Adrian Ronsse

September 8, 2021

Foundations Of Programming: Python

Assignment 08

**Objects and Classes**

GitHub repository: https://github.com/adrianronsse/ITFdn110-Mod08

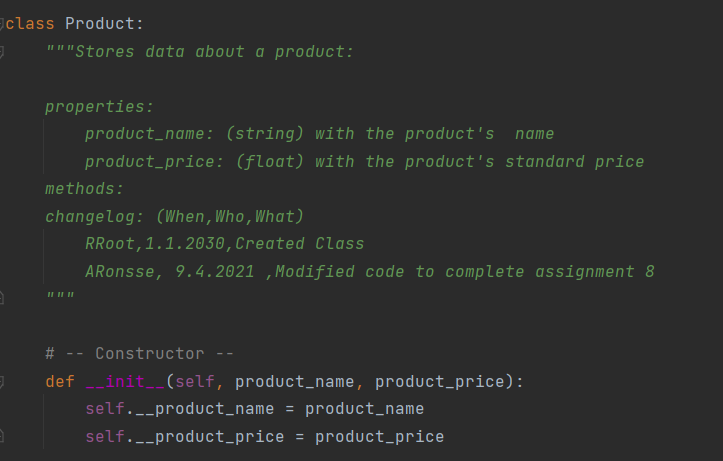
**Introduction:**

This assignment covered using classes to create different object instances. We then utilized this concept to collect user data for different products using a menu, similar to previous assignments.

**Writing the code:**

Step 1 – Create class and setting up constructor

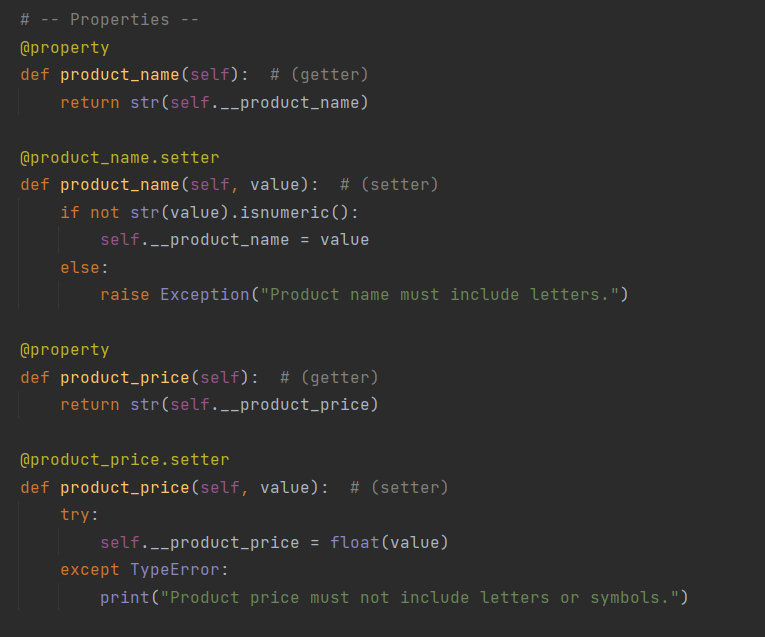
Here I set up a class called product to serve as a blueprint for the objects to be created. The first thing to do is to add a constructor method that can set the initial attribute values of an object, although in this case it does not set any values, it just initializes the attributes within the class.



*Figure 1 – Setting up a class and constructor*

Step 2—Setting up properties with getter and setter methods

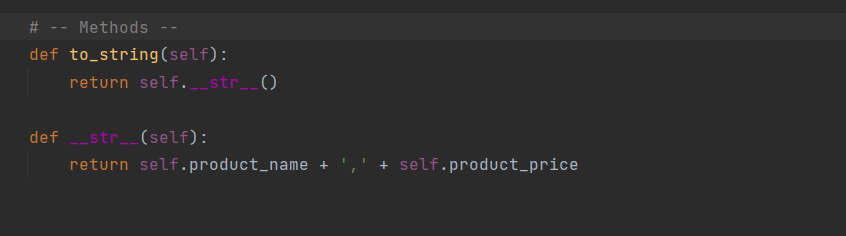
Properties allow me to set requirements for the attributes in the class. Here I use error handling techniques learned in Assignment 7 to require that the product\_name attribute should not be numeric, and that the product\_price attribute should be a float value.



*Figure 2 – Properties and error handling*

Step 3: Methods

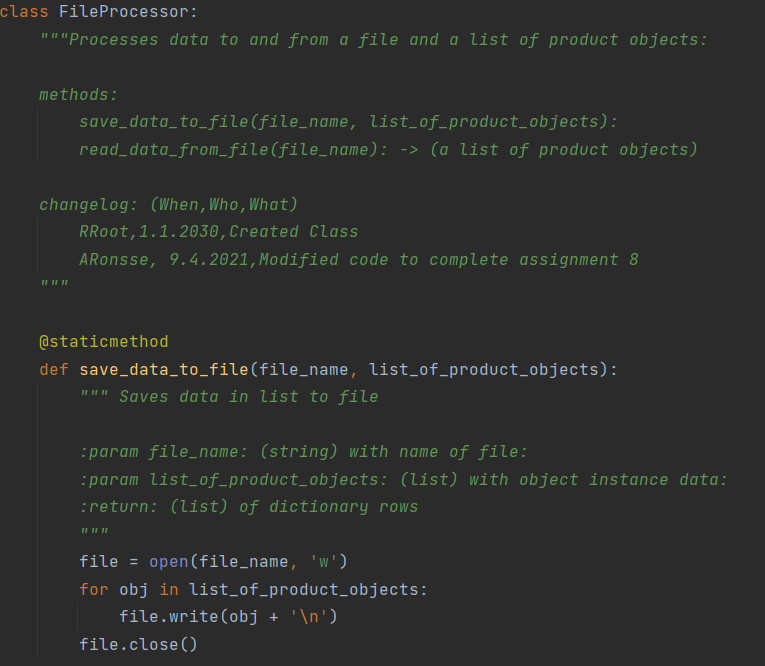
This part of the class makes the data in the object usable for other classes in the program. Producing the value as a string allows the program to work with the characters stored rather than a reference to the information’s location.



*Figure 3 – Preparing object data to be read as a string*

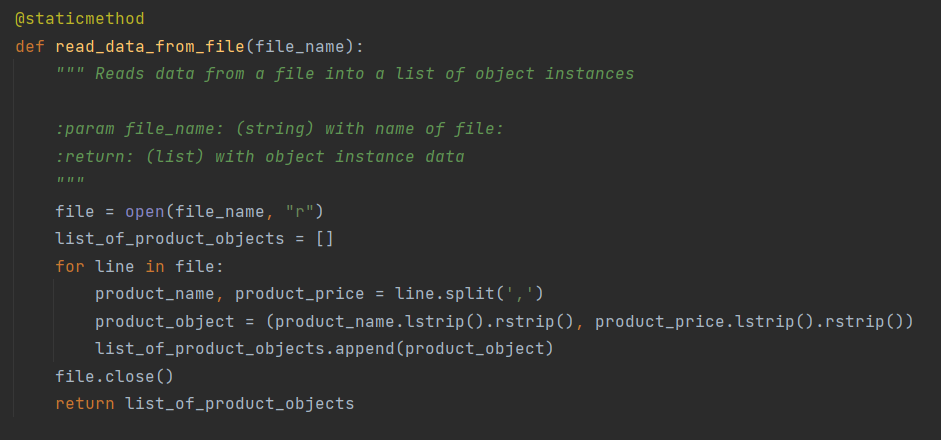
Step 4: File Processor class

I needed to create another class that would handle file storage and retrieval for the data collected in the program. Per Mr. Root’s framework, this class was called File Processor. I used a static method for this, since there is no need to create multiple object instances of this class.



*Figure 4 – Writing list to file*

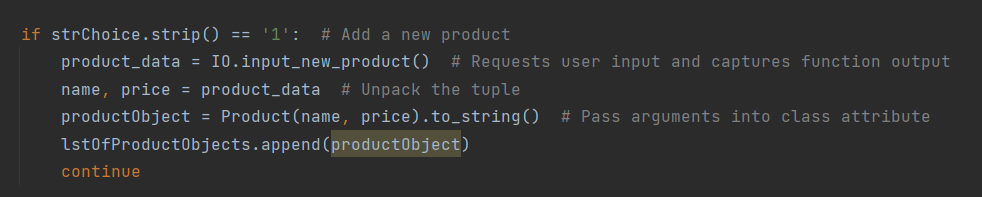
I also needed a method to read the information from a file. I had a lot of trouble getting this format to match the new inputs that would be created in a single user session in the program, and I failed to ever produce a perfect match, but here is the best effort I was able to generate:



*Figure 5 – Reading data from file*

Step 5: IO Class and main body of program

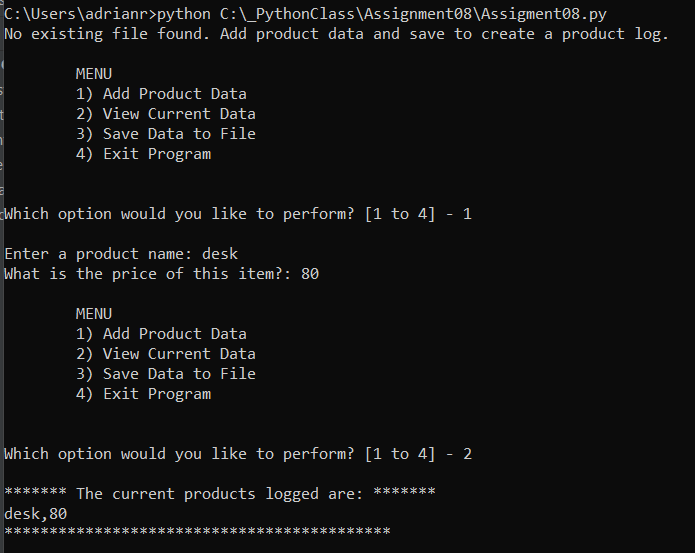
To complete these sections I largely recreated the same code that I had used in Assignments 6 and 7 to achieve the same results. These sections of the code only interacted with my Product class once, so I have provided a snapshot of that event below, where the product method output feeds as a string into storage in the list of product objects:



*Figure 6 – Storing data as a product object*

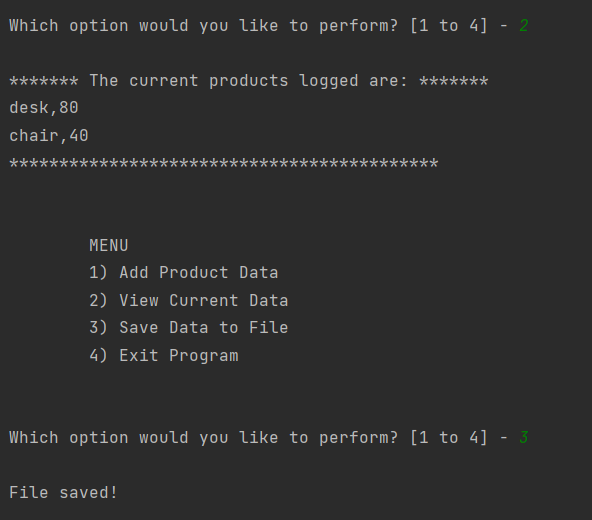
**Testing the code:**

After preparing the code, I needed to test it to make sure it was working as expected. First I opened the program in the command console to test. Here I see that I am able to navigate the program and enter new product data:



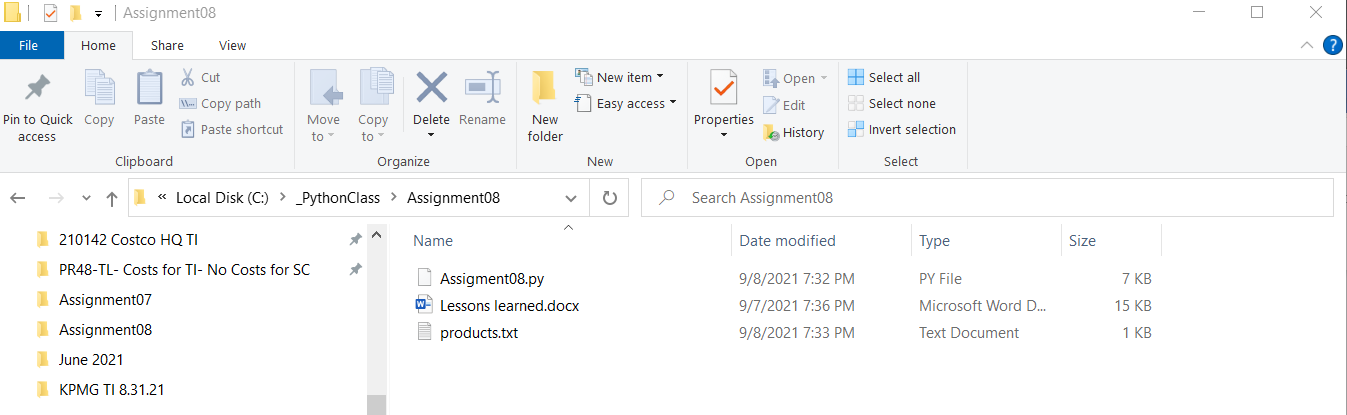
*Figure 7 – Command console test*

Then I tested it in PyCharm as well and found I was able to add more data and save to file.



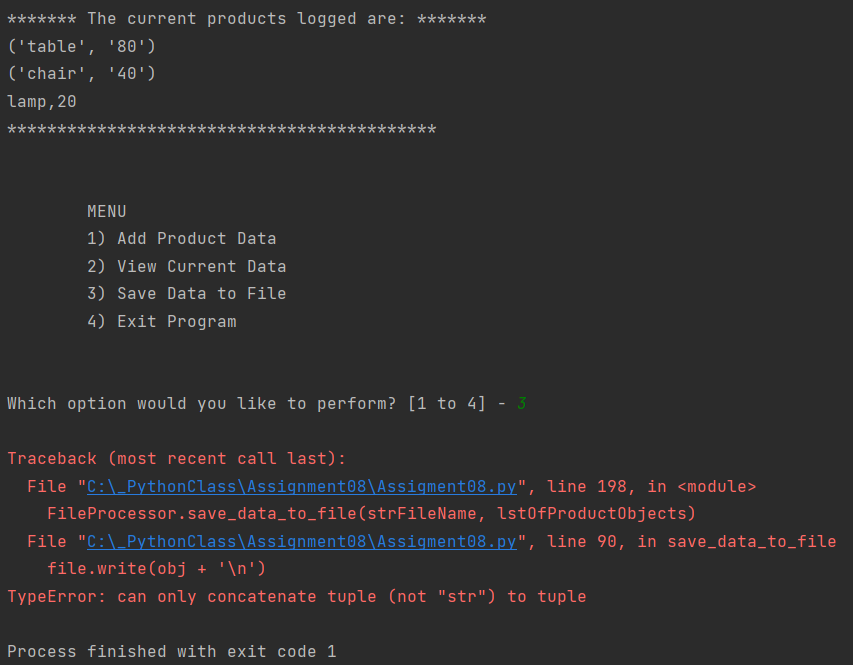
*Figure 8 – PyCharm test: data saves successfully*

I can confirm too, that the file has populated in my folder:



*Figure 9—Files saved in folder*

Unfortunately when I restart the program and try to save additional data, I see that the formats are not compatible. My data is messy when it prints out from the saved file and I am unable to save new data to the file.



*Figure 7 – Error saving additional data*

**Summary:**

In this coding exercise I was able to write a class that would produce multiple object instances. I included error handling in setting the properties for those object instances, but the error handling did not seem to engage when I fed data into the class. I also set up some methods to save the data to a file, but ran into some difficulty when trying to reopen the file from the program and save additional data.