Loren Ormerod
August 19, 2020
IT FDN 110 B
Assignment06

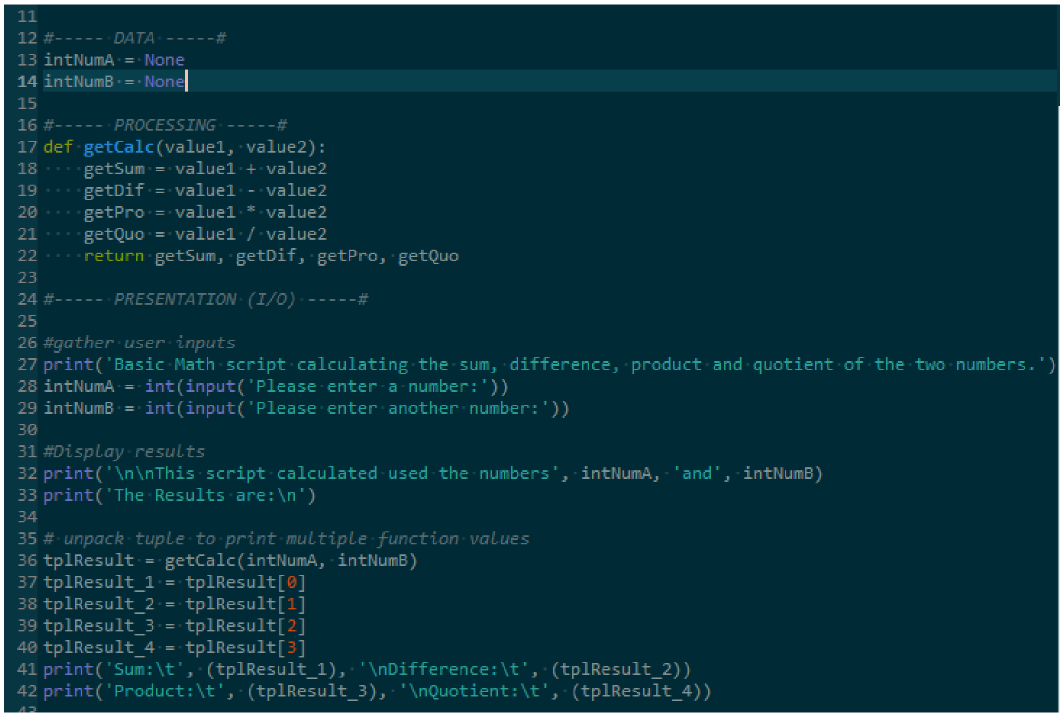
## Purpose:

Document actions taken and knowledge gained while working through Module06 course content.

## Weekly Content:

Functions are groups of statements that are accessible through one programmer defined name. In general, a function must be defined within the script before the function is available to be executed/return results. [1] Parameters, or arguments, are used to pass values into a function for processing. Variables may be passed to functions as arguments.

Return values are the results of variables or arguments being passed through a function. Thus far in the course we have worked with return values that are calculated upon each run of the script. It is possible, however, to save return values as a new variable so that you are doing the calculation once and storing the data rather than having to rerun the program to obtain the return value again. If a return statement has multiple values, Python language will automatically package the values as a tuple. See Figure 1 for an example of returning multiple values from a tuple.



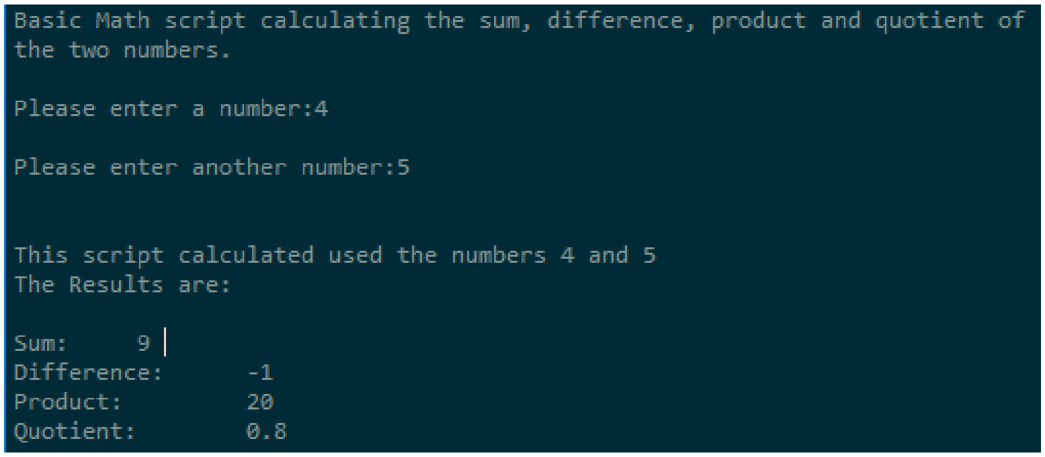
```python

#----- DATA -----#
intNumA = None
intNumB = None

#----- PROCESSING -----#
def getCalc(value1, value2):
    getSum = value1 + value2
    getDif = value1 - value2
    getPro = value1 * value2
    getQuo = value1 / value2
    return getSum, getDif, getPro, getQuo

#----- PRESENTATION (I/O) -----#

#gather user inputs
print('Basic Math script calculating the sum, difference, product and quotient of the two numbers.')
intNumA = int(input('Please enter a number:'))
intNumB = int(input('Please enter another number:'))

#Display results
print('\n\nThis script calculated used the numbers', intNumA, 'and', intNumB)
print('The Results are:\n')

# unpack tuple to print multiple function values
tplResult = getCalc(intNumA, intNumB)
tplResult_1 = tplResult[0]
tplResult_2 = tplResult[1]
tplResult_3 = tplResult[2]
tplResult_4 = tplResult[3]
print('Sum:\t', (tplResult_1), '\nDifference:\t', (tplResult_2))
print('Product:\t', (tplResult_3), '\nQuotient:\t', (tplResult_4))
```

*Figure 1 - Return multiple values from a tuple*

The script shown in Figure 1 works because we have grouped all of the calculations into one function, then requested multiple values to be returned, this automatically packages the return values as a tuple

[1] FDN_Py_Module_06.pdf, page 2

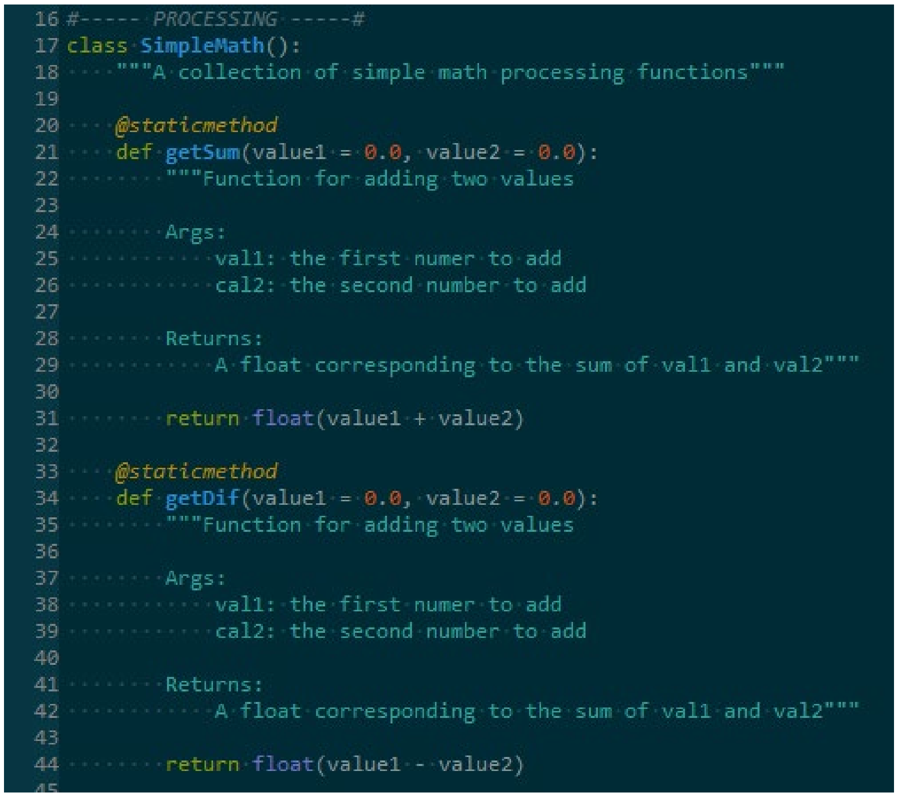
in Python 3. In order to print the tuple has been unpacked and fed as individual variables to the print statements. Results shown in Figure 2.



```
Basic Math script calculating the sum, difference, product and quotient of
the two numbers.

Please enter a number:4

Please enter another number:5


This script calculated used the numbers 4 and 5
The Results are:

Sum:     9
Difference:      -1
Product:         20
Quotient:        0.8
```

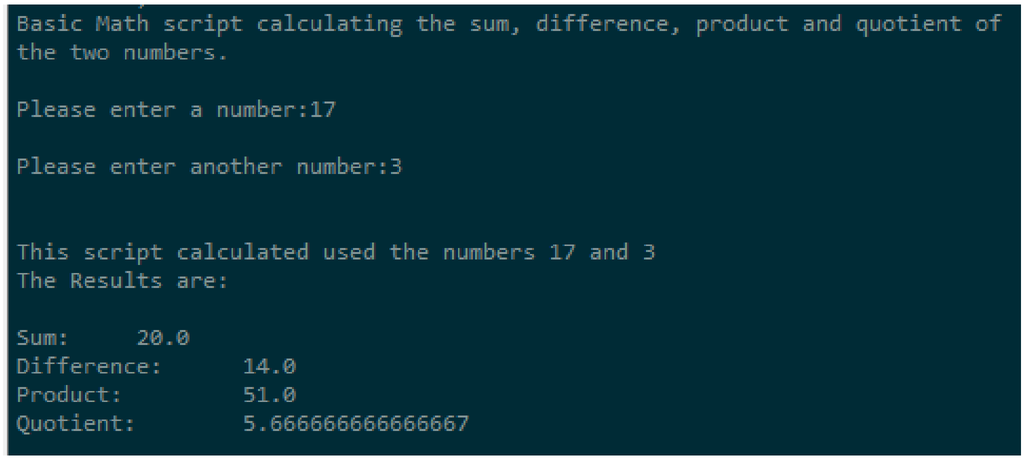
*Figure 2 - Results of multiple value return from tuple*

Functions can be organized into classes with accompanying docstring explanations. Classes are groupings of functions, variables or constants. This organization allows for readability of the program and makes it easier for programmers to read code they did not personally write. See Figures 3 and 4 for an example of a program organized using classes and docstring.



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

    @staticmethod
    def getSum(value1 = 0.0, value2 = 0.0):
        """Function for adding two values

        Args:
            val1: the first numer to add
            cal2: the second number to add

        Returns:
            A float corresponding to the sum of val1 and val2"""

        return float(value1 + value2)

    @staticmethod
    def getDif(value1 = 0.0, value2 = 0.0):
        """Function for adding two values

        Args:
            val1: the first numer to add
            cal2: the second number to add

        Returns:
            A float corresponding to the sum of val1 and val2"""

        return float(value1 - value2)
```

*Figure 3 - Functions organized into classes with docstrings*



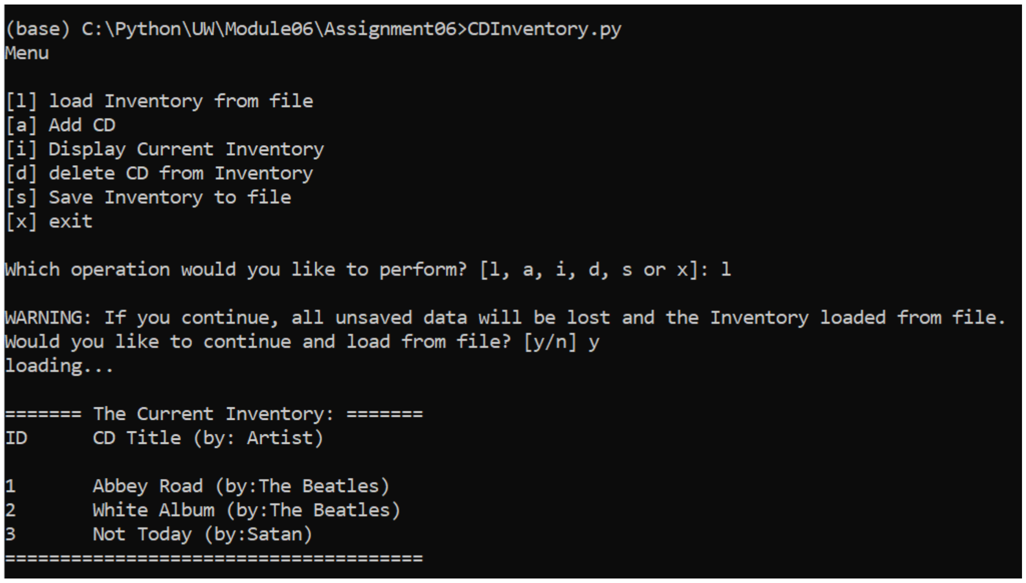
```
Basic Math script calculating the sum, difference, product and quotient of
the two numbers.

Please enter a number:17

Please enter another number:3


This script calculated used the numbers 17 and 3
The Results are:

Sum:     20.0
Difference:      14.0
Product:         51.0
Quotient:        5.666666666666667
```

*Figure 4 - Result of Functions organized into classes*

## Summary

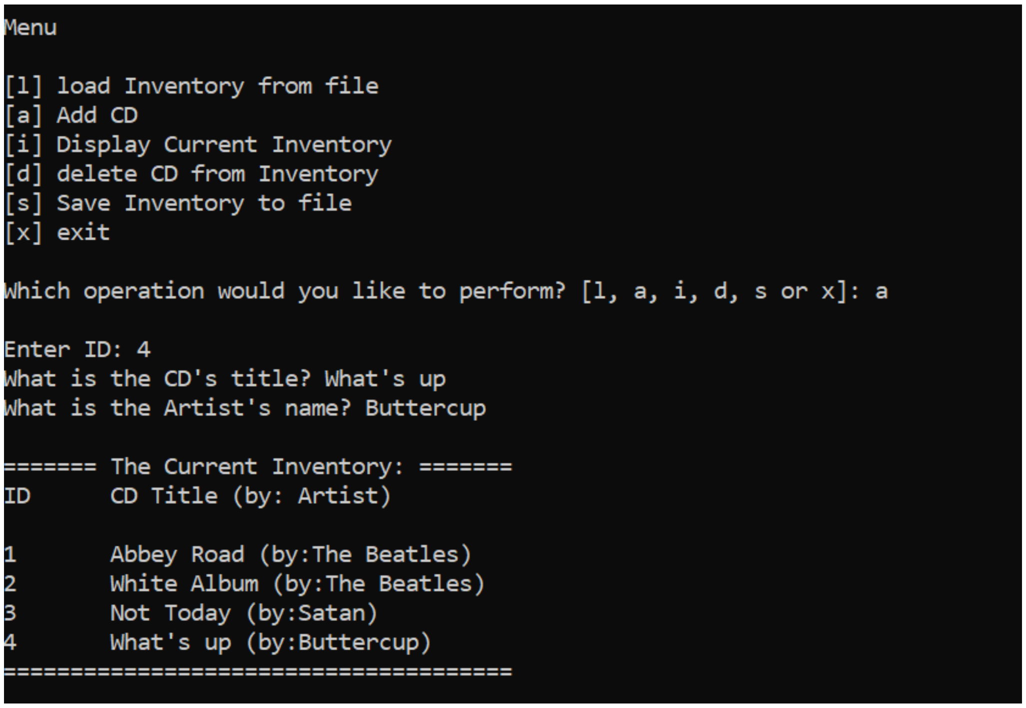
Organization of code and appropriate documentation is extremely important in programming. In Module 06 we covered docstring, organizing functions by class, and got hands on experience using SoC in a program. Adhering to standard programming language formats it is easier to pass scripts between users and achieve clarity of understanding between all involved parties. Appendix I holds screenshots of my executed code in Anaconda Prompt, the full script is available on GitHub.

## APPENDIX I



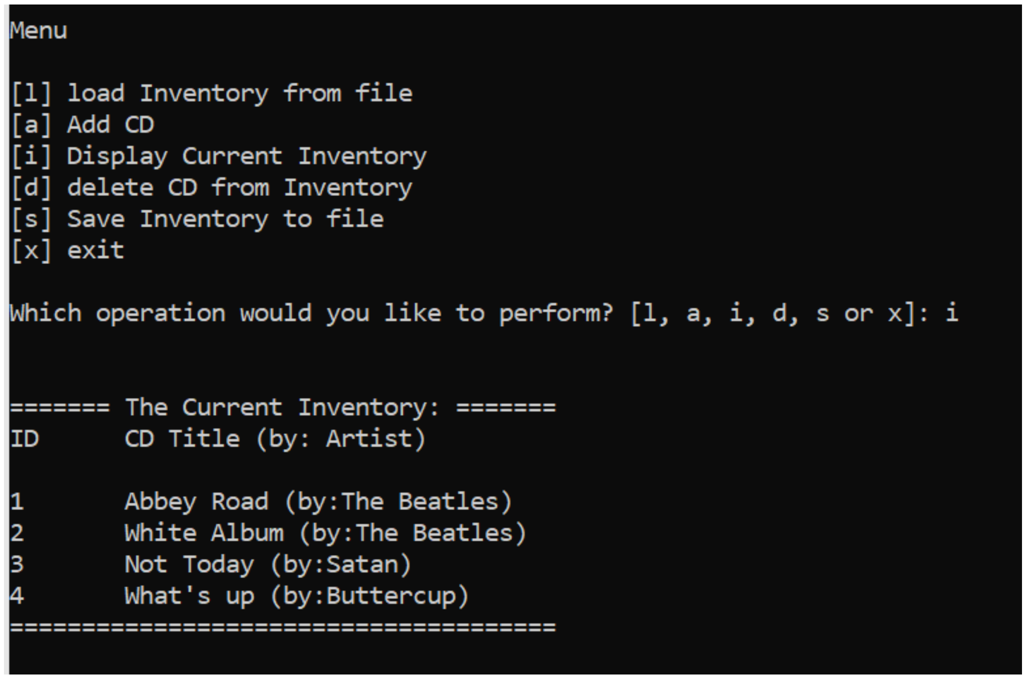
```
(base) C:\Python\UW\Module06\Assignment06>CDInventory.py
Menu

[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 loaded from file.
Would you like to continue and load from file? [y/n] y
loading...

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

1       Abbey Road (by:The Beatles)
2       White Album (by:The Beatles)
3       Not Today (by:Satan)
======================================
```

*Figure 5 - Execute Load Inventory from file*



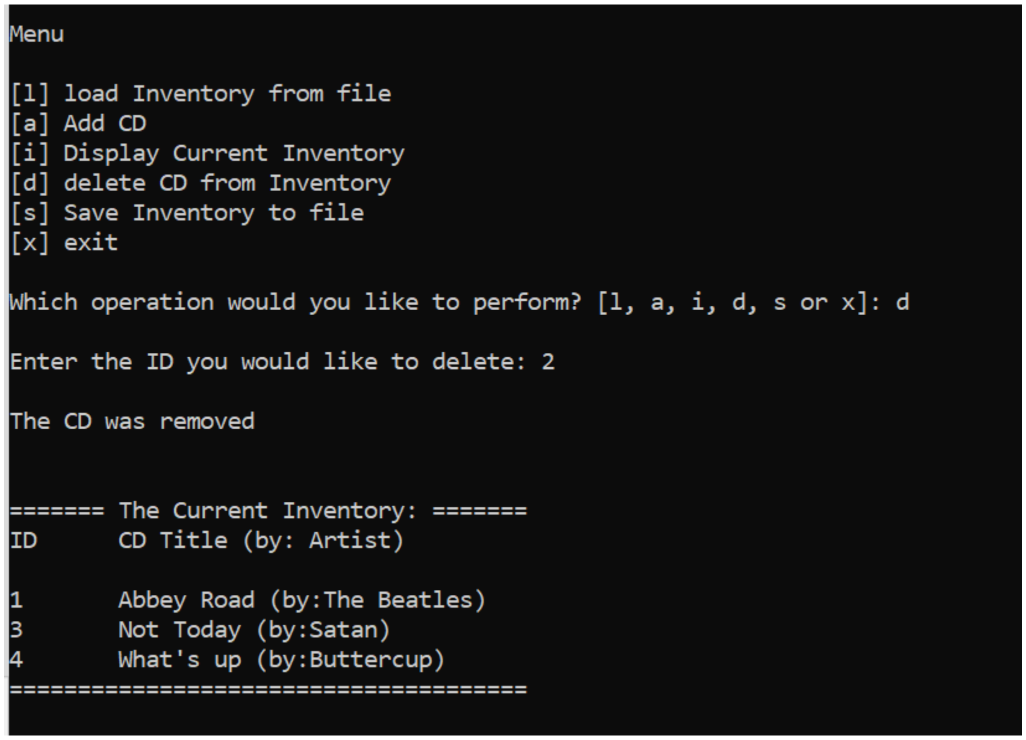
```
Menu

[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? What's up
What is the Artist's name? Buttercup

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

1       Abbey Road (by:The Beatles)
2       White Album (by:The Beatles)
3       Not Today (by:Satan)
4       What's up (by:Buttercup)
======================================
```

*Figure 6 - Execute Add CD*



```
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       Abbey Road (by:The Beatles)
2       White Album (by:The Beatles)
3       Not Today (by:Satan)
4       What's up (by:Buttercup)
======================================
```

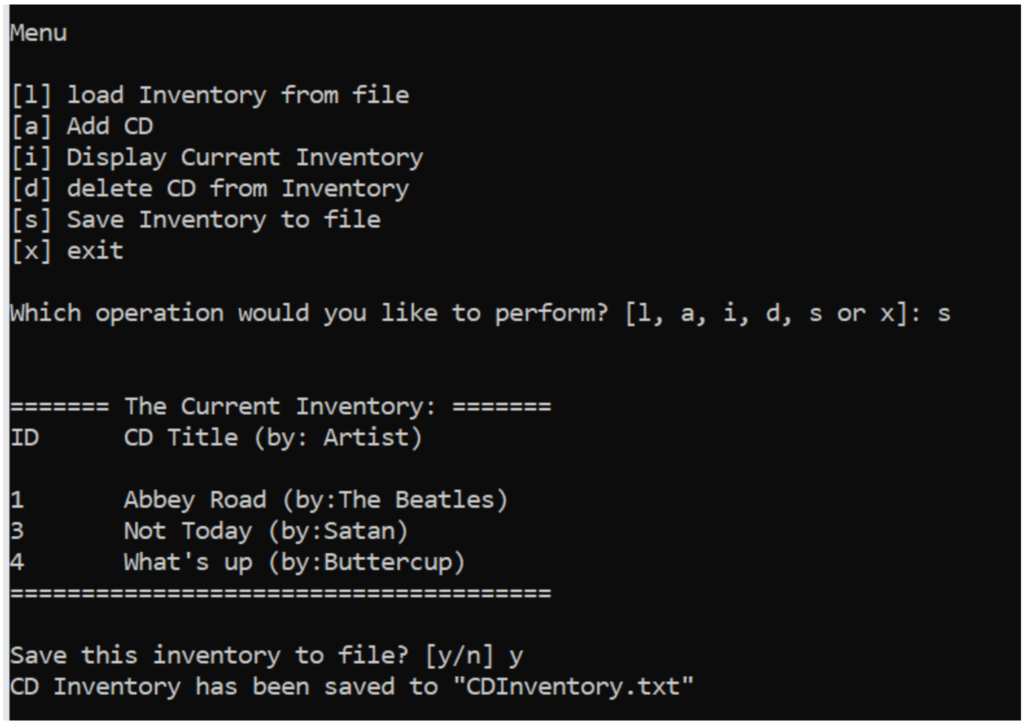
*Figure 7 - Execute Display Current Inventory*



```
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

Enter the ID you would like to delete: 2

The CD was removed


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

1       Abbey Road (by:The Beatles)
3       Not Today (by:Satan)
4       What's up (by:Buttercup)
======================================
```

48

49 *Figure 8 - Execute Delete CD from Inventory*

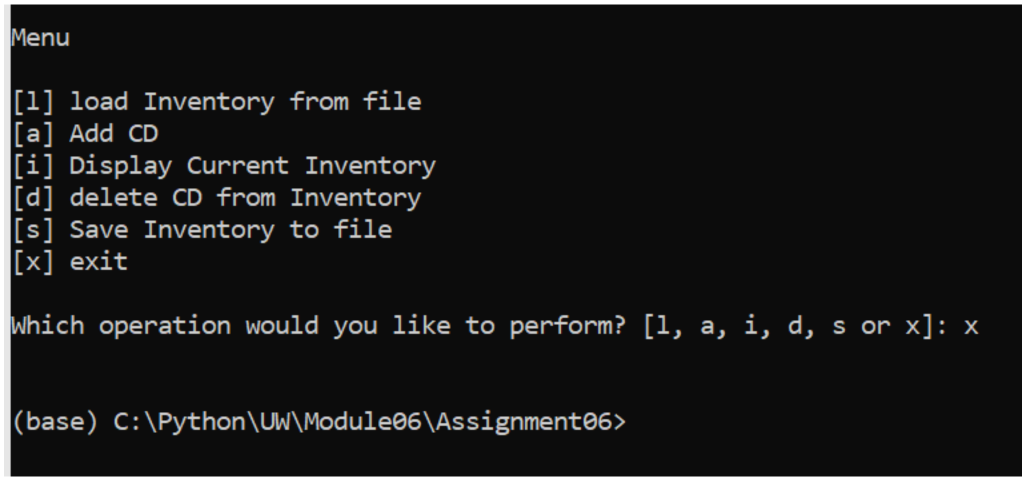


```
Menu

[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       Abbey Road (by:The Beatles)
3       Not Today (by:Satan)
4       What's up (by:Buttercup)
======================================

Save this inventory to file? [y/n] y
CD Inventory has been saved to "CDInventory.txt"
```

50

51 *Figure 9 - Execute Save Inventory to file*

52



```
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


(base) C:\Python\UW\Module06\Assignment06>
```

*Figure 10 - Execute exit*