Hannah Chung

June 6, 2020

IT FDN 100 Foundations of Programming: Python

Assignment 08

<https://github.com/hannahinissaquah/IntroToProg-Python-Mod08>

# **Assignment08: Objects and Classes**

Introduction

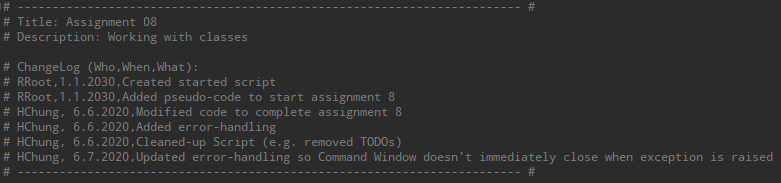
This week we learned about creating scripts using custom classes and were introduced to the basics of object-oriented programming (OOP). Classes are a way of organizing data and functions. When the class’s code is loaded into memory, it can either be used directly or indirectly by creating an object instance of the class.

For this week’s homework assignment, we were provided with starter code in a file called “Assigment08-Starter.py”, and our task was to manage a list of products and their prices using what we learned about objects and classes and adding code to make the application work.

Script

**Getting started**

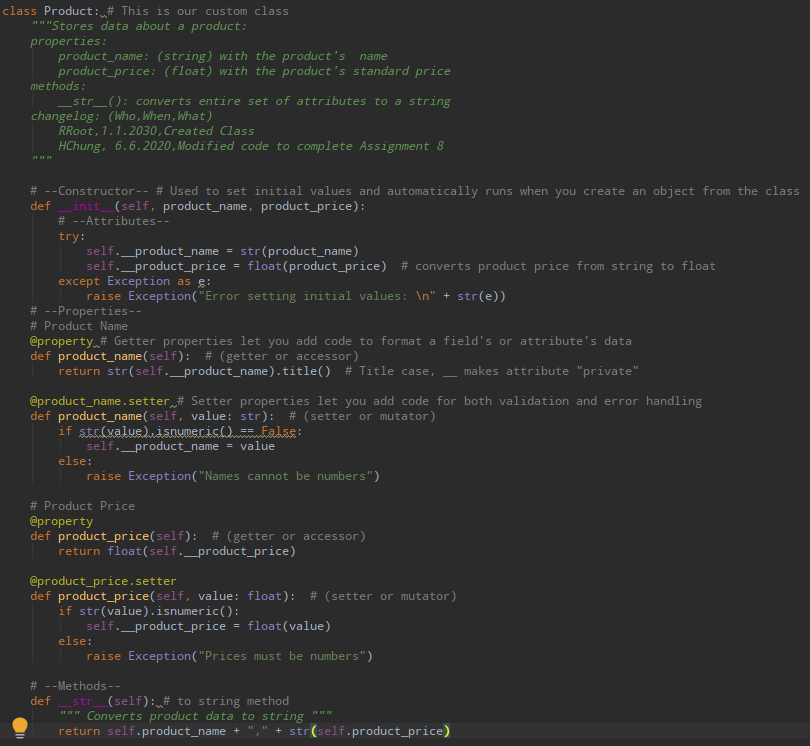
First, I created a new PyCharm project that uses the \_PythonClass\Assignment08 folder as its location. I added the Assigment08\_Starter.py file to my project and loaded it into PyCharm. I then saved a copy of it and named the new file Assignment08.py. I made my first entry into the change log as shown in Figure 1.



**Figure 1 Screenshot of Script Header for Assignment08.py**

**Data**

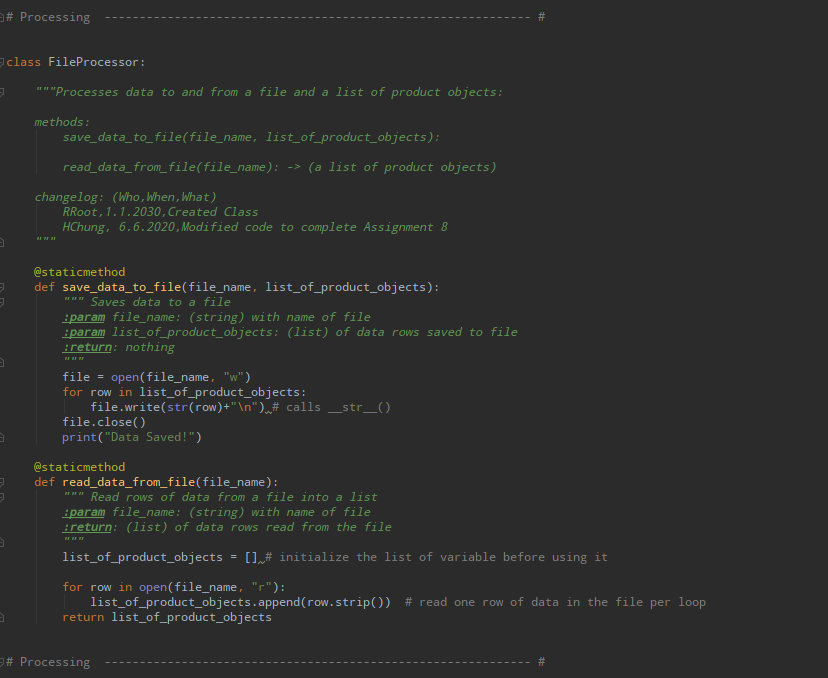
The program is separated into four main sections: Data, Processing, Input/Output (or Presentation), and the Main Body of the Script. The Data section contains two variables as well as a custom class named “Product”, which is used to store data and the methods to process the data. The starter code for the Product class contained a doc string that provided clues to what needed to be added. Using the information provided within the doc string along with the Lecture Notes, I added a Constructor (“\_\_init\_\_”), Attributes, Properties, and Methods to complete the code for this custom class. Figure 2 shows a screenshot of my final code for the Product class. I tested the code using the code shown in Figure 6 before moving onto the Processing layer.

******

**Figure 2 Screenshot of Data Layer with Product class**

**Processing**

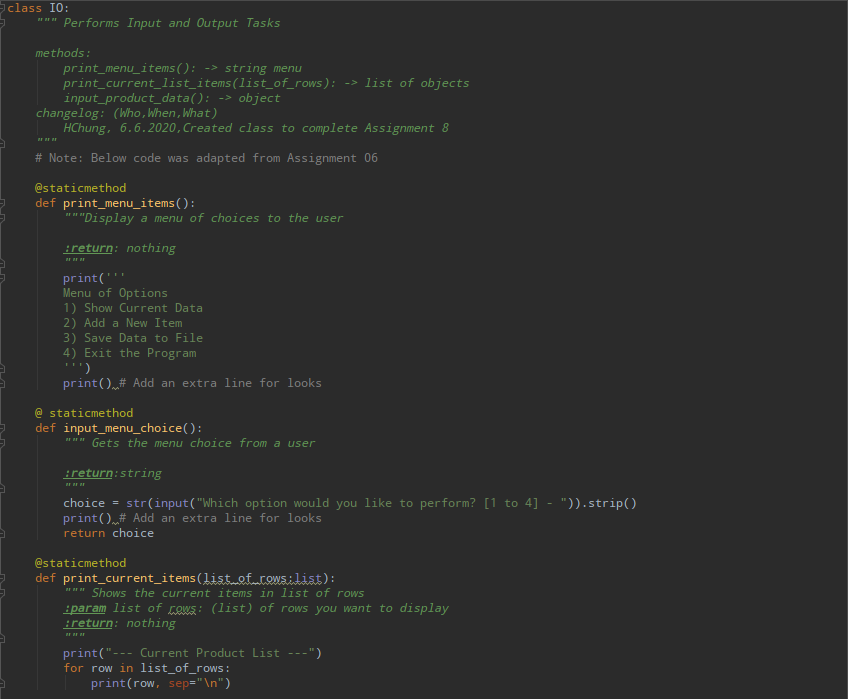
The Processing layer is made up of one class called “FileProcessor”, which processes data to and from a file and a list of product objects. It includes two methods: save\_data\_to\_file and read\_data\_to\_file. The data that is pulled from the file is put into a list of product objects rather than a list of dictionaries. My task was to complete the code to make the two methods work. Figure 3 shows a screenshot of the completed Processing section. Again, I tested the code using the code shown in Figure 6 before moving onto the Presentation Layer.

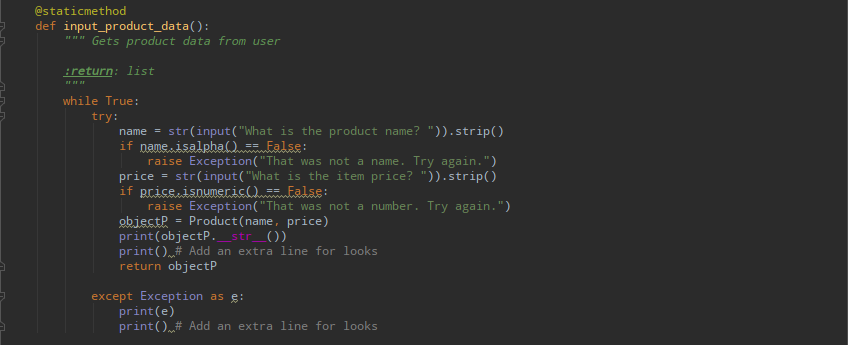


**Figure 3 Screenshot of Processing Layer**

**Presentation**

The Presentation Section is the layer that performs input and output tasks. A class named “IO” was created and my task was to add code to show menu to user, to get the user’s choice, to show the current data from the file to user, and to get product data from the user. Much of the code that was used for this section was adapted from Assignment06. One difference is that an object instance of the Product custom class is created using the values for product name and product price that is provided by the user. The name and price are passed to the \_\_init\_\_function of the Product class, and the init function creates an instance of that class, which is assigned to the name objectP. Figure 4 show the completed code for the Presentation section and Figure 6 shows the test code.

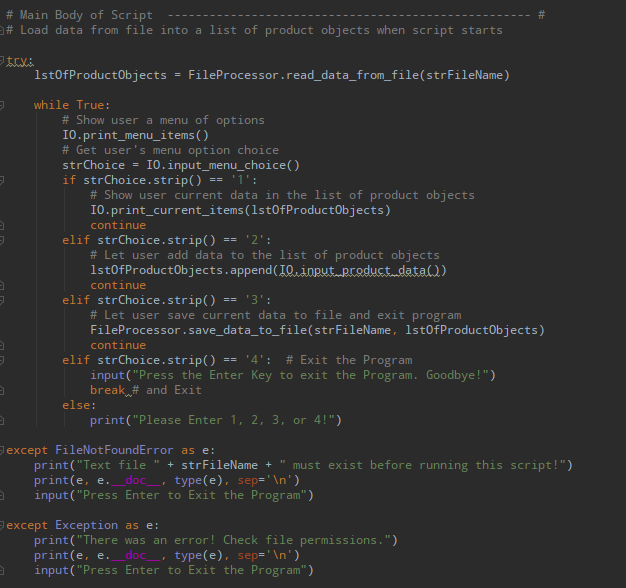
******

******

**Figure 4 Screenshot of Presentation Layer**

**Main Body of Script**

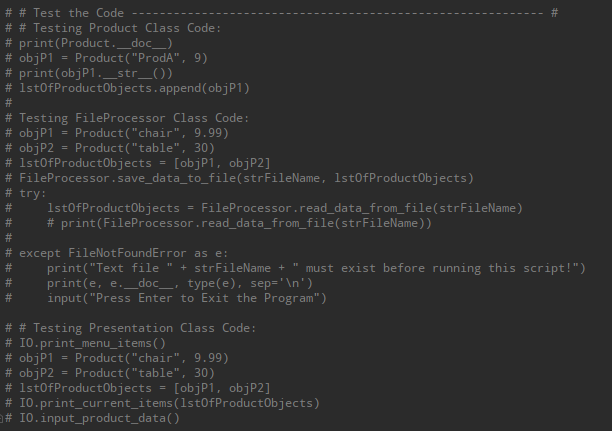
The Main Body of the Script is the layer of the code where I call the functions defined earlier. In this section, I added code to load data from the file into a list of product objects when the script starts. Figure 5 shows a screenshot of the Main Body of Script. Again, much of the code here was adapted from Assignment06, and I also included some error-handling (try-except).



**Figure 5 Screenshot of Main Body of Script**

**Testing the Code**

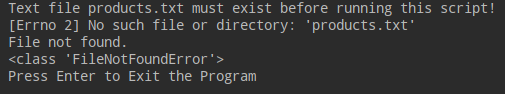
I worked on each layer separately until I got the code to work. Figure 6 shows a screenshot of the code that was used to test each section.

****

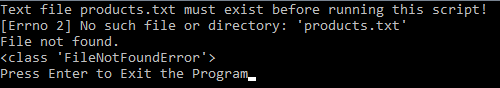
**Figure 6 Testing the Code**

**Running the Script**

When the program starts, data is loaded from the products.txt file if it exists. If the file doesn’t exist, an error message will appear. Figure 7 shows the script running in PyCharm without the existence of the products.txt file, and Figure 8 shows the script running in the Command Window.

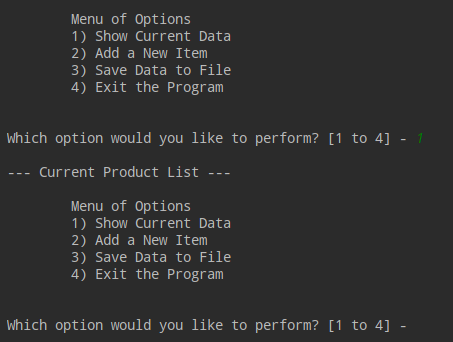
******

**Figure 7 Script running in PyCharm – FileNotFoundError**

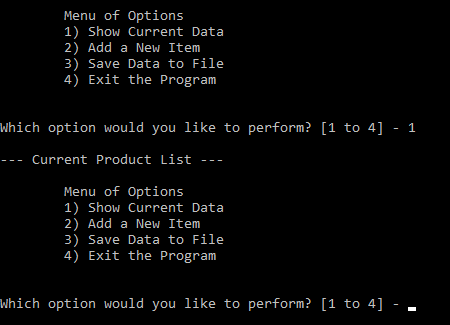
****

**Figure 8 Script running in Command Window – FileNotFoundError**

Once the text file is created, the program is re-started. Figure 9 shows a screenshot of the Menu of Options with user choice = ‘1’ in PyCharm, and Figure 10 shows a screenshot of the Menu of Options with user choice = ‘1’ in the Command Window. Since there is no data inside the text file, it returns an empty list.

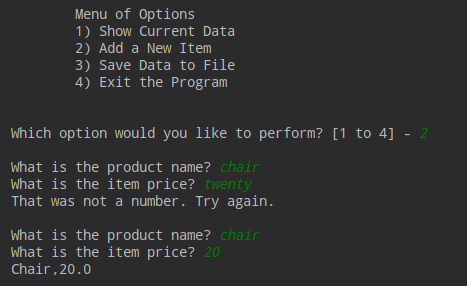
******

**Figure 9 Screenshot of Menu of Options and Option = 1 in PyCharm**

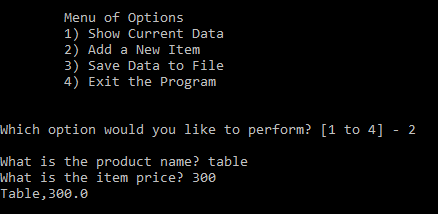


***Figure 10 Screenshot of Menu of Options and Option = 1 in Command Window***

Figure 11 shows a screenshot of Option 2 being selected in PyCharm, including an example of the exception handling being raised. Figure 12 shows a screenshot of Option 2 selected in the Command Window.

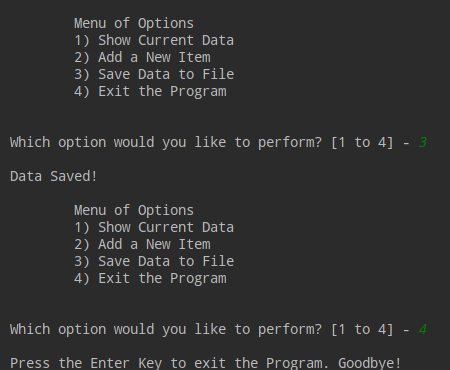


**Figure 11 Screeneshot of Option 2 in PyCharm**

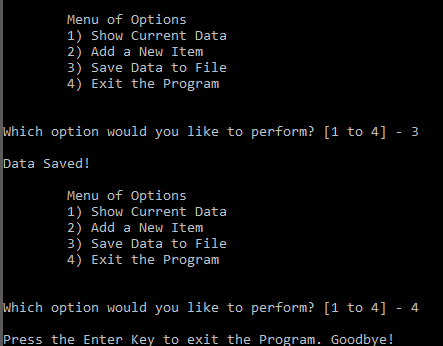


**Figure 12 Screenshot of Option 2 in Command Window**

Figure 13 shows a screenshot of Option 3 being selected to save data to file and then Option 4 being selected to exit the program in PyCharm. Figure 14 shows a screenshot of Options 3 and 4 in the Command Window.



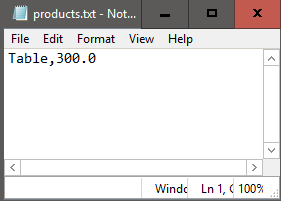
**Figure 13 Screenshot of Options 3 and 4 in PyCharm**



**Figure 14 Screenshot of Options 3 and 4 in Command Window**

**Verifying the Script Worked**

I verified that the script worked by opening the products.txt located in the \_PythonClass\Assignment08 folder. Figure 15 shows a screenshot of its contents.

****

**Figure 15 Screenshot of products.txt**

Summary

For this week’s homework assignment, I modified a script containing starter code from a file called “Assigment08-Starter.py” to manage a list of products and their prices using what we learned about objects and classes and adding code to make the application work. This was again, a challenging assignment for me, but it helped to write and test each layer of code separately. Where I felt I struggled the most and spent the most of my time with this assignment was trying to use the setter properties to perform the validation and error-handling. I did not want the program to exit abruptly or not give the user the opportunity to correct the error, so I addressed it by creating a try-except block in the input\_product\_data() method of the Presentation class. The script executes as I want it to, but I am a bit disappointed that I could not quite figure out how the error handling within the setter properties would be used.