Kyle Biondich

8/29/2023

[IT FDN 110 A](https://canvas.uw.edu/courses/1655585)

Assignment 08

Class Funs

# Introduction

Week 8 of the course introduced classes, class attributes, setters and getters. The following paragraphs outline the steps I went through to finish adding code to the starter assignment file for creating the product class constructor, parameters, and methods, as well as the file processor and IO methods for reading, writing, and saving data to a text file.

# Intended Outcome

The intended outcome of this week is to present a menu of choices to the user, read product names and prices from a text file, add a new item to the list of products, save the list of items to a text file, and then exit out of the application.

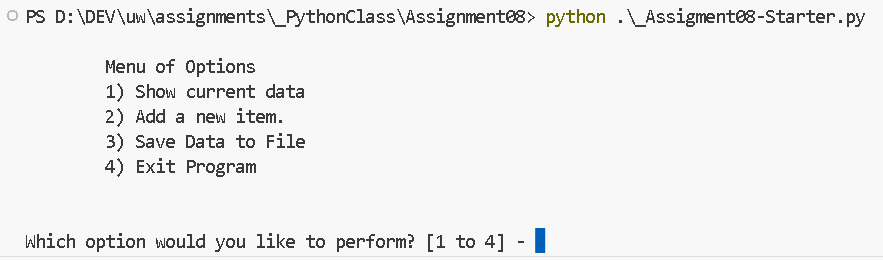


Figure 1: Intended Outcome: Menu Choices

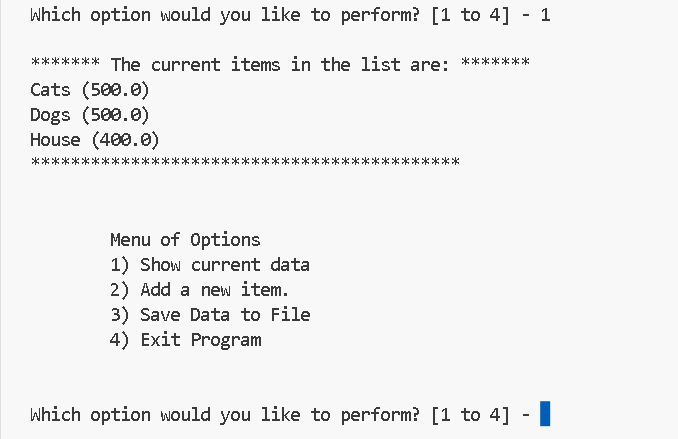


Figure 2: Intended Outcome: Read Current Items

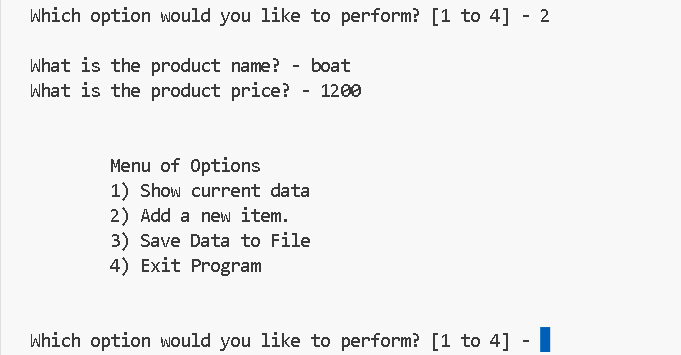


Figure 3: Intended Outcome: Add a new item

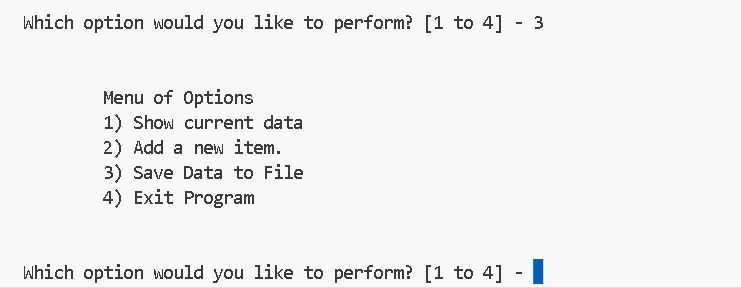


Figure 4: Intended Outcome: Save data to file

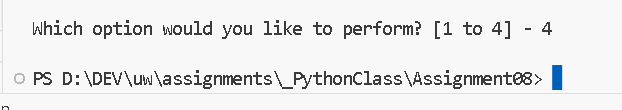


Figure 5: Intended Outcome: Exit Program

**Adding Code to the Product Class**

The first task was to add the code for the constructor for the products class that contained properties product\_name and product\_price. This was done by first creating the initialization method and passing in self, product\_name, and product\_price, then setting them to the object upon initialization, as can be seen in lines 27 through 31 in Figure 6. Next, the properties were defined along with their setter methods, lines 33 through 54 in Figure 6. And finally the to\_string method was created in a methods subsection, lines 56 through 62 in Figure 6.

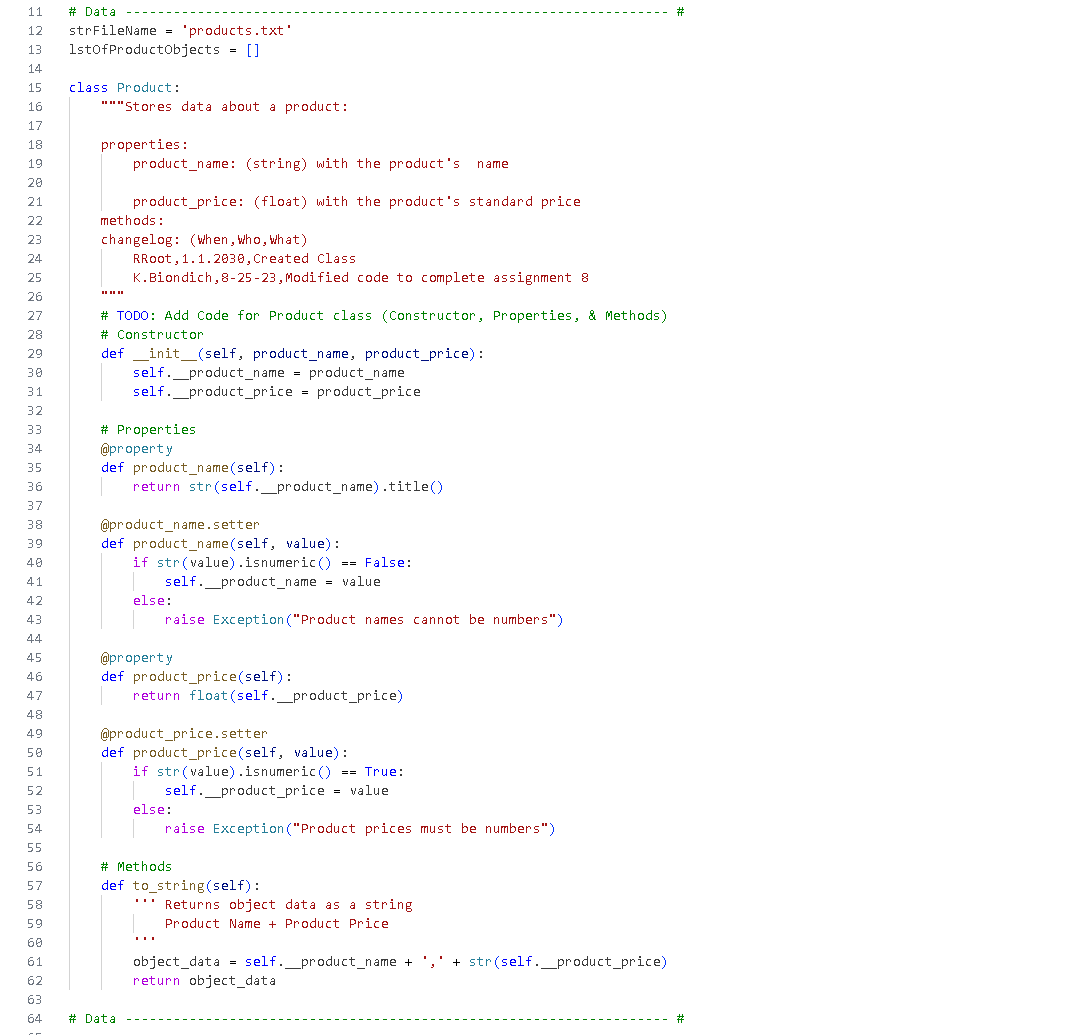


Figure 6: Product Class

## File Processor Class

The next class is the FileProcessor class, where functions pertaining to data processing into and out of a file are contained. The first method, read\_data\_from\_file, lines 80-94 in Figure 7, reads data from a file into a list of objects. The next method, save\_data\_to\_file, lines 97-108 in Figure 7, writes data from a list of objects to a products.txt file.

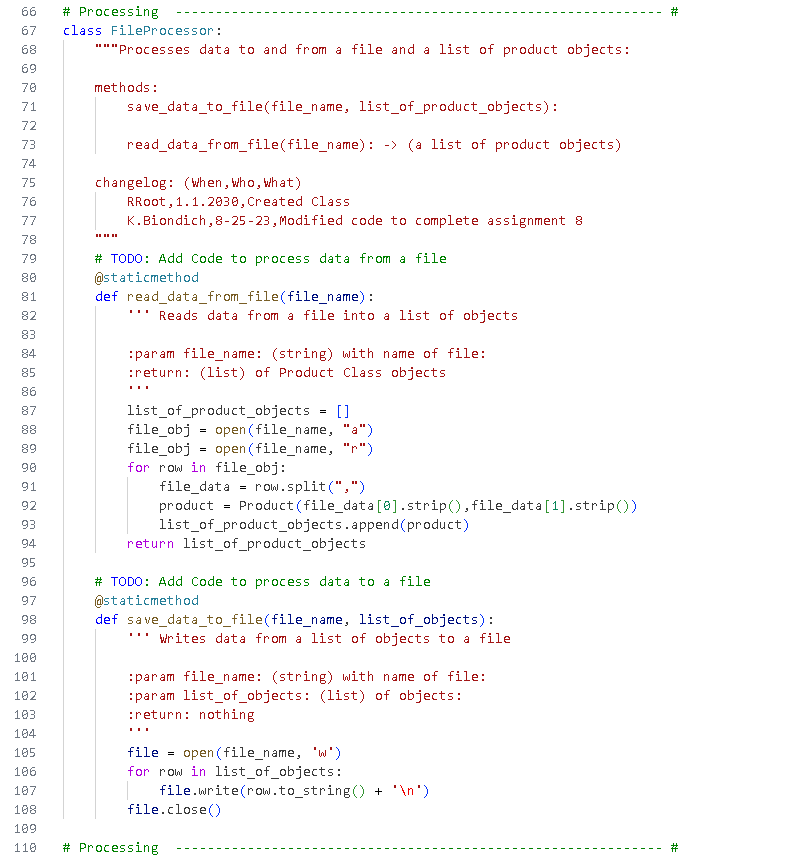


Figure 7: File Processor Class

# IO Class

The IO class contains the methods that pertain to interaction with the user. The first, print\_menu\_items, lines 125-136 in Figure 8, produces the printed menu list when called. The next method input\_menu\_choice, lines 138-152 in Figure 9, gets the menu choice from the user. The next method, print\_current\_list\_items, lines 154 – 165 in Figure 8, shows the current items in the list of product objects. And lastly, the method input\_product\_data, lines 167 – 187 in Figure 9, gets data for a product object from the user.

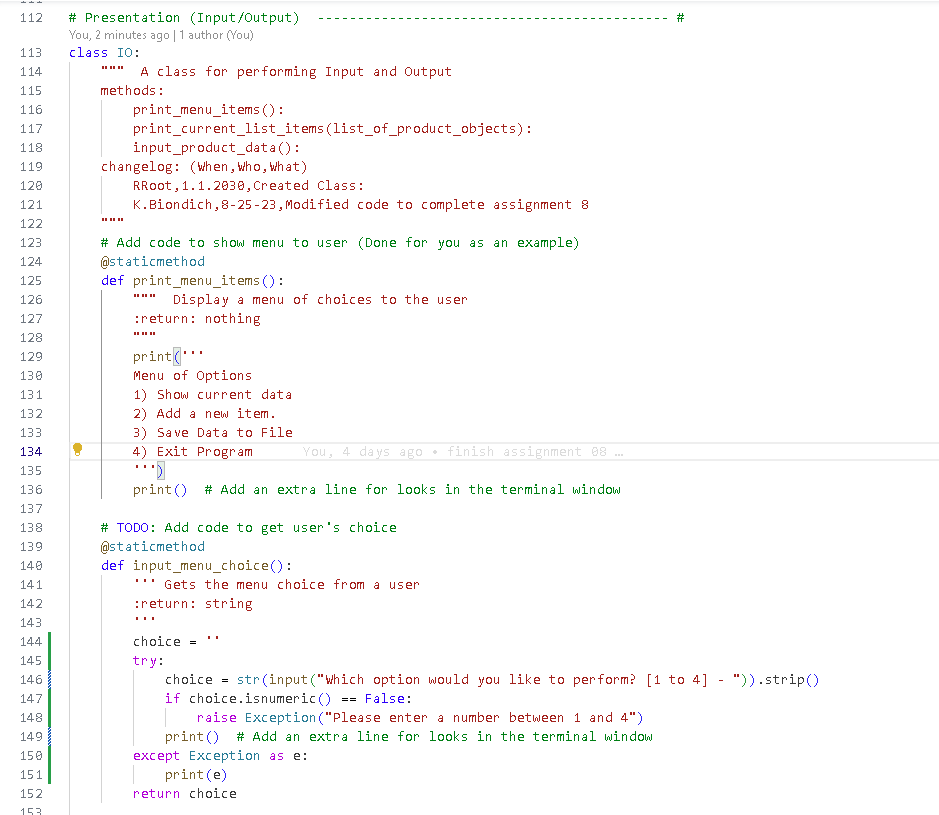


Figure 8: IO Class

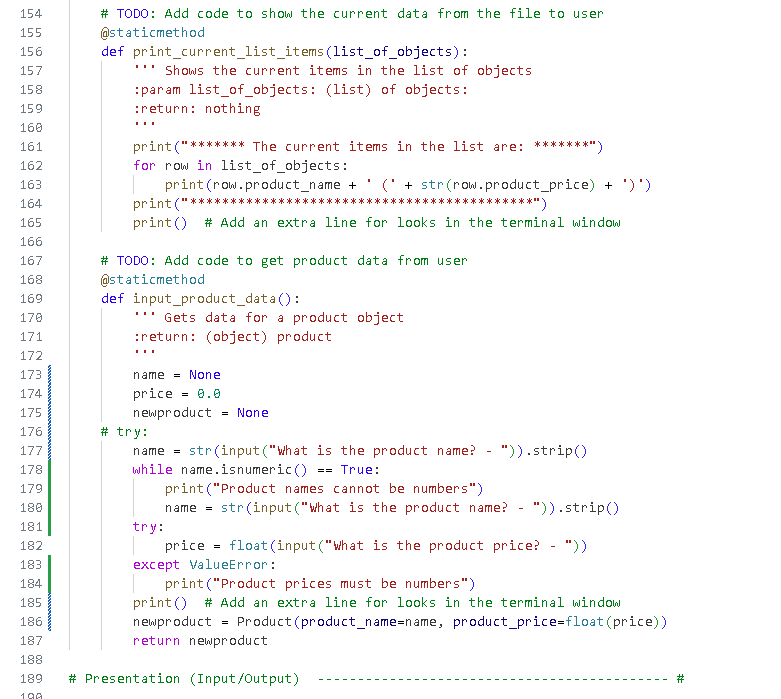


Figure 9: IO Class

# Main Body Of Script

The main body of the script contains the code for running the script and initializes the list of product objects that may be stored in the products.txt file. Line 195 in Figure 9 creates the lstOfProductObjects variable and calls the FileProcessor.read\_data\_from\_file method, passing it the strFileName variable that was initialized at runtime. Lines 197 – 219 in Figure 9 starts a while loop to always present the user with available script options, presenting first the menu (line 199), then a variable to capture the user’s choice (line 201), then option 1 (lines 203 – 205), printing a list of the current items in the list, then option 2 (lines 207-209) the option to save new data to the list, option 3 (lines 211-213) the option to save the list to the products.txt file, and finally option 4 (lines 215-216) the option to exit the script.

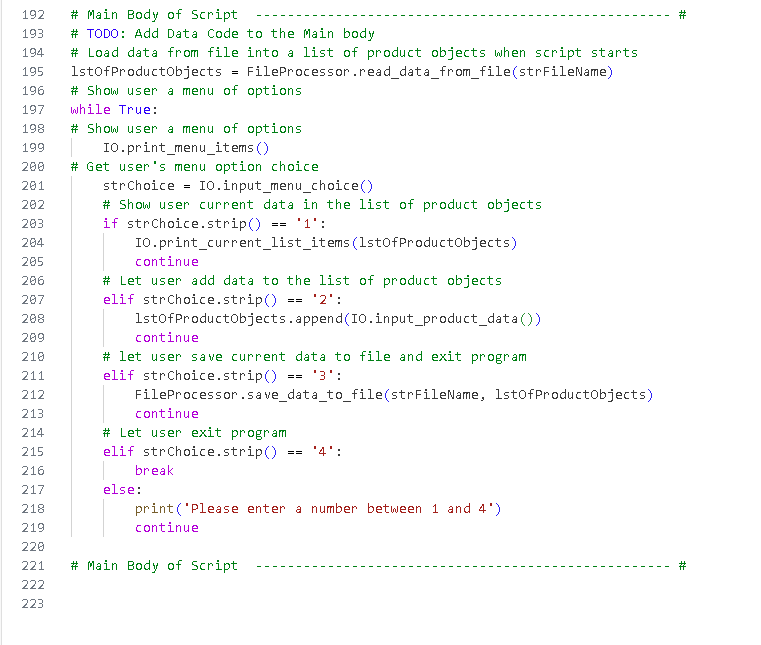


Figure 9: Main

# Observations

This was a difficult assignment. The most difficult part was in the “input\_product\_data” method, and trying to capture the exception of a string passed into the price variable that wouldn’t end the script. I eventually created something that works, but I’m not sure if it’s a good solution. I used a while loop for capturing the product name, but then had to use a try block to stop the “cannot convert string to float” error from exiting out of the script.

# Summary

In summary, utilizing all the resources provided to the class and the online lecture, this paper outlines all the steps that were taken to create a python script that results in a successful execution of the intended outcome (Figure 1). Following the steps outlined above will allow for the audience to recreate the presented result.

# References

*Questions*. (n.d.). Retrieved from Stack Overflow: https://stackoverflow.com/questions/736043/checking-if-a-string-can-be-converted-to-float-in-python