*Yong Son*

*Dec 05, 2020*

*IT FDN 110 A Au20: Foundations of Programming: Python*

*Assignment 08*

gitHub link: <https://github.com/yms7/ITFnd100-Mod08>

**Assignment 08: Objects and Classes**

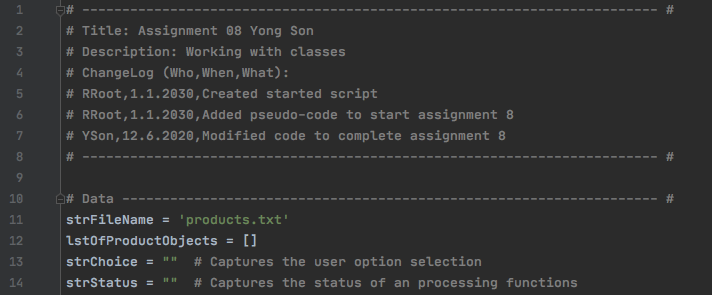
Introduction

In week 06, we were introduced to the concept called Classes and this week we review more in-depth characteristics of classes. Just as other programming patterns, there’s the general design pattern to construct the coding structure inside the Class. Our main goal for this week is to learn how to manage these coding patterns inside of Class structure. Another main topic we learned this week was to use Objects to create multiple instances of the Class with different address in memory.

There are five main design patterns programmers use inside of the Class:

Field, constructor/destructor, Attributes, Properties and Methods. I will try to explain each field as we go through my scripts for Assignment 08.

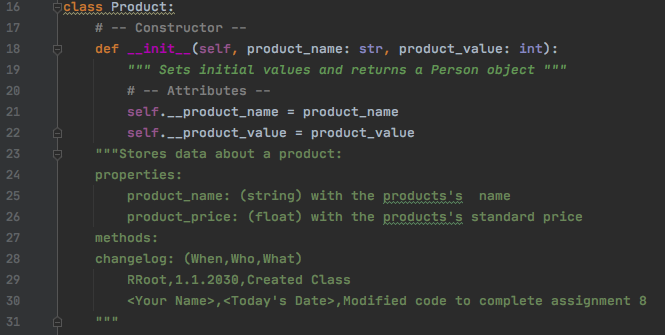
Just as I always started, I create header and data set for Assignment 08. Basic Data entry that I set were “trFilename” for text file I will be saving into the hard drive, “lstOfProductObjects” for list that carries all the Product Objects, “strChoice” for user’s choice input, and “strStatus” for status return messages. (Figure 8-1)



***Figure 8-1: Classes***

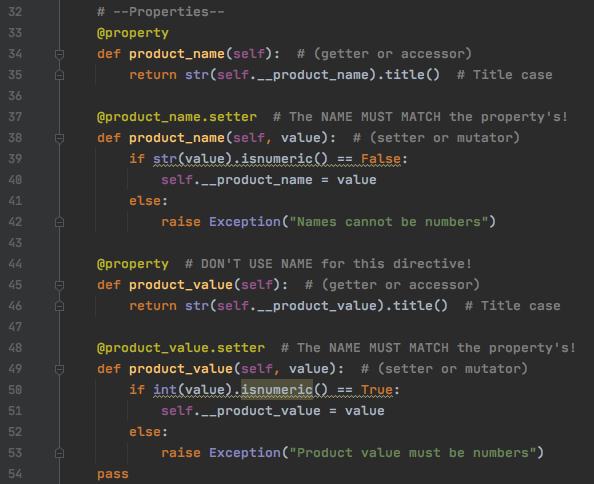
Objects (collection of data in Class) are very useful when multiple data entry needs to be created using Class. In order to manage multiple product information, I create Class called “Product”. “Product” is created to manage multiple product data entries including product name and its value.

On this Product Class, I followed general design pattern and create a function call \_\_init\_\_. “\_\_init\_\_” is a reserved method in python classes. Function \_\_init\_\_ initialize the attributes of the class by assigning data members of the class. On the attributes section, product names and product values were initiated using “\_\_init\_\_” function. Attributes are “virtual” fields that hold internal data (Figure 8-2). Product name and product value will be used indirectly; therefore, both attributes are renamed to be “private” per \_\_Mod8PythonProgrammingNotes.



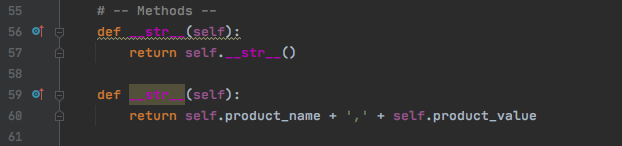
***Figure 8-2: Class “Product”***

Properties are functions used to manage field or attribute data. Programmers typically create two properties for each field/attributes, one for “getting” data and one for “setting data” (Properties, “\_Mod8PythonProgrammingNotes.pdf”, 2020).



***Figure 8-3: Class “Property”***

\_\_str\_\_ methods only returns the name of class and an address identifier. However, this method also can return the contents of the class’s attributes. Function to return class’s attribute is made to return the attribute value of product name and product value (Figure 8-4).



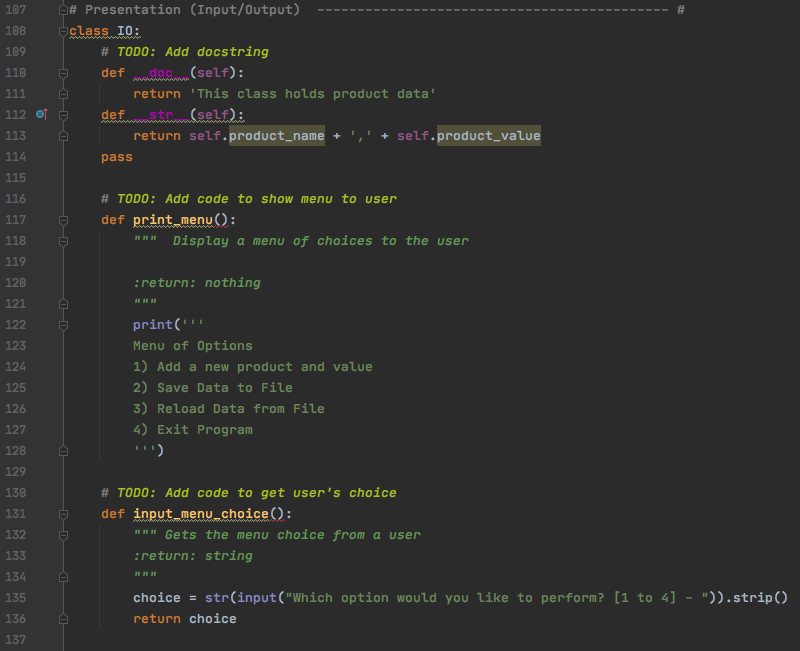
***Figure 8-4: Class “Method”***

Another Class I created is “FileProcessing”. These codes are exactly same as assignment 06. There are one function to write date to file and another function to read data from the file. Also there’s another function to append data into list of data (Figure 8-5).



***Figure 8-5: Class “Property”***

Input and Output Class is also similar to Assignment 06. I only add \_\_doc\_\_ and \_\_str\_\_ function but not sure if those two functions are actually needed for I/O class. All other scripts are similar to Assignment 06. There is function to display menu choices, function to take user’s input choices, function to display the information on the list of table, and other functions to take user’s choices such as yes or no or continue (Figure 8-6 & 8-7).



***Figure 8-6: Class “IO”***

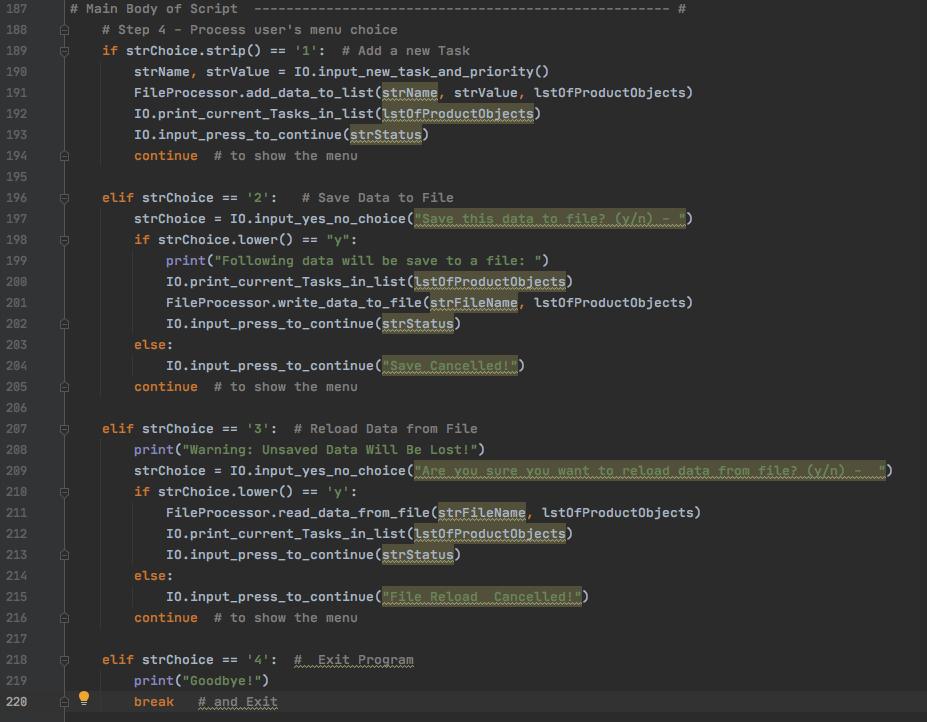


***Figure 8-7: Class “IO”***

Main Scripts are just a way to bring all functions from different classes to display, process and present all the information based on user’s input. Main script is also similar to Assignment 06 (Figure 8-8).

If user chooses option menu “1”, function from IO class will ask for a user input for different product name and value. Then another function from FileProcessor class will add these inputs into the list. Print function from IO class will display all data saved in the list and then script will ask user’s input for continue the program.

Other elif statements will function similarly by bring in functions from different class to perform the task (Figure 8-8).



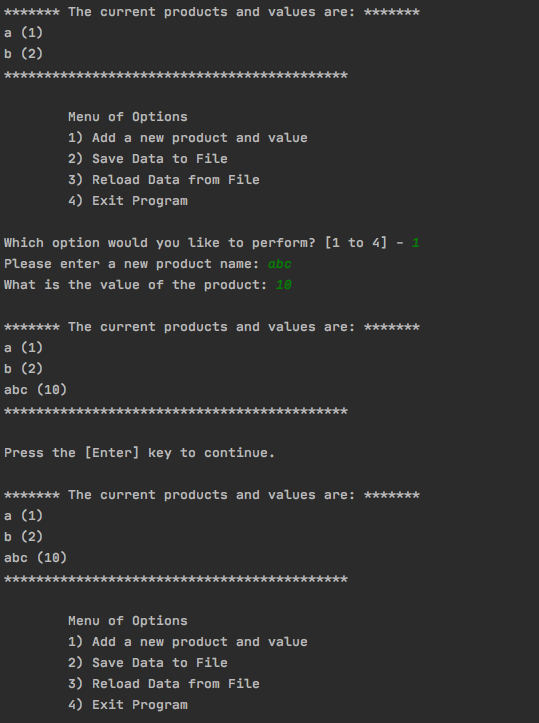
***Figure 8-8: Main***

Conclusion

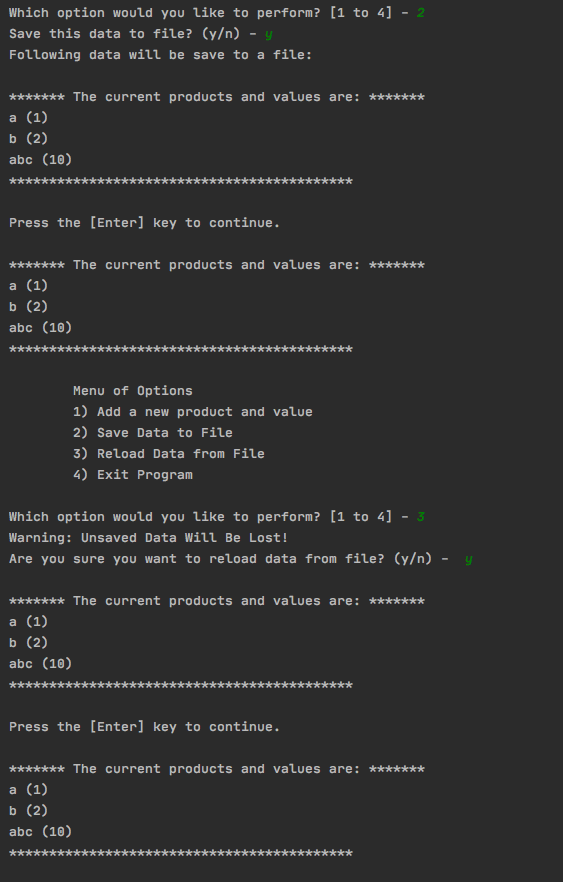
In week 08, we learn how to use Class to work with multiple data inputs. When processing multiple similar data entries, it is important to use object in classes to storing data. Also scripts in Classes should follow general design patterns: Field, constructor/destructor, Attributes, Properties and Methods.

Generally most of the coding patterns were similar to Assignment 06. However, on assignment 08, we learned advance concept call objects in class and this is very useful method to processing massive data entries. See the result of my program on (Figure 8-9 & 8-10 & 8-11 & 8-12 & 8-12).

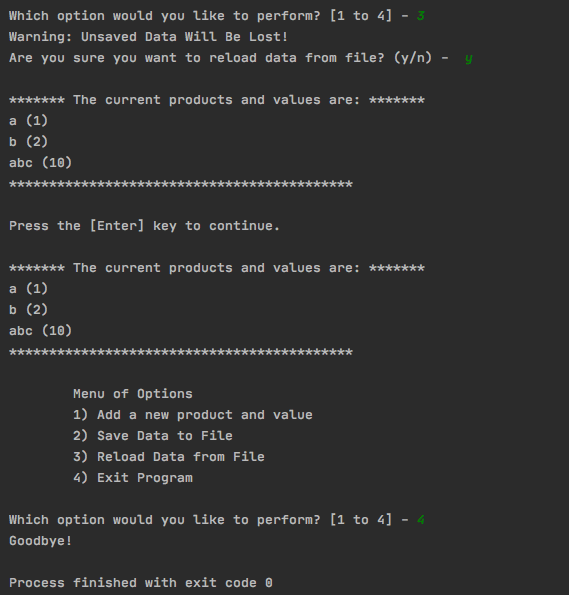
Result: (Figure 8-9 & 8-10 & 8-11 & 8-12 & 8-12)



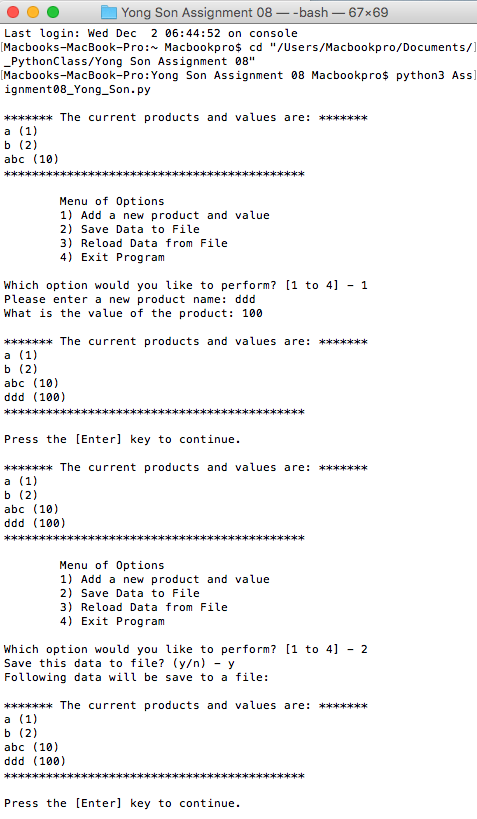
***Figure 8-9: Result***



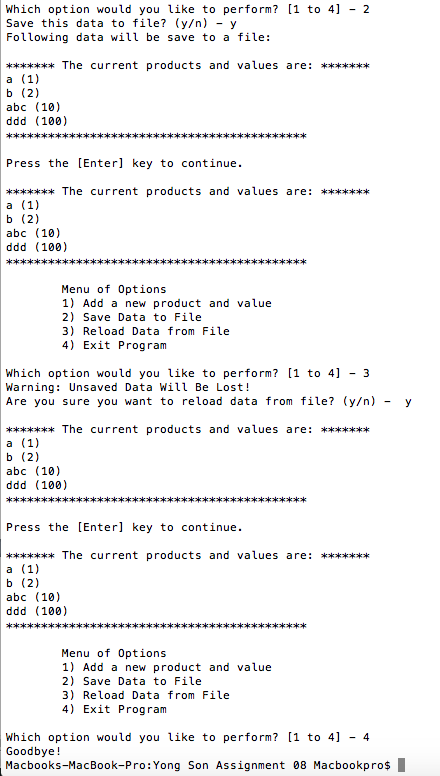
***Figure 8-10: Result***



***Figure 8-11: Result***



***Figure 8-12: Result***



***Figure 8-13: Result***