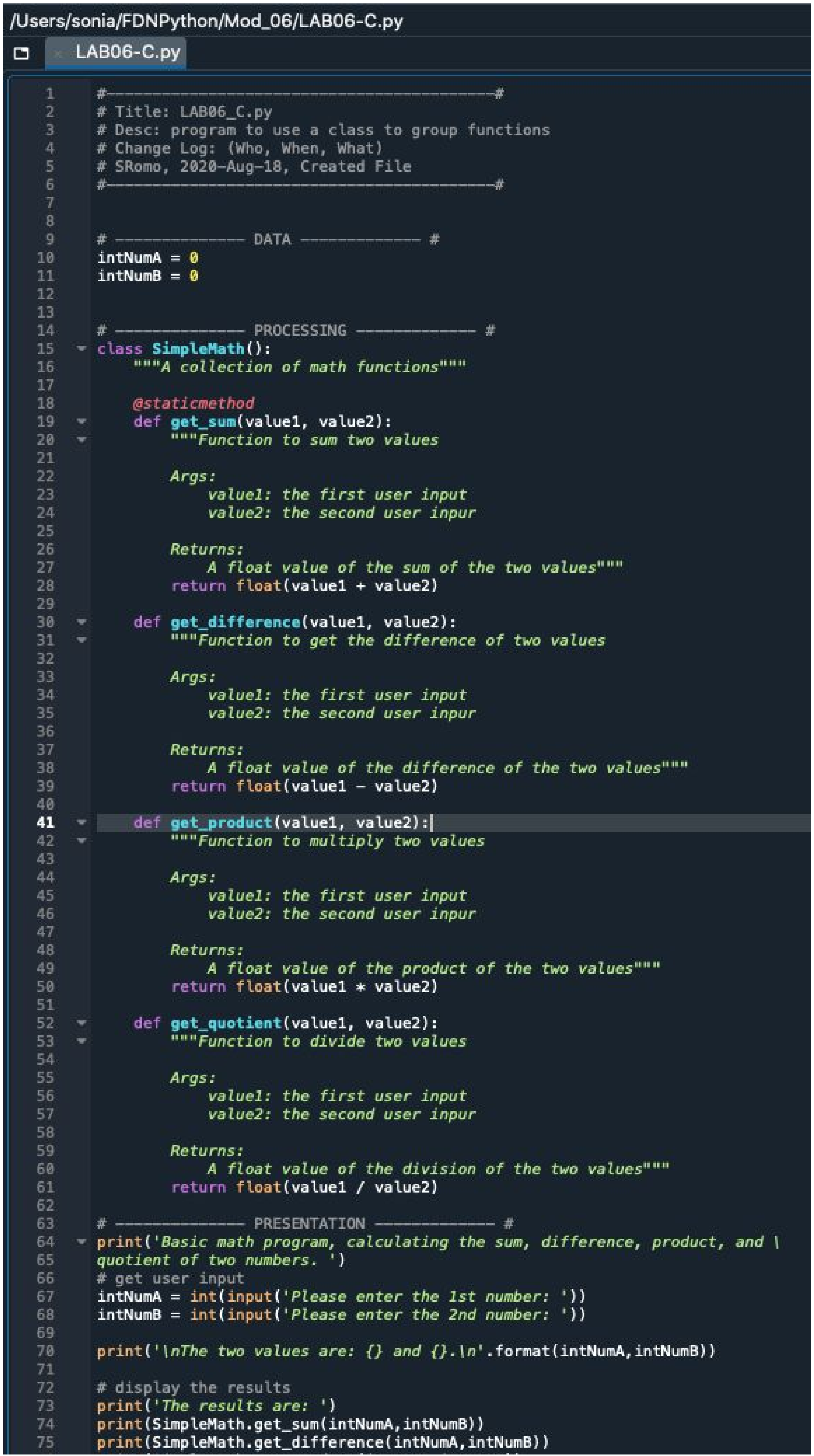
```
In [116]: runfile('/Users/sonia/FDNPython/Mod_06/LAB06-B.py', wdir='/Users/sonia/
FDNPython/Mod_06')
Basic math program, calculating the sum, difference, product, and quotient of two
numbers.

Please enter the 1st number: 6

Please enter the 2nd number: 20

The two values are: 6 and 20.

sum: 26
difference 2: -14
product 3: 120
quotient 4: 0.3
```

*Figure 04 - LAB06-B results*

## LAB06-C

In this lab, I created a class of functions to include each equation as a separate function. I also included docstrings to explain what each function does.

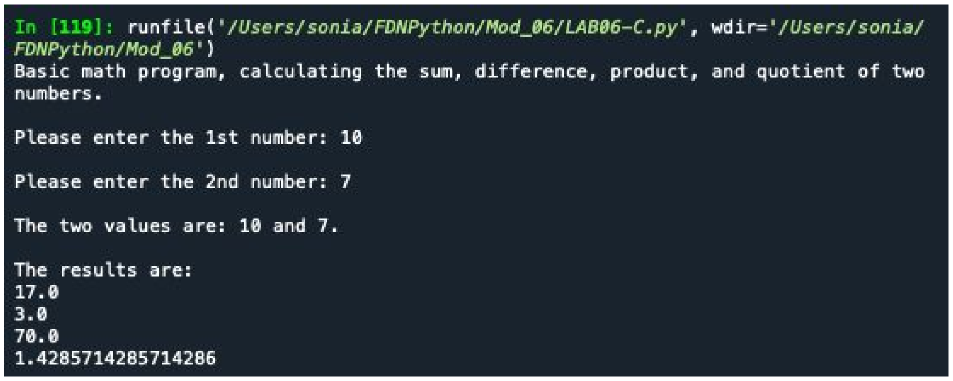
First, the script declares the variable. It then groups the functions into a class called `SimpleMath()`. It then describes each function and returns the values as floats. Lastly, it asks for the user input and performs the class of functions.


/Users/sonia/FDNPython/Mod_06/LAB06-C.py

LAB06-C.py

```
#-----------------------------------------#
# Title: LAB06_C.py
# Desc: program to use a class to group functions
# Change Log: (Who, When, What)
# SRomo, 2020-Aug-18, Created File
#-----------------------------------------#


# -------------- DATA ------------- #
intNumA = 0
intNumB = 0


# -------------- PROCESSING ------------- #
class SimpleMath():
    """A collection of math functions"""

    @staticmethod
    def get_sum(value1, value2):
        """Function to sum two values

        Args:
            value1: the first user input
            value2: the second user inpur

        Returns:
            A float value of the sum of the two values"""
        return float(value1 + value2)

    def get_difference(value1, value2):
        """Function to get the difference of two values

        Args:
            value1: the first user input
            value2: the second user inpur

        Returns:
            A float value of the difference of the two values"""
        return float(value1 - value2)

    def get_product(value1, value2):
        """Function to multiply two values

        Args:
            value1: the first user input
            value2: the second user inpur

        Returns:
            A float value of the product of the two values"""
        return float(value1 * value2)

    def get_quotient(value1, value2):
        """Function to divide two values

        Args:
            value1: the first user input
            value2: the second user inpur

        Returns:
            A float value of the division of the two values"""
        return float(value1 / value2)

# -------------- PRESENTATION ------------- #
print('Basic math program, calculating the sum, difference, product, and \
quotient of two numbers. ')
# get user input
intNumA = int(input('Please enter the 1st number: '))
intNumB = int(input('Please enter the 2nd number: '))

print('\nThe two values are: {} and {}.\n'.format(intNumA,intNumB))

# display the results
print('The results are: ')
print(SimpleMath.get_sum(intNumA,intNumB))
print(SimpleMath.get_difference(intNumA,intNumB))
```

*Figure 05 - LAB06-C code*



```
In [119]: runfile('/Users/sonia/FDNPython/Mod_06/LAB06-C.py', wdir='/Users/sonia/
FDNPython/Mod_06')
Basic math program, calculating the sum, difference, product, and quotient of two
numbers.

Please enter the 1st number: 10

Please enter the 2nd number: 7

The two values are: 10 and 7.

The results are:
17.0
3.0
70.0
1.4285714285714286
```

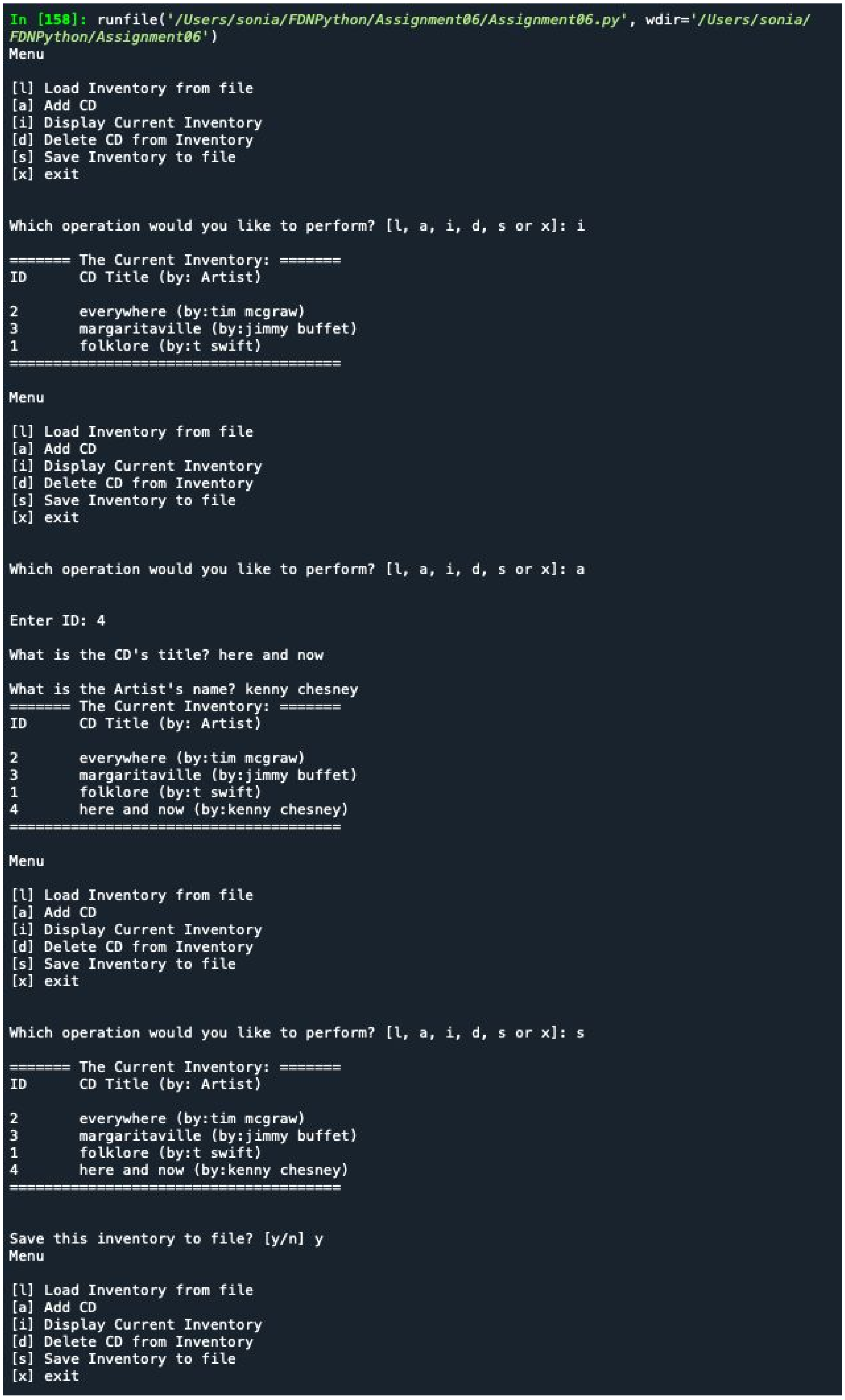
*Figure 06 - LAB06-C results*

## Homework - Using Functions

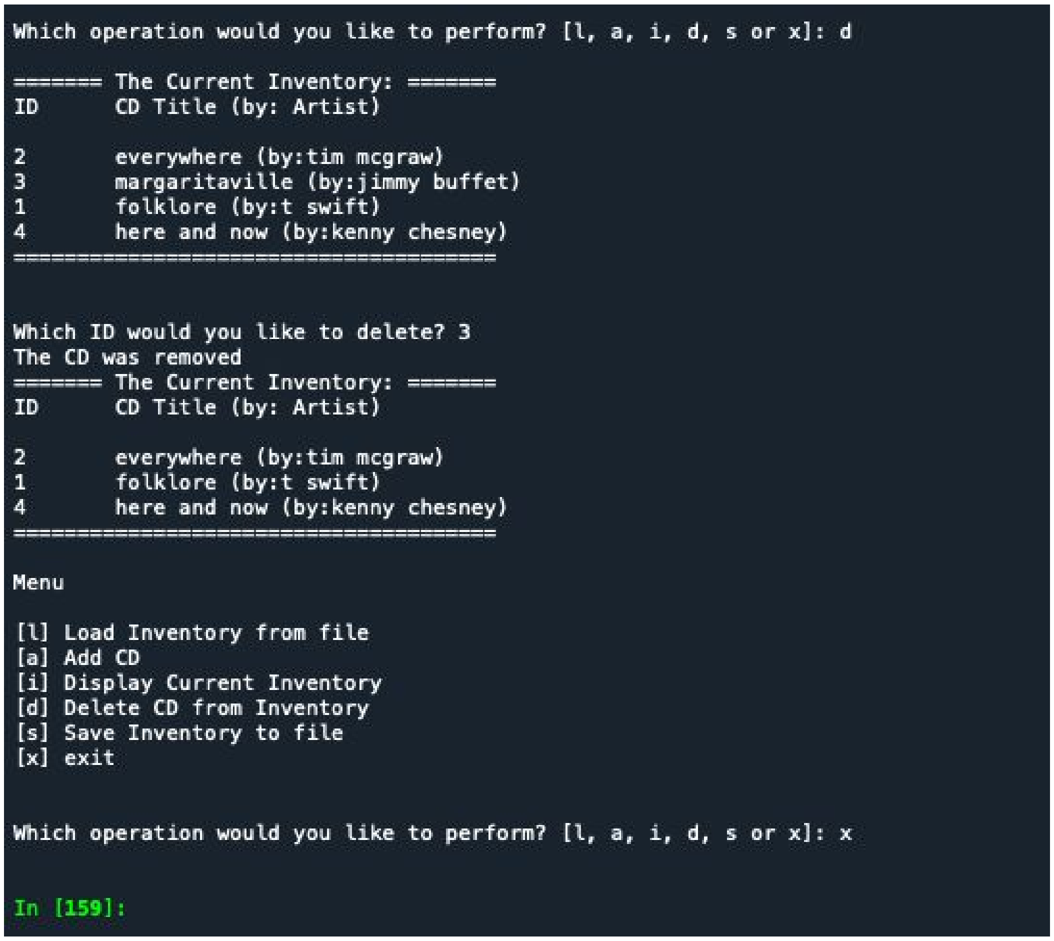
This week's homework was a little easier for me because the code to put into the functions was mostly written, we just needed to modify it to work in a function.

One area that gave me trouble was unpacking the tuple into variables for adding a CD. I was trying to unpack the tuple after running the function `IO.add_cd()`. As soon as I combined the two ideas (unpacking a tuple and running the function) into one line of code (line 211), it solved my problem.

Another area that gave me trouble was the `FileProcessor.write_file()` function. I modified the code to work as a function, but completely forgot to pass the arguments into the function when calling it (line 240). What solved my issue was literally closing my laptop for the night on Tuesday and reopening on Wednesday with a clear mind - immediately I realized that I needed to add the arguments to line 240.

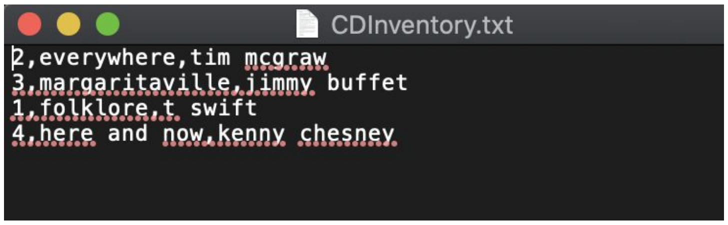


```
In [158]: runfile('/Users/sonia/FDNPython/Assignment06/Assignment06.py', wdir='/Users/sonia/
FDNPython/Assignment06')
Menu

[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]: i

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

2       everywhere (by:tim mcgraw)
3       margaritaville (by:jimmy buffet)
1       folklore (by:t swift)
======================================

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? here and now

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

2       everywhere (by:tim mcgraw)
3       margaritaville (by:jimmy buffet)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
======================================

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       everywhere (by:tim mcgraw)
3       margaritaville (by:jimmy buffet)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
======================================


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]: d

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

2       everywhere (by:tim mcgraw)
3       margaritaville (by:jimmy buffet)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
======================================


Which ID would you like to delete? 3
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       everywhere (by:tim mcgraw)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
======================================

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


In [159]:
```

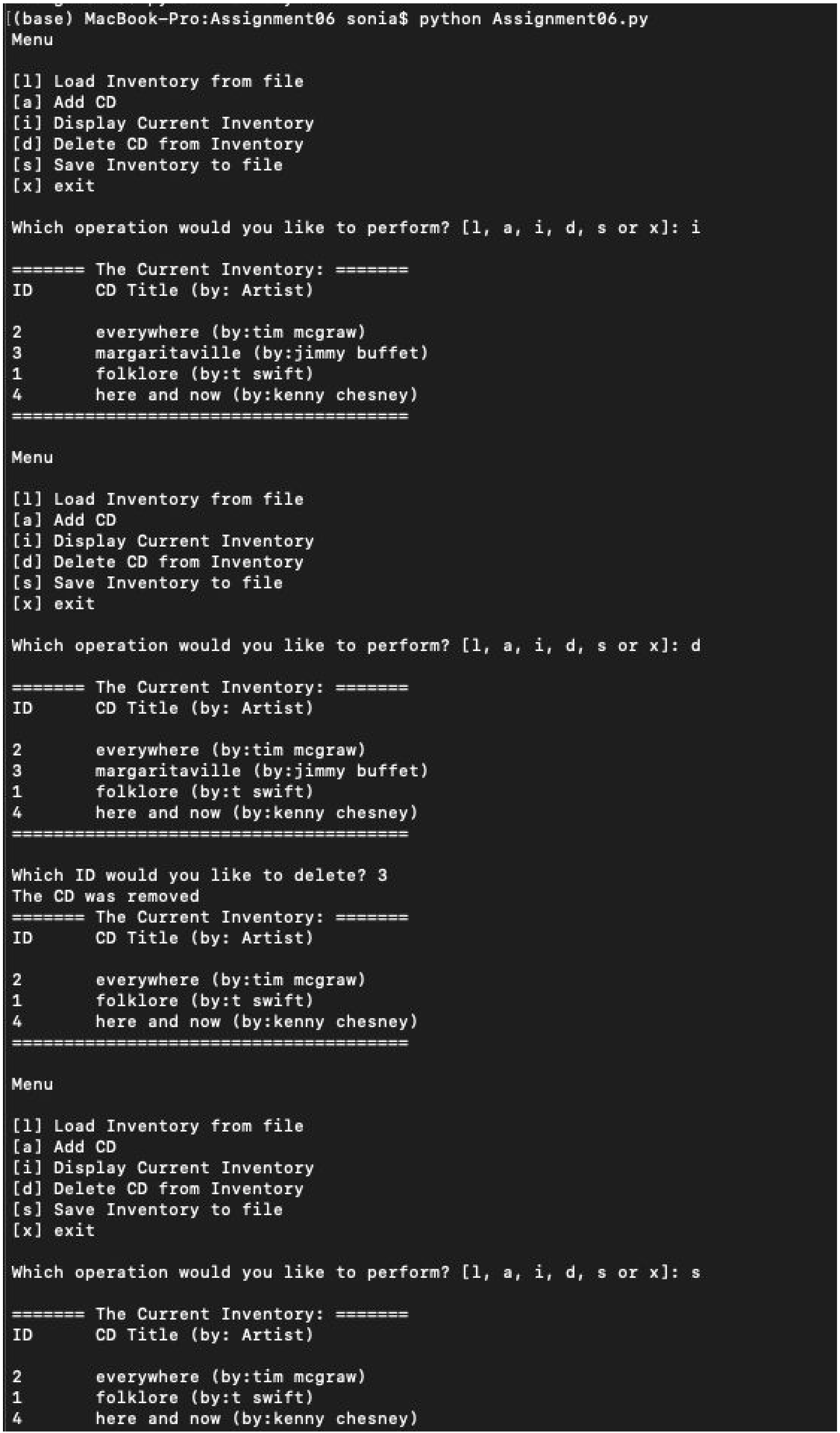
*Figure 07 - Spyder run*



CDInventory.txt

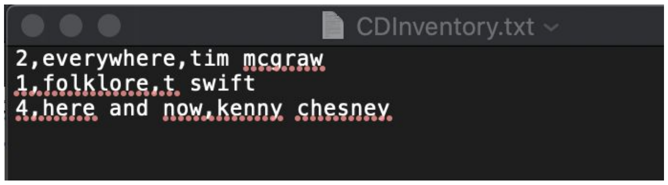
```
2,everywhere,tim mcgraw
3,margaritaville,jimmy buffet
1,folklore,t swift
4,here and now,kenny chesney
```

*Figure 08 - txt file after Spyder run*



```
[(base) MacBook-Pro:Assignment06 sonia$ 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]: i

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

2       everywhere (by:tim mcgraw)
3       margaritaville (by:jimmy buffet)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
======================================

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)

2       everywhere (by:tim mcgraw)
3       margaritaville (by:jimmy buffet)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
======================================

Which ID would you like to delete? 3
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       everywhere (by:tim mcgraw)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
======================================

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       everywhere (by:tim mcgraw)
1       folklore (by:t swift)
4       here and now (by:kenny chesney)
```

*Figure 09 - CLI run*



```
2,everywhere,tim mcgraw
1,folklore,t swift
4,here and now,kenny chesney
```

*Figure 10 - txt file after CLI run*

## Summary

This week, my knowledge of using functions was really solidified in how to set them up, how to pass parameters and arguments, and how to use return values in other functions. I also used `.format()` for the first time and feel more comfortable with that concept. Lastly, unpacking tuples made more practical sense as I could see a solid use for why one would unpack a taple.

## Appendix

Complete homework source code:

```python
#------------------------------------------#
# Title: Assignment06.py
# Desc: Working with classes and functions.
# Change Log: (Who, When, What)
# SRomo, 2020-Aug-18, Created File
# SRomo, 2020-Aug-18, Added add_cd functions, remove_cd function
# SRomo, 2020-Aug-19, Added write to file function
#------------------------------------------#

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


# -- PROCESSING -- #
class DataProcessor:
    """Processing the in-memory data"""

    @staticmethod
    def add_cd(intID, strTitle, strArtist):
        """Function to add the user input into the table

        Args:
            intID: CD ID
            strTitle: title of CD
            strArtist: artist

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


    @staticmethod
    def remove_cd(table, intIDDel):
        """Function to remove a CD from the in-memory table

        Args:
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime
            intIDDel: user inputted ID to delete

        Returns:
            None.

        """
        intRowNr = -1
        blnCDRemoved = False
        for row in table:
            intRowNr += 1
            if row['ID'] == intIDDel:
                del table[intRowNr]
                blnCDRemoved = True
                break
        if blnCDRemoved:
            print('The CD was removed')
        else:
            print('Could not find this CD!')


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(file_name, '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(file_name, table):
        """Function to write data from the table to a file

        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
        """

            objFile = open(file_name, 'w')
            for row in table:
                lstValues = list(row.values())
                lstValues[0] = str(lstValues[0])
                objFile.write(','.join(lstValues) + '\n')
            objFile.close()


    # -- PRESENTATION (Input/Output) -- #

    class IO:
        """Handling Input / Output"""

        @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 string 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('======================================\n')

    @staticmethod
    def add_cd():
        """Allows user to add a CD

        Args:
            strID: user input for CD ID
            strTitle: CD title
            strArtist: artist name

        Returns:
            intID, strTitle, strArtist

        """
        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)
        return intID, strTitle, strArtist


# 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 from file
    if strChoice == 'l':
        print('WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.')
        strYesNo = input('Do you want to continue? [y/n] ')
        if strYesNo.lower() == 'y':
            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
        intID, strTitle, strArtist = IO.add_cd()

        # 3.3.2 Add item to the table
        DataProcessor.add_cd(intID, strTitle, strArtist)
```

```python
            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 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.remove_cd(lstTbl, intIDDel)
            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
                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.
        # 3.7 catch-all should not be possible, as user choice gets vetted in IO, but to be save:
        else:
            print('General Error')
```