Name: Charles Shin

Date: 5/31/2021

Course: IT FDN 110 A Sp21: Foundations of Programming: Python

Assignment 08

GitHub URL:

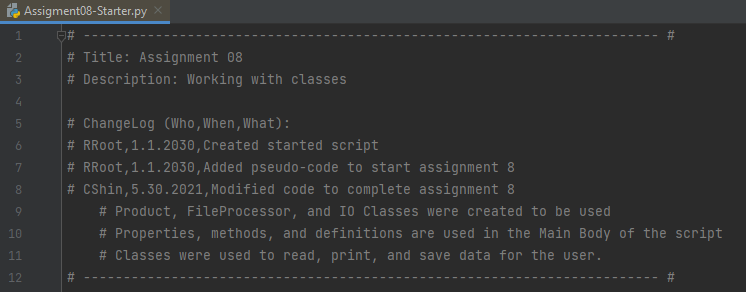
# Assignment 08: Classes & Objects

## Introduction

In this assignment, Classes are described. This script will be used for a user to see the current data (name and price of items) and save the new data. A new project was created for this assignment under the sub-folder called Assignment08. To use pre-made script *Assignment08-Starter.py*, the file was copied to the folder created. In the script, the following three classes were used to define properties and methods in each of the classes: *Product, FileProcessor, and IO*. Each class and how the user interface was built will be discussed in the following section of this assignment document.

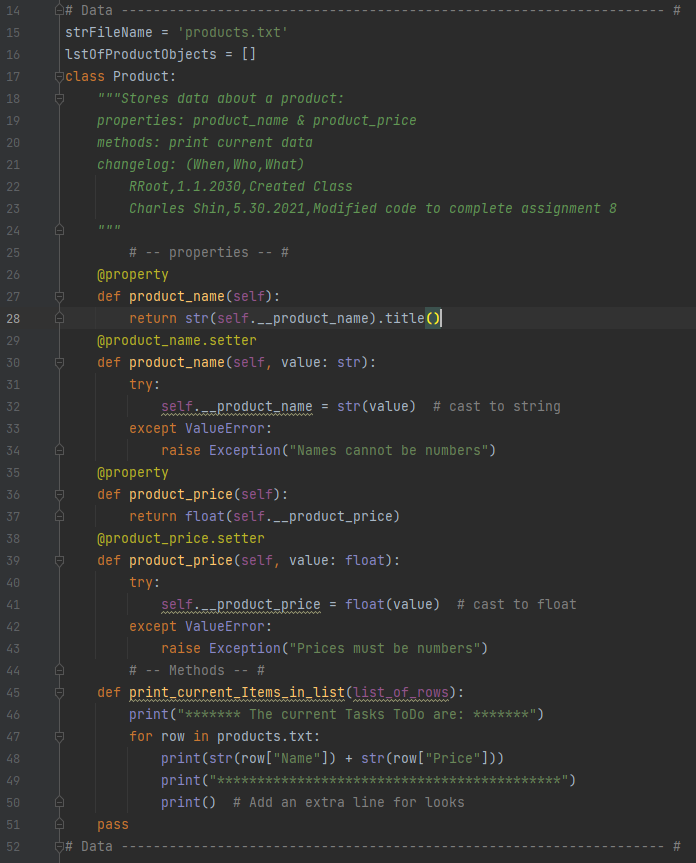
## Discussion

Like all other assignments, the first thing that was done on this script file was to create a header. Figure 1 shows the header of the script. It shows the change log as well as briefly describing what had been added/changed.



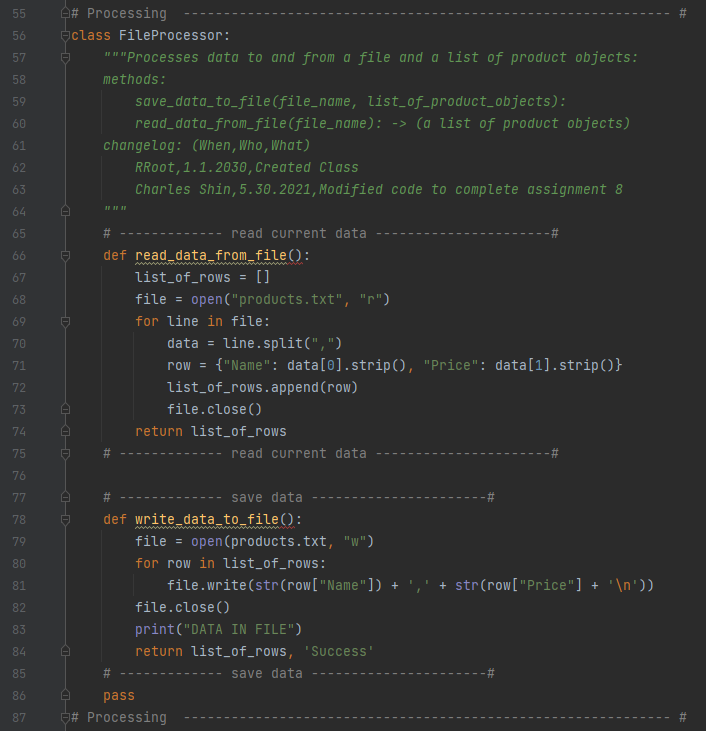
***Figure 1. Header of the script***

Figure 2 shows how the data was handled in the script. First, the string file name *products.txt* was defined. Then, the *Product* class was defined. As shown in the figure, the product name and product price properties were defined in the class. Note *try-except* was used, as it was suggested by the instructor. Furthermore, the definition method of *print\_current\_Items\_in\_list* was defined. This method was defined to be used in case the user wanted to see the current items in the data file.



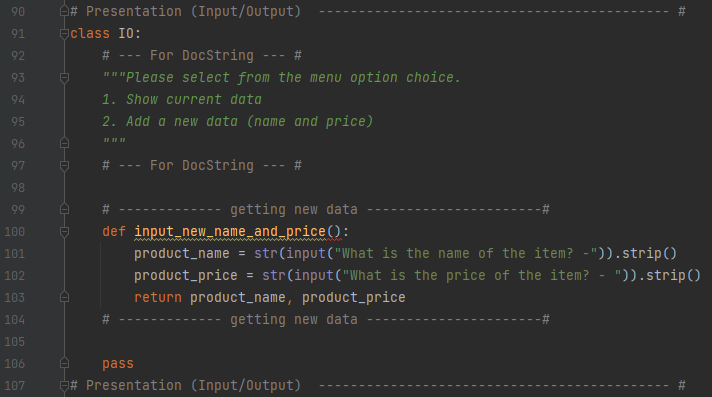
***Figure 2. Product Class definition***

Once the *Product* class was defined, a new class called *FileProcessor* was written.Figure 3 shows the *FileProcessor* class script. As shown in the figure, the class starts with the header to describe the purpose of the class. This class is basically used for processing the data. Thus, the class contains two definitions: *read\_data\_from\_file* & *write\_data\_to\_file.* Once they were defined within the class, the script moved onto the next class.



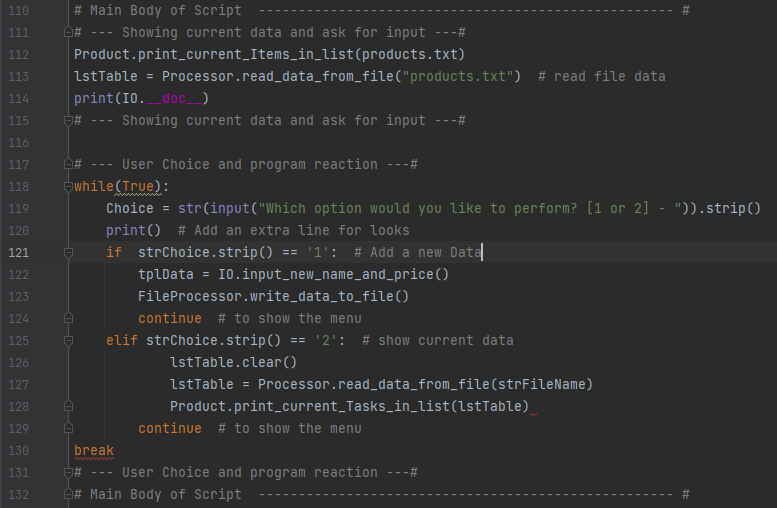
***Figure 3. FileProcessor Class definition***

The *IO* class was created to basically communicate with the user. Figure 4 shows the *IO* class script. The first thing that was done on this class was to add the strings of comments that will be used as a doc string when showing the main menu to the user. As shown in the figure, *Show current data* and *Add a new data (name and price)* were added as string comments in order to be used as doc strings. Continuing in the *IO* class, the definition of *input\_new\_name\_and\_price* was defined in order to get the new data from the user.



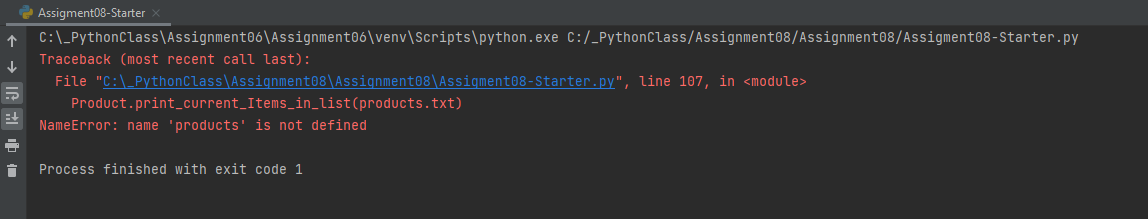
***Figure 4. IO Class definition***

Once all the classes were defined, the main body of the script was constructed. Figure 5 shows the main body constructed to communicate with the user. As shown in the script, the main body uses all the pre-defined methods from the three classes created previously in the script. First, it processes and print the current data in the data file. Then, it prints the doc string to ask the user what the user would like to do next. The rest of the script used rest of the defined methods in classes to react to depending on what the user chose. As shown in the doc string description, there were two choices that the user could choose, and those were to either add a new data or show the current data in the data file.



***Figure 5. Main Body of the script***

Once the script construction was done, it was tried to be run. Unfortunately, the script came out with several errors. It was tried to be fixed but failed to be run. Figure 6 shows the last error message received by trying to run the script.



***Figure 6. Result of running the script***

## Summary

In this assignment, how to use the Class was learned and described. It was learned that there are several components (fields, constructor, attributes, properties, and methods) that makes up the class. Also, it was learned that classes can be a great tool to distinctively separate and define methods (functions) within the script. Although the script constructed in this assignment failed to be run, the importance of using the classes within the script was learned.