Artem Aleshin
August 14th, 2022
IT FDN 110 A
Assignment 06

Assignment 06 Knowledge Document

Introduction

The Knowledge Document covers Points of Interest from Module 06, the work done as part of the sixth set of Laboratories and the completed Assignment. As part of working through the tasks, I learned about Functions, Classes, DocStrings and Variable Scope.

Module 06 Points of Interest

A topic that I found interesting from Module 06 is docstrings. Spyder does a great job of letting the user know what the information for the function is. I am also interested in classes and would like to know how the script knows when a class ends. Lastly, I think that the tic tac toe program in the book is interesting, albeit quite complicated. I would be interested in learning more about how one can improve the AI for the program.

Module 06 Laboratories

LAB 06_A

1. The script was duplicated.
2. The script was modified and is as follows:

```python
#-----------------------------------------#
# Title: Lab06_A.py
# Desc: Script demonstrating Functions concept, based on Basic_Math.py (Assignment02)
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# DBiesinger, 2030-Jan-01, Modified to demonstrate SoC
# DBiesinger, 2030-Jan-01, Modified to demonstrate Functions
# aaleshin, 2022-Aug-13, Modified to demonstrate Attributes and Return Values
#-----------------------------------------#

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

# -- PROCESSING -- #
# Process the data
def getSum(intNumA, intNumB):
    resultS = intNumA + intNumB
    return resultS

def getDif(intNumA, intNumB):
    resultD = intNumA - intNumB
    return resultD

def getPro(intNumA, intNumB):
    resultP = intNumA * intNumB
    return resultP

def getQuo(intNumA, intNumB):
    resultQ = intNumA / intNumB
    return resultQ

# -- PRESENTATION (Input/Output) -- #
# Get User input data
print('Basic Math script. Calculating the Sum, DIfference, Product and Quotient of two numbers.')
intNumA = int(input('Please enter the 1st Number: '))
intNumB = int(input('Please enter the 2nd number: '))
# Display the Results
print('\n\nThis script calculated using the Numbers', intNumA, 'and', intNumB)
print('The Results are:\n')
print('Sum:\t\t', getSum(intNumA, intNumB), '\nDifference:\t', getDif(intNumA, intNumB))
print('Product:\t', getPro(intNumA, intNumB), '\nQuotient:\t', getQuo(intNumA, intNumB))
```

Figure 1: Lab 06-A Script.

3. The script was tested.
    a. The output is as follows:

```
In [1]: runfile('C:/_FDProgramming/Lab_06/Lab06_A.py', wdir='C:/_FDProgramming/Lab_06')
Basic Math script. Calculating the Sum, DIfference, Product and Quotient of two numbers.
Please enter the 1st Number: 12
Please enter the 2nd number: 4


This script calculated using the Numbers 12 and 4
The Results are:

Sum:         16
Difference:  8
Product:     48
Quotient:    3.0
```

Figure 2: Lab 06-A Output.

b. The script works by first initializing two variables that will be used throughout the rest of the code. Four functions are then defined with each one returning a result based on the algebraic calculation that was performed. The user is then prompted to enter two values which then get assigned to the variables initialized in the beginning of the script. The two values then get plugged into the functions which return the results on separate lines along with accompanying text describing which mathematical calculation was performed.

LAB 06-B

1. A copy of LAB06-A was made.
2. The script was modified and is as follows:

```python
#-----------------------------------------#
# Title: Lab06_B.py
# Desc: Script demonstrating Functions concept, based on Basic_Math.py (Assignment02)
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# DBiesinger, 2030-Jan-01, Modified to demonstrate SoC
# DBiesinger, 2030-Jan-01, Modified to demonstrate Functions
# aaleshin, 2022-Aug-13, Modified to demonstrate Attributes and Return Values
# aaleshin, 2022-Aug-13, Modified to demonstrate Tuple packing
#-----------------------------------------#

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

# -- PROCESSING -- #
# Process the data
def doAlgebra(intNumA, intNumB):
    resultS = intNumA + intNumB
    resultD = intNumA - intNumB
    resultP = intNumA * intNumB
    resultQ = intNumA / intNumB
    return resultS, resultD, resultP, resultQ

# -- PRESENTATION (Input/Output) -- #
# Get User input data
print('Basic Math script. Calculating the Sum, DIfference, Product and Quotient of two numbers.')
intNumA = int(input('Please enter the 1st Number: '))
intNumB = int(input('Please enter the 2nd number: '))

# -- PROCESSING -- #
# Run function with inputs and assign to Tuple
doAlgebra(intNumA, intNumB)
answerS, answerD, answerP, answerQ = doAlgebra(intNumA, intNumB)

# Display the Results
print('\n\nThis script calculated using the Numbers', intNumA, 'and', intNumB)
print('The Results are:\n')
print('Sum:\t\t', answerS, '\nDifference:\t', answerD)
print('Product:\t', answerP, '\nQuotient:\t', answerQ)
```

Figure 3: Lab 06-B Script.

3. The script was tested,
   a. The output is as follows:

```
In [1]: runfile('C:/_FDProgramming/Lab_06/Lab06_B.py', wdir='C:/_FDProgramming/Lab_06')
Basic Math script. Calculating the Sum, DIfference, Product and Quotient of two numbers.
Please enter the 1st Number: 12
Please enter the 2nd number: 4


This script calculated using the Numbers 12 and 4
The Results are:

Sum:         16
Difference:  8
Product:     48
Quotient:    3.0
```

Figure 4: Lab 06-B Output.

b. The script works by first initializing two variables that will be used throughout the rest of the code. One function is then defined to perform four algebraic calculations. The function returns four results in a single line corresponding to each algebraic operation. The user is then prompted to enter two values which then get assigned to the variables initialized in the beginning of the script. The two values then get plugged into the function and the four results get assigned to a Tuple. The script then returns each one of the values in the Tuple along with accompanying text describing which mathematical calculation was performed.

LAB 06-C

1. A copy of LAB06-A was made.
2. The script was modified per the instructions and is as follows:

```python
#-----------------------------------------#
# Title: Lab06_C.py
# Desc: Script demonstrating Functions concept, based on Basic_Math.py (Assignment02)
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# DBiesinger, 2030-Jan-01, Modified to demonstrate SoC
# DBiesinger, 2030-Jan-01, Modified to demonstrate Functions
# aaleshin, 2022-Aug-13, Modified to demonstrate Attributes and Return Values
# aaleshin, 2022-Aug-13, Modified to demonstrate Classes and Docstrings
#-----------------------------------------#

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

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

    @staticmethod
    def add_values(val1 = 0.0, val2 = 0.0):
        """Function for adding two values

        Args:
            val1: the first number to add
            val2: the second number to add


        Returns:
            A float corresponding to the sum of val1 and val2
        """
        return float(val1 + val2)

    @staticmethod
    def subtract_values(val1 = 0.0, val2 = 0.0):
        """Function for subtracting two values

        Args:
            val1: the number to subtract from
            val2: the number to subtract


        Returns:
            A float corresponding to the difference of val1 and val2
        """
        return float(val1 - val2)

    @staticmethod
    def multiply_values(val1 = 0.0, val2 = 0.0):
        """Function for multiplying two values

        Args:
            val1: the first number to multiply
            val2: the second number to multiply


        Returns:
            A float corresponding to the product of val1 and val2
        """
        return float(val1 * val2)

    @staticmethod
    def divide_values(val1 = 0.0, val2 = 0.0):
        """Function for dividing two values

        Args:
            val1: the number to divide
            val2: the number to divide by


        Returns:
            A float corresponding to the quotient of val1 and val2
        """
        return float(val1 / val2)

# -- PRESENTATION (Input/Output) -- #
# Get User input data
print('Basic Math script. Calculating the Sum, DIfference, Product and Quotient of two numbers.')
intNumA = int(input('\nPlease enter the 1st Number: '))
intNumB = int(input('\nPlease enter the 2nd number: '))
# Display the Results
print('\n\nThis script calculated using the Numbers', intNumA, 'and', intNumB)
print('\nThe Results are:\n')
print('Sum:\t\t', SimpleMath.add_values(intNumA, intNumB), '\nDifference:\t', SimpleMath.subtract_values(intNumA, intNumB))
print('Product:\t', SimpleMath.multiply_values(intNumA, intNumB), '\nQuotient:\t', SimpleMath.divide_values(intNumA, intNumB))
```

Figure 5: Lab 06-C Script.

3. The script was tested,
    a. The output is as follows:

```
In [1]: runfile('C:/_FDProgramming/Lab_06/Lab06_C.py', wdir='C:/_FDProgramming/Lab_06')
Basic Math script. Calculating the Sum, DIfference, Product and Quotient of two numbers.

Please enter the 1st Number: 12

Please enter the 2nd number: 4


This script calculated using the Numbers 12 and 4

The Results are:

Sum:         16.0
Difference:  8.0
Product:     48.0
Quotient:    3.0
```

Figure 6: Lab 06-C Output.

    b. The script works by first initializing two variables that will be used throughout the rest of the code. A class SimpleMath() is then created containing four functions with each function corresponding to one of four algebraic calculations. The class and each of the four functions contain docstrings that add descriptions about what the functions do. The functions all return a float. The user is then prompted to enter two values which then get assigned to the variables initialized in the beginning of the script. The two values then get plugged into each of the four functions and the four results get returned along with accompanying text describing which mathematical calculation was performed.

Assignment 06

1. Folder called Assignment06 was created.
2. The code was modified and is as follows:

```python
#------------------------------------------#
# Title: Assignment06_Starter.py
# Desc: Working with classes and functions.
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# aaleshin, 2022-Aug-13, Modified File to move code into functions under classes
# aaleshin, 2022-Aug-14, Modified File to include docstrings
#------------------------------------------#

# -- 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

file = open(strFileName, 'a+')  # Creates CDInventory.txt if it does not exit.
file.close()

# -- PROCESSING -- #
class DataProcessor:
    # TODone add functions for processing here
    """Manipulating the data in memory"""

    @staticmethod
    def delete_CD(lstTbl):
        """Deletes CD from in memory table

        Args:
            lstTbl (list of dict): In memory 2D data structure (list of dicts) that holds the data during runtime

        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!')

    @staticmethod
    def add_CD(lstTbl, strID, strTitle, strArtist):
        """Adds CD information to in memory table

        Args:
            lstTbl (list of dict): 2D data structure (list of dicts) that holds the data during runtime
            strID (String): String containing CD ID.
            strTitle (String): String containing CD Title.
            strArtist (String): String containing CD Artist.

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

```

```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(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, lstTbl):
        # TODone Add code here
        """Function to manage writing data from table to file

        Writes data from 2D table (list of dicts) to file one row at a time.

        Args:
            file_name (string): name of file used to read the data from
            lstTbl (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()

# -- 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')
```

```python

    @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('======================================')

    # TODone add I/O functions as needed


    @staticmethod
    def get_ID():
        """Gets user input for CD ID.

        Args:
            None.

        Returns:
            Returns the CD ID as a string.
        """
        strID = input('Enter ID: ').strip()
        return strID

    @staticmethod
    def get_Title():
        """Gets user input for CD Title.

        Args:
            None.

        Returns:
            Returns the CD Title as a string.
        """
        strTitle = input('What is the CD\'s title? ').strip()
        return strTitle
```

```python

    @staticmethod
    def get_Artist():
        """Gets user input for CD Artist.

        Args:
            None.

        Returns:
            Returns the CD Artist as a string.
        """
        strArtist = input('What is the Artist\'s name? ').strip()
        return 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
    if 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
        strID = IO.get_ID()
        strTitle = IO.get_Title()
        strArtist = IO.get_Artist()
        # 3.3.2 Add item to the table
        # ToDone move processing code into function
        DataProcessor.add_CD(lstTbl, strID, strTitle, strArtist)
        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
        # TODone move processing code into function
        DataProcessor.delete_CD(lstTbl)
        IO.show_inventory(lstTbl)
        continue  # start loop back at top.
```

```python
    # 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
            # TODone move processing code into function
            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')
```

Figure 7: Assignment 06 Script

3. The script was tested:
    a. In Spyder:

```
In [1]: runfile('C:/_FDProgramming/Assignment06/CDInventory.py', wdir='C:/_FDProgramming/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 canceledyes
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

======================================
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: 1
What is the CD's title? Ten
What is the Artist's name? Pearl Jam
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Ten (by:Pearl Jam)
======================================
```

```
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: 2
What is the CD's title? Nevermind
What is the Artist's name? Nirvana
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Ten (by:Pearl Jam)
2   Nevermind (by:Nirvana)
======================================
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   Ten (by:Pearl Jam)
2   Nevermind (by:Nirvana)
======================================
Menu
```

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

1   Ten (by:Pearl Jam)
2   Nevermind (by:Nirvana)
======================================
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   Ten (by:Pearl Jam)
2   Nevermind (by:Nirvana)
======================================
Which ID would you like to delete? 1
The CD was removed
======= The Current Inventory: =======
ID  CD Title (by: Artist)

2   Nevermind (by:Nirvana)
======================================
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   Nevermind (by:Nirvana)
======================================
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   Nevermind (by:Nirvana)
======================================
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]: x
```

Figure 8: Assignment 06 Spyder Output.

b. In Anaconda Prompt:

```
(base) C:\_FDProgramming\Assignment06>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)

2       Nevermind (by:Nirvana)
======================================
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: 1
What is the CD's title? Ten
What is the Artist's name? Pearl Jam
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       Nevermind (by:Nirvana)
1       Ten (by:Pearl Jam)
======================================
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       Nevermind (by:Nirvana)
1       Ten (by:Pearl Jam)
======================================
Which ID would you like to delete? 2
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Ten (by:Pearl Jam)
======================================
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       Ten (by:Pearl Jam)
======================================
Save this inventory to file? [y/n] y
Menu

[l] load Inventory from file
[a] Add CD
```

Figure 9: Assignment 06 Anaconda Prompt Output

4. The final inventory after running the script in Spyder and Anaconda Prompt is as follows:

```
1,Ten,Pearl Jam
```

Figure 10: Assignment 06 Resulting CDInventory.txt

In order to perform the assignment, I first read through the code and took note that there are three different classes. I then started with the code backwards, first moving the portion of the script that saves the 2D Table into the write_file function in the FileProcessor class. Next, I moved the script that deletes a row from the 2D Table based on ID to its own function in the DataProcessor class. I then created 3 functions under the IO class that request the user to input the ID, Title and Artist of the CD, respectively and return the strings containing the information. I then moved the portion of the script that adds a new CD to its own function under the DataProcessor class. For each function that was created, I made sure to properly call it in the sections where the script for the function was moved from. Lastly, I wrote the docstrings for each of the functions and confirmed that the code works both in Spyder and Anaconda Prompt.

Summary

The document covers the Points of Interest, the Laboratories, and the Assignment from Module 06. By completing Module 06, I learned about Functions, Classes, DocStrings and Variable Scope.