Josh McMillen

12-08-2020

IT FDN 110 A

Assignment 08

<https://github.com/mcmillj2/IntroToProg-Python-Mod08>

Foundations Of Programming: Python

Performing Assignment 08

**Introduction**

This document lays out how to perform assignment 8 in the Foundations of Program: Python course at the University of Washington. In order to complete this assignment I used PyCharm to create a script that uses classes for product names and values to create objects.

**Creating the Python script and overall flow**

Since PyCharm was already setup on my machine I simply created a new project in a Assignment08 folder on my C drive. This assignment was to create a script that used object classes to display product names and values, then allow the users to update these items and save them. To accomplish this I decided to make a menu based program that would have 4 hard coded products and values. These values would be read when the script was opened and a menu would allow the user to view the items, update them, save and exit.

**Script header and overall flow**

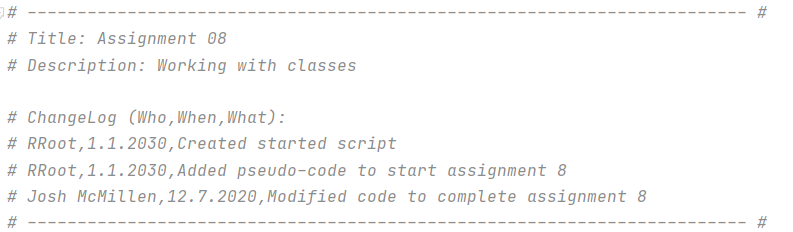
I updated the basic script header which included my name/date/desc (figure 1). Next the text file was introduced with a hardcoded name. Next a list of objects was declared, but empty. The main body of the script was a simple while loop that would go through the various user options.

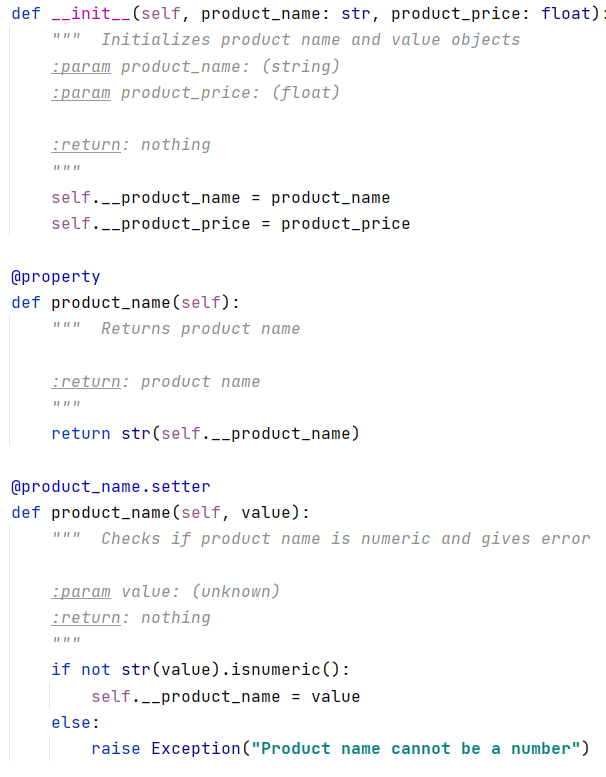
**Class Product**

Creating the class product to setup the objects was the main new part of this assignment. Here the first piece was to use the “\_\_init\_\_(self, product\_name: str, product\_price: float):” code as seen in figure 2. The purpose of this is to initialize an object with a product name and price property. These properties are then setup to have a “getter” and “setter”, which gather and display the specific properties (name or price in this case) of that object. The idea is that each “product” is an object that has a “name” and “value”, like talking about a thing (person/place/etc.), rather than storing this data in a more abstract way. This way additional properties could be added later, like “location” or “quantity” etc. and it would be easy as a developer. Last the “getter” properties have error handling that checks for the right format, this is used in the input/output section later (output seen in figure 4 and 5).

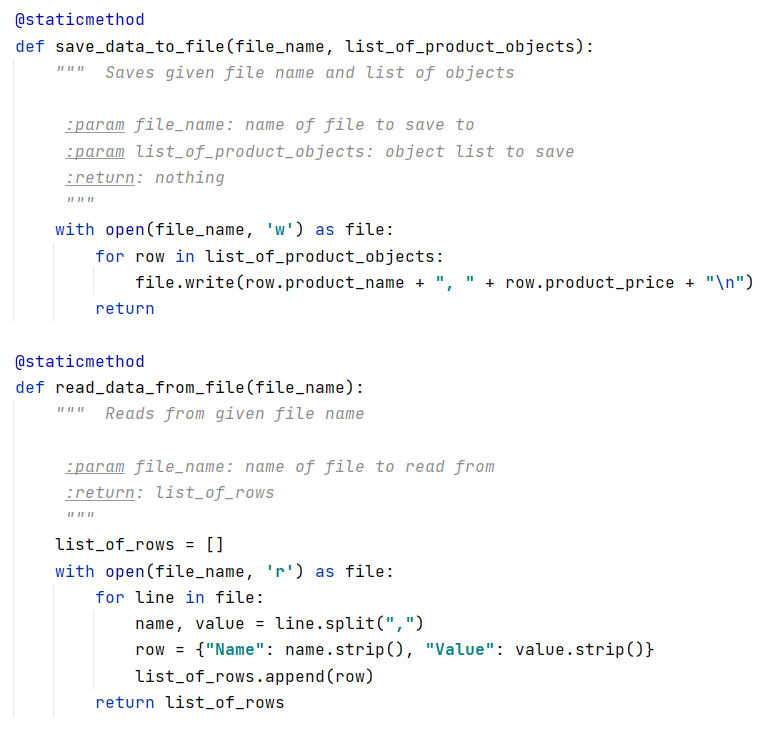
**File Processor**

The File Processor class was the other main new item in this assignment. Here the idea was to have two functions that used the product class objects as seen in figure 3. Here the main idea was to have a function that saved data and another to read data. The main difference from prior assignments was to use the class properties (name and price) in a more straight forward way than the previous methods that were more brute force. The method here is simply to use “row.product\_name” for example to get the product name. For reading the data I choose to store the data in a list of dictionaries, that would then be converted into objects using those dictionary references. The output examples for these functions are seen in figure 6, 7 and 8.

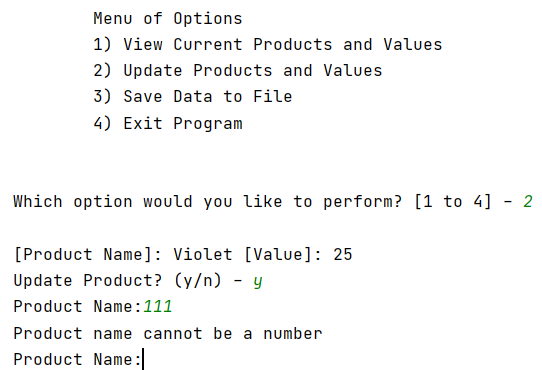
**Figure 1: Header**

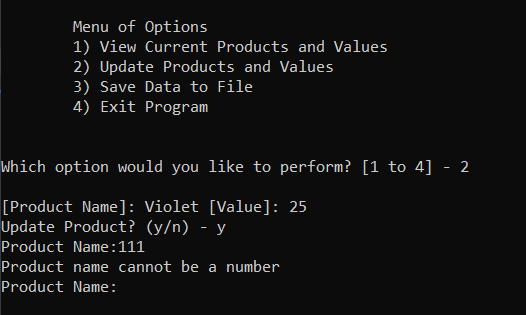


**Figure 2: Initializing product name and price, plus properties**

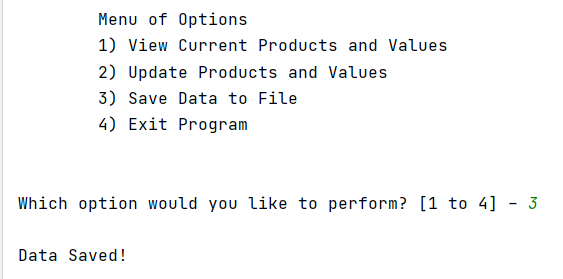


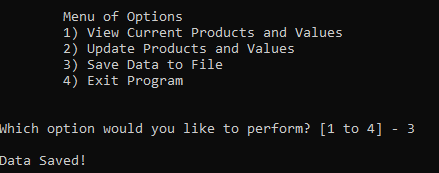
**Figure 3: Reading and saving object classes to a file**



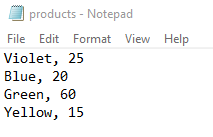


**Figure 4 and 5: Error handling in PyCharm and Command Line**





**Figure 6 and 7: Saving to a file example in PyCharm and Command Line**



**Figure 8: Text Output**

**Summary**

To fulfil the requirements of assignment 8 I updated a start script to use mainly objects created by classes. A main user loop interface would give the user the options to view, update, save and exit the program, using the product objects along the way.