Andrei Arevalo

March 6th, 2021

Foundations of Programming, Python

Assignment08

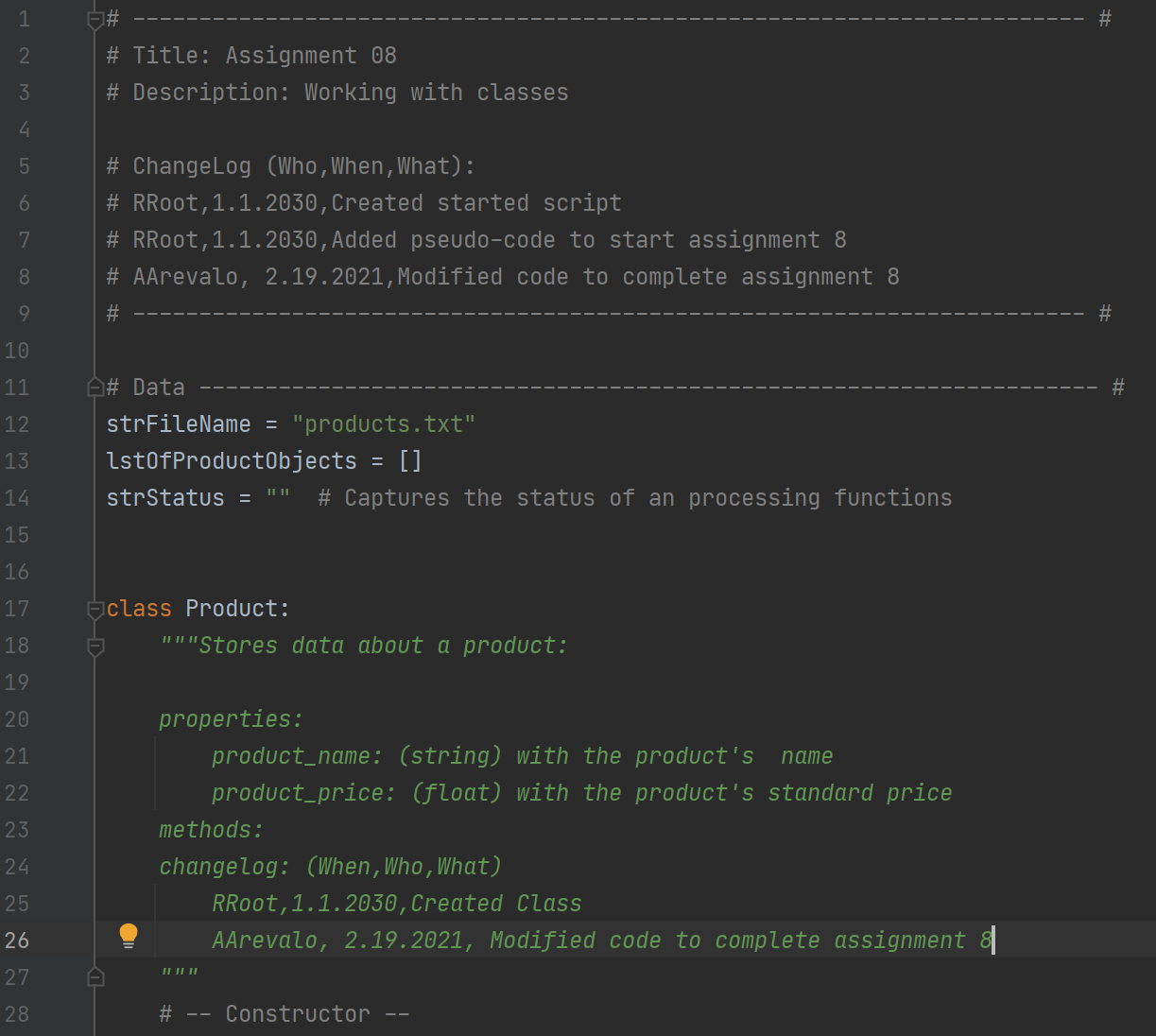
Working with Classes

# **Introduction**

In this assignment we are tasked with creating a script that manipulates product information including both the name and value. The format of this assignment is slightly similar to previous weeks in that we are manipulating data that gets saved into a text file. This was a good method of learning how to work with classes and using the different fields within the class. This homework was difficult in a sense that some elements of the classes were harder to understand that previous topics in other weeks.

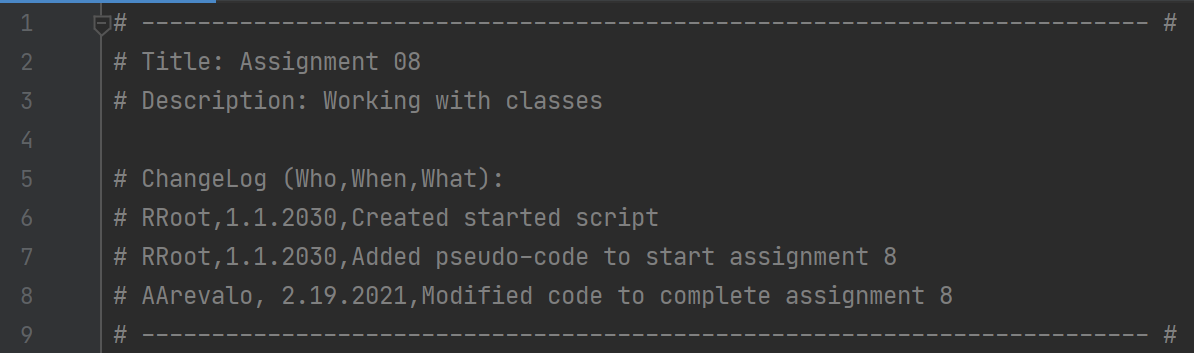
# **Creating the Script**

1. The purpose of this program is to create a script that manipulates and saves a product’s name and value to a text file. The format of this assignment is different than previous assignments in that this week we are asked to focus more on the use of glasses and the different parameters within a class such as methods and properties. (figure 1)



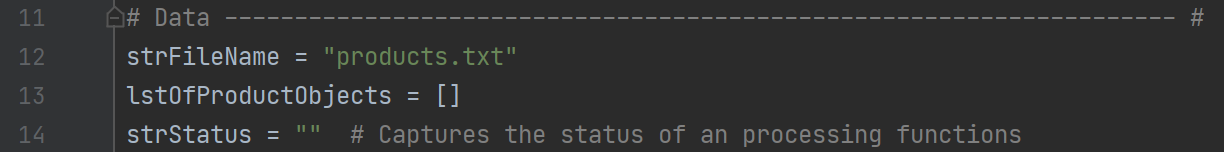
***Figure 1: The main script***

1. At the top of the script is the title block, which shows information about the script and keeps track of revision history in case the file needs to be revised in the future. The author and date are also included in this title block. In addition below the title block is a summary of the assignment with the task outlined. The goal of this assignment was to create a program that saves product data to a text file. (Figure 2)



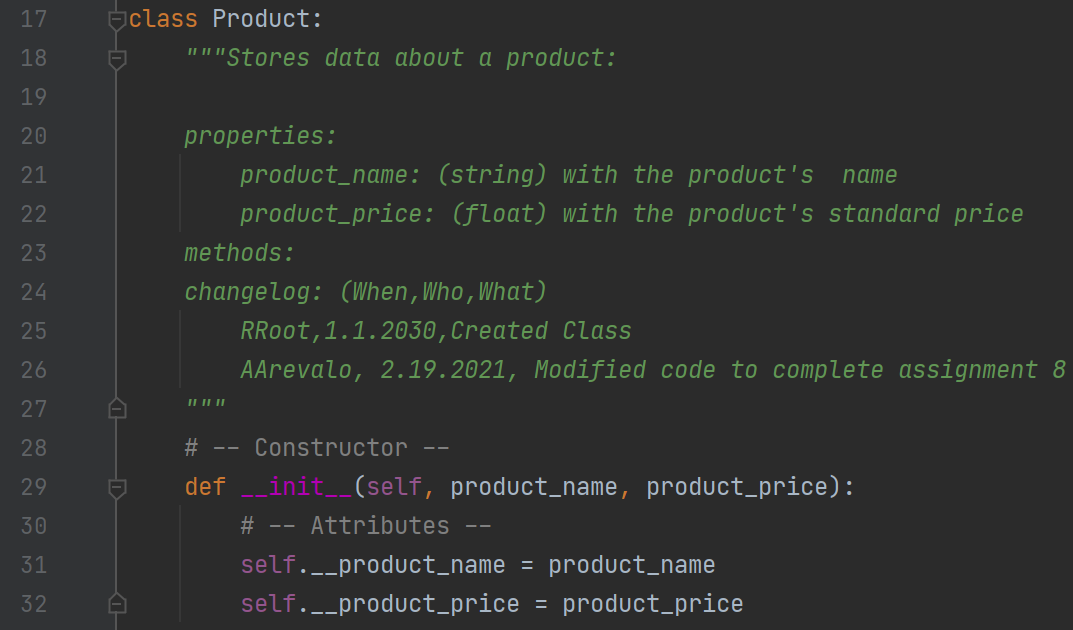
***Figure 2: Title block of the script***

1. At the beginning of the script, I introduce the variables that will be used in the script, which in this case is a list of product objects. In addition, I also call a txt file where the product name and values are too be stored. (Figure 3)



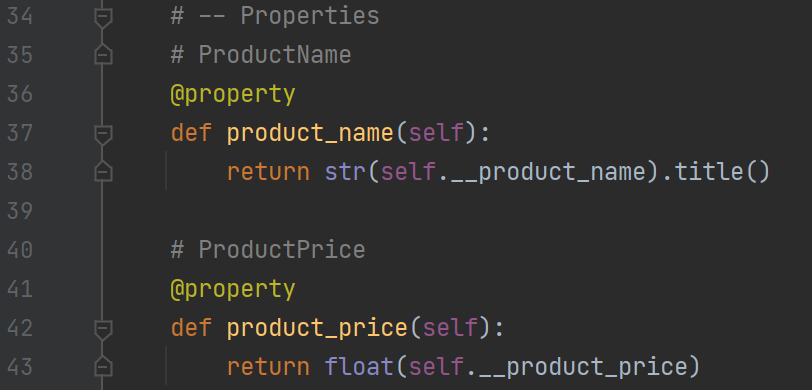
***Figure 3: Calling the text file and creating the list***

1. The first class that is introduced in the code is the “Class Product.” This class is what stores the information about the products name and value. At the beginning of the class, I list out the docstring, which highlights which properties will be put into the class. The first constructor that is used creates the product name and product price. The double underline is used to make it a private reference. (Figure 4)



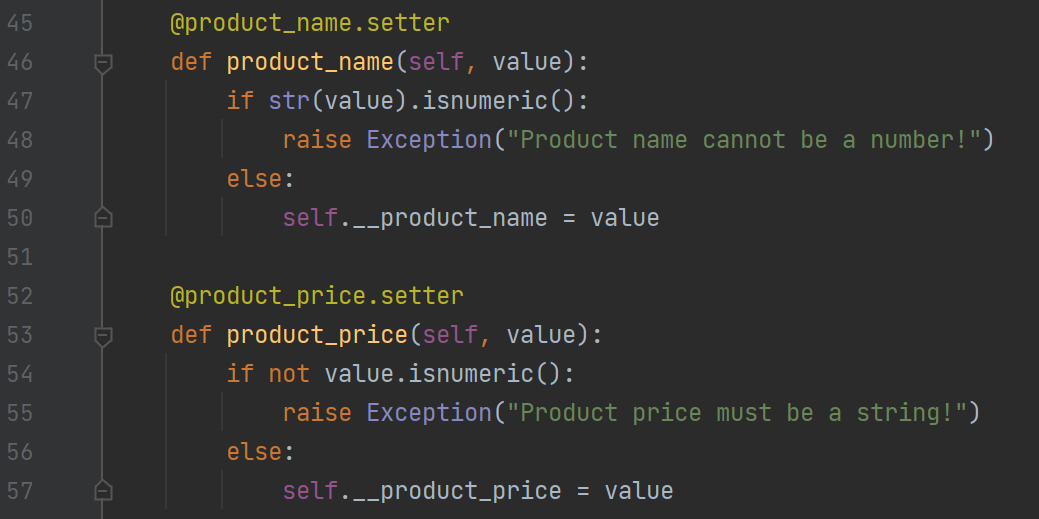
***Figure 4: Class - Product***

1. The next lines of code detail the properties of both the name and the value of the product. For the product name, the use input is turned into a string. Likewise, for the product value, the number is turned into a floating point number. (Figure 5)



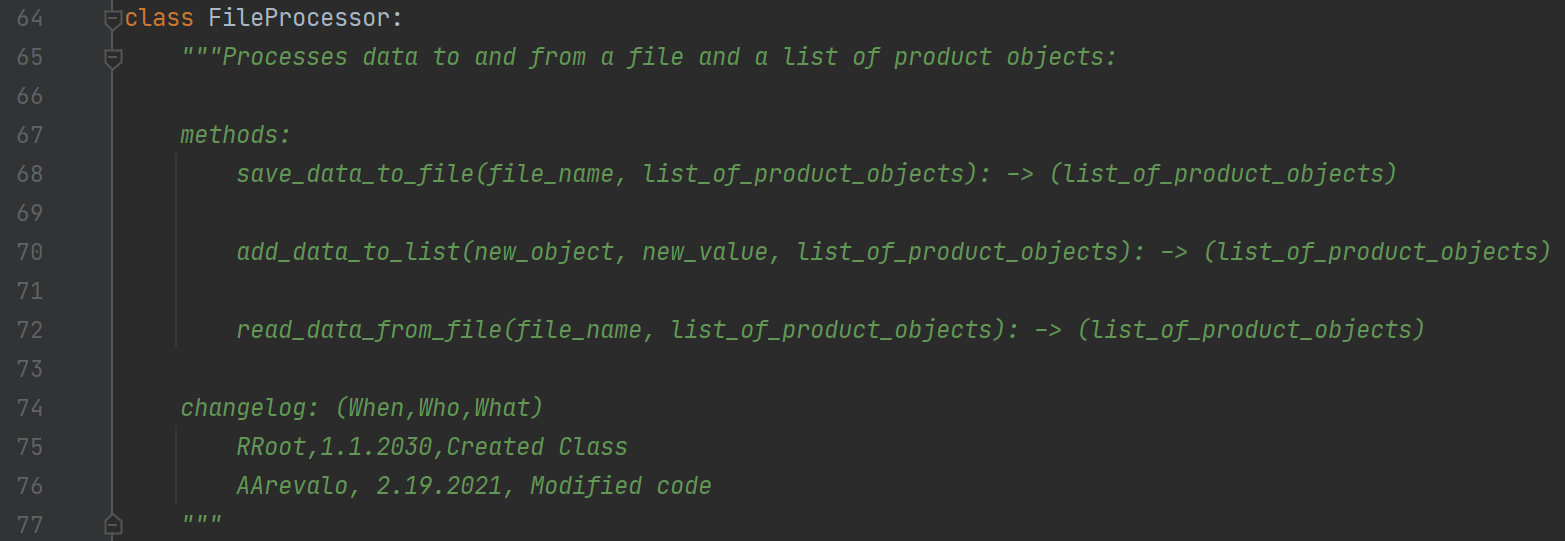
***Figure 5: Product Properties***

1. In the next lines of code, a setter is used to format the data that is being entered. For example, if a number is used for the product name, an exception will be made telling the user the name can only be a number. Likewise, the value of the object must be a number. (Figure 6)



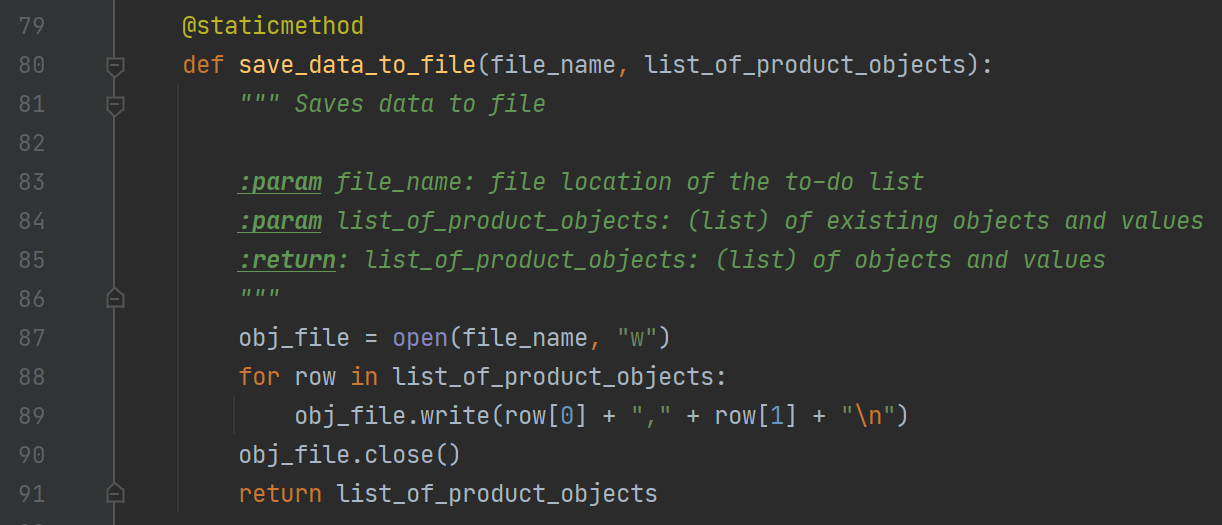
***Figure 6: Try and Except within the Class***

1. In the next section of code, I introduce the “Class File-Processor” which is used to manipulate the data. The three main methods as part of this class are saving data, adding data, and reading data. (Figure 7)



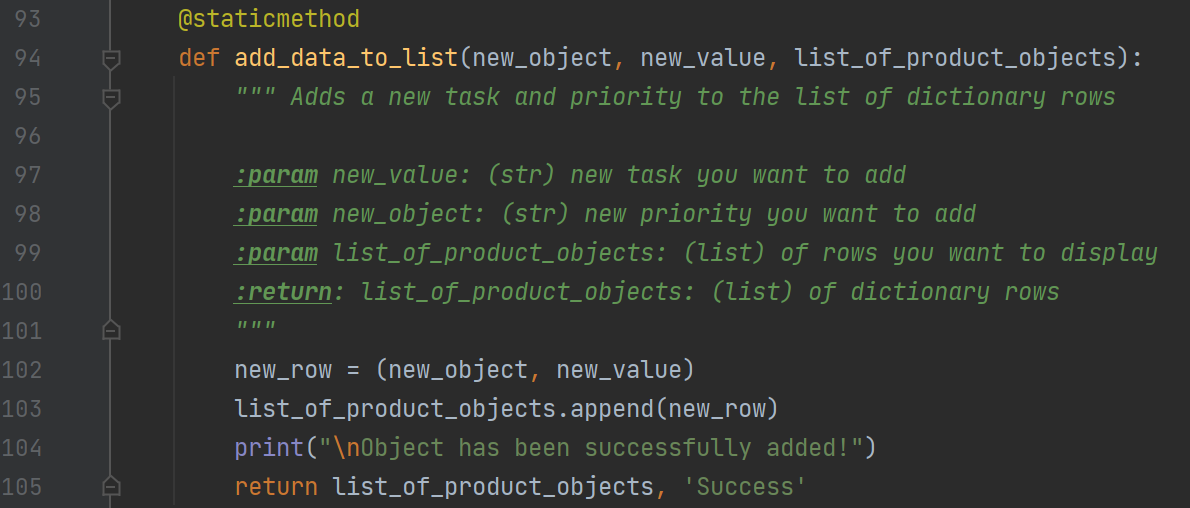
***Figure 7: Class – File Processor***

1. The first of the three methods in the file processing class is saving data, which goes through the list of product objects and writing the data to rows within the text file. The object and value are stored within each row of the file. (Figure 8)



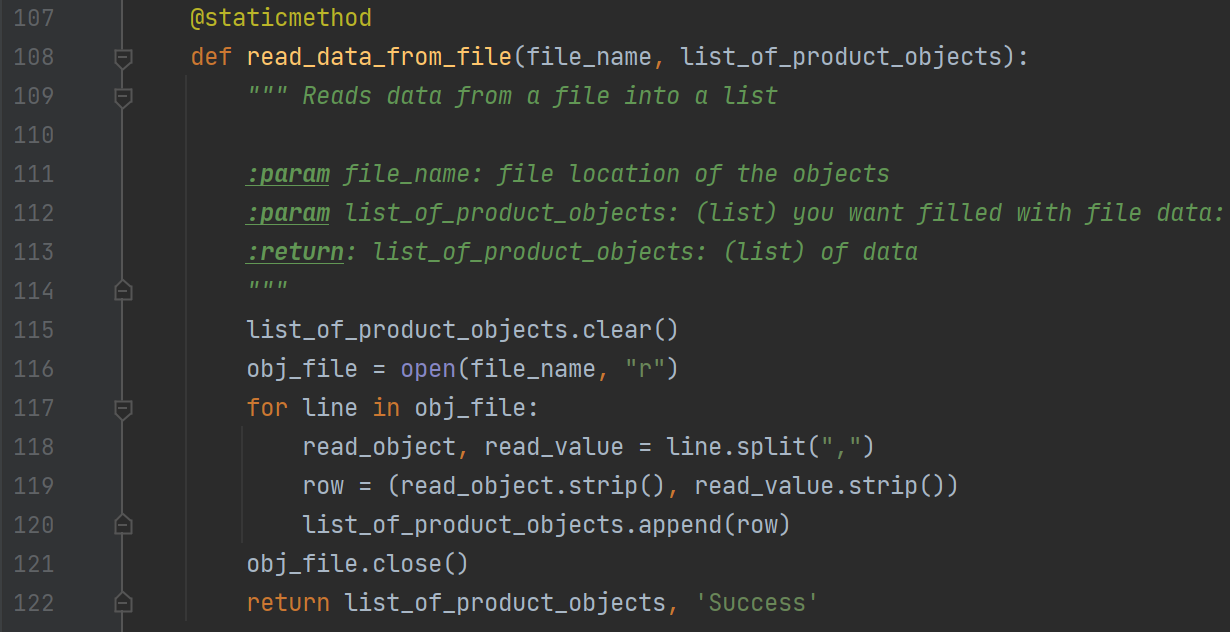
***Figure 8: Saving the Data***

1. The next method in the file processing class is the adding data method. This method goes through the new object and new value, and then adds this data to the list of product objects. (Figure 9)



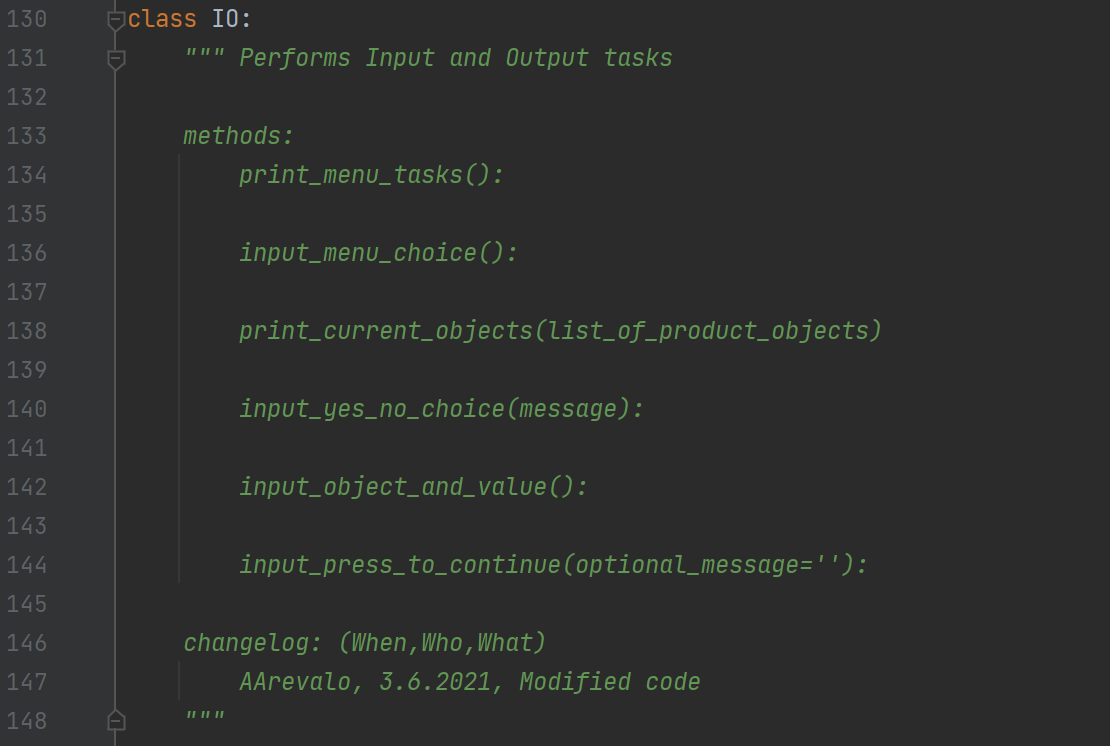
***Figure 9: Adding Data***

1. The third method within the file processing class is reading data. This method works by going through the existing file and reading the name and value from that list. From there, the list is then amended with that imported data, and the list of product objects is created. (Figure 10)



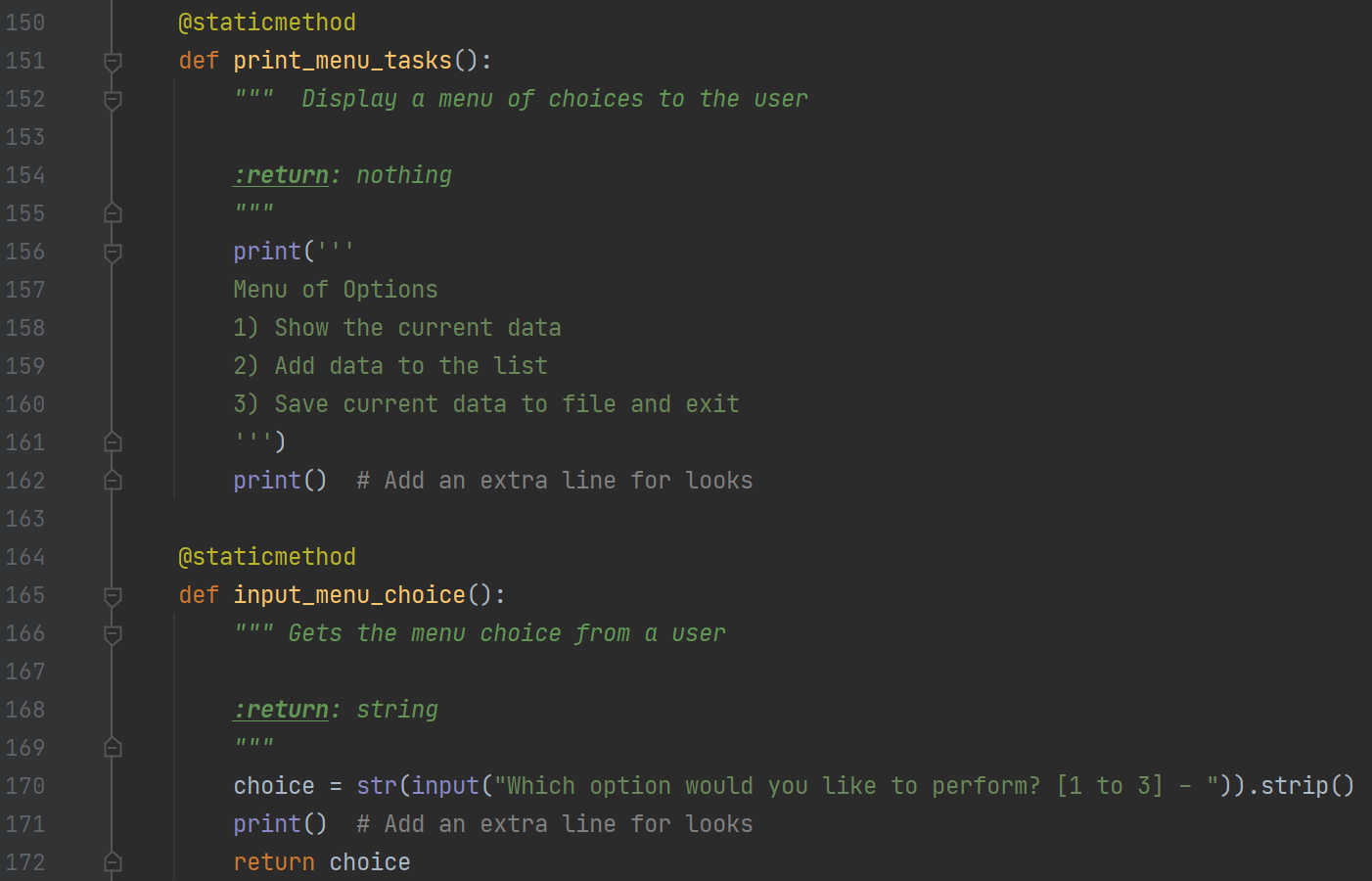
***Figure 10: Reading Data***

1. The last of the classes is the IO class, which performs all the outputs and inputs from the user. This class is used for methods outside of manipulating the product data. (Figure 11)



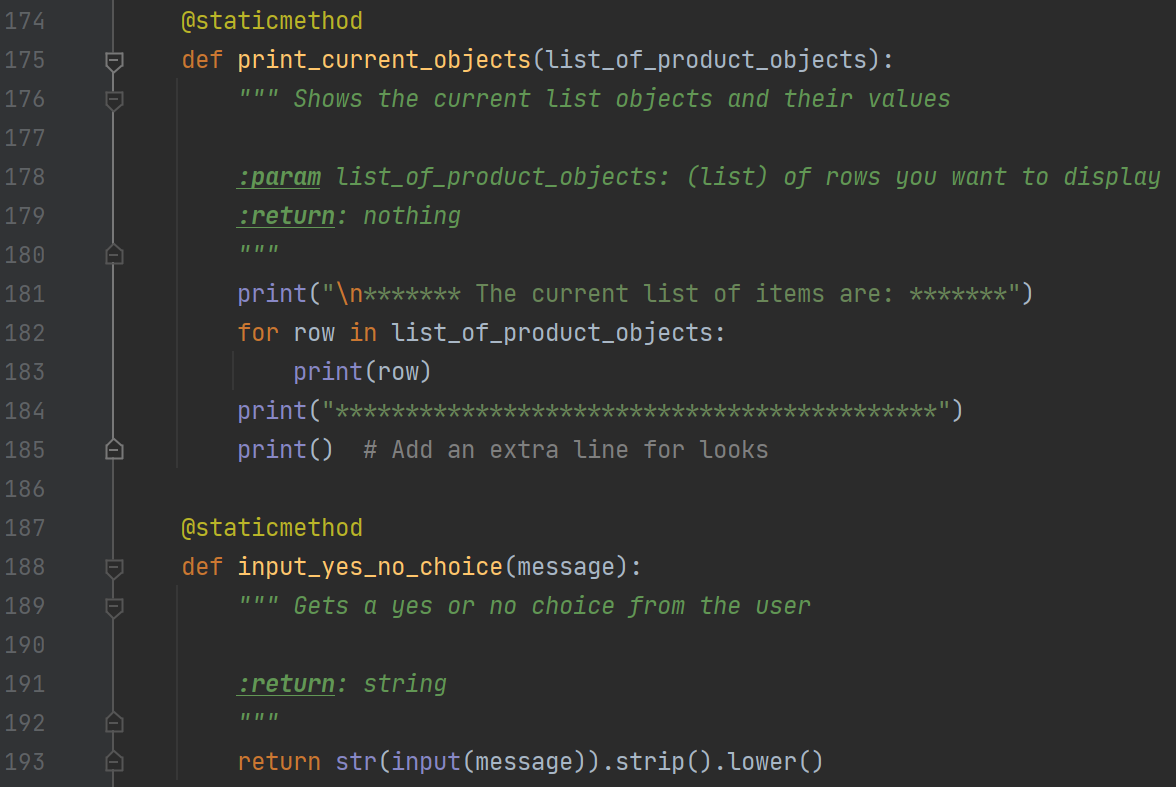
***Figure 11: Class IO***

1. The first two methods within the IO class are printing the menu and collect user input on the menu selection. The first of which uses the print function to show the menu, and the second method uses the input function to collect the user input. (Figure 12)



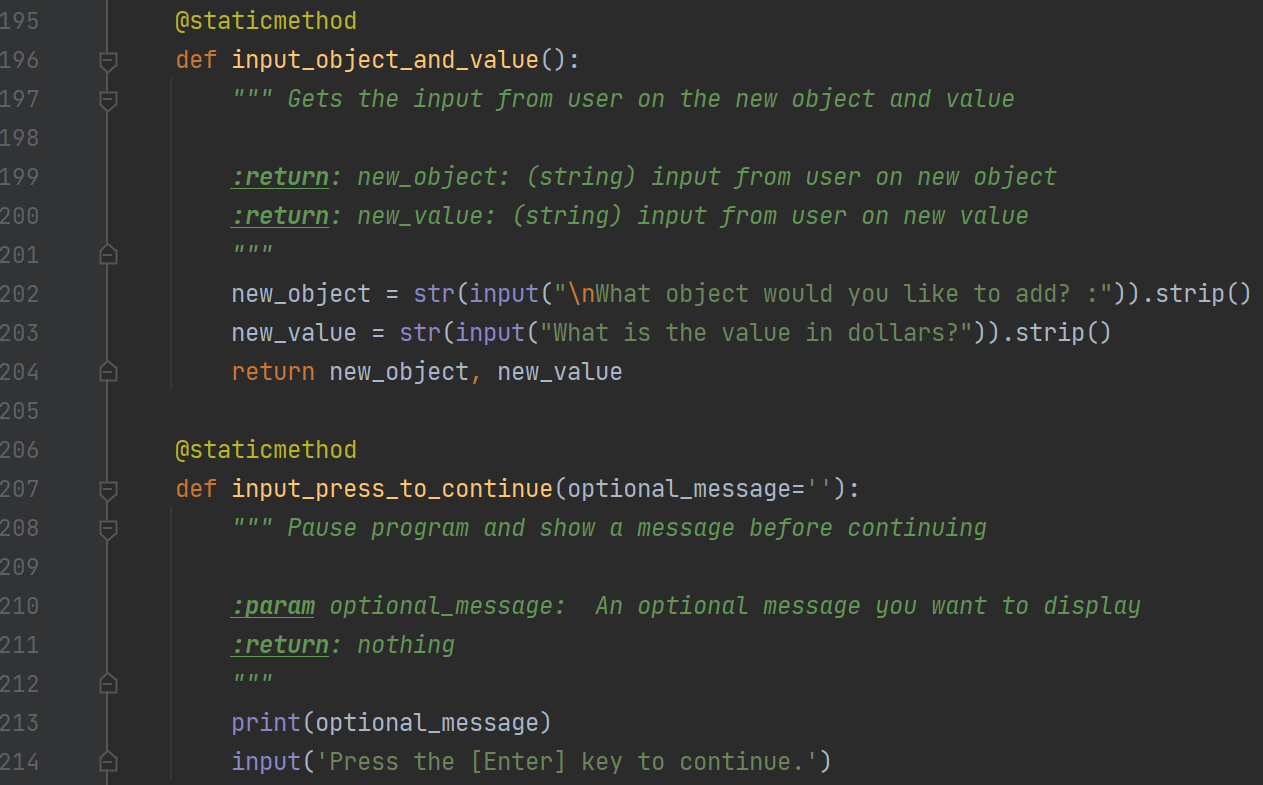
***Figure 12: Print Menu and Get User Selection***

1. The next methods are the print current objects and yes/no input. The first of which prints the existing data in the list of product objects. The second method gets a yes or no from the user which can be used for other methods later. (Figure 13)



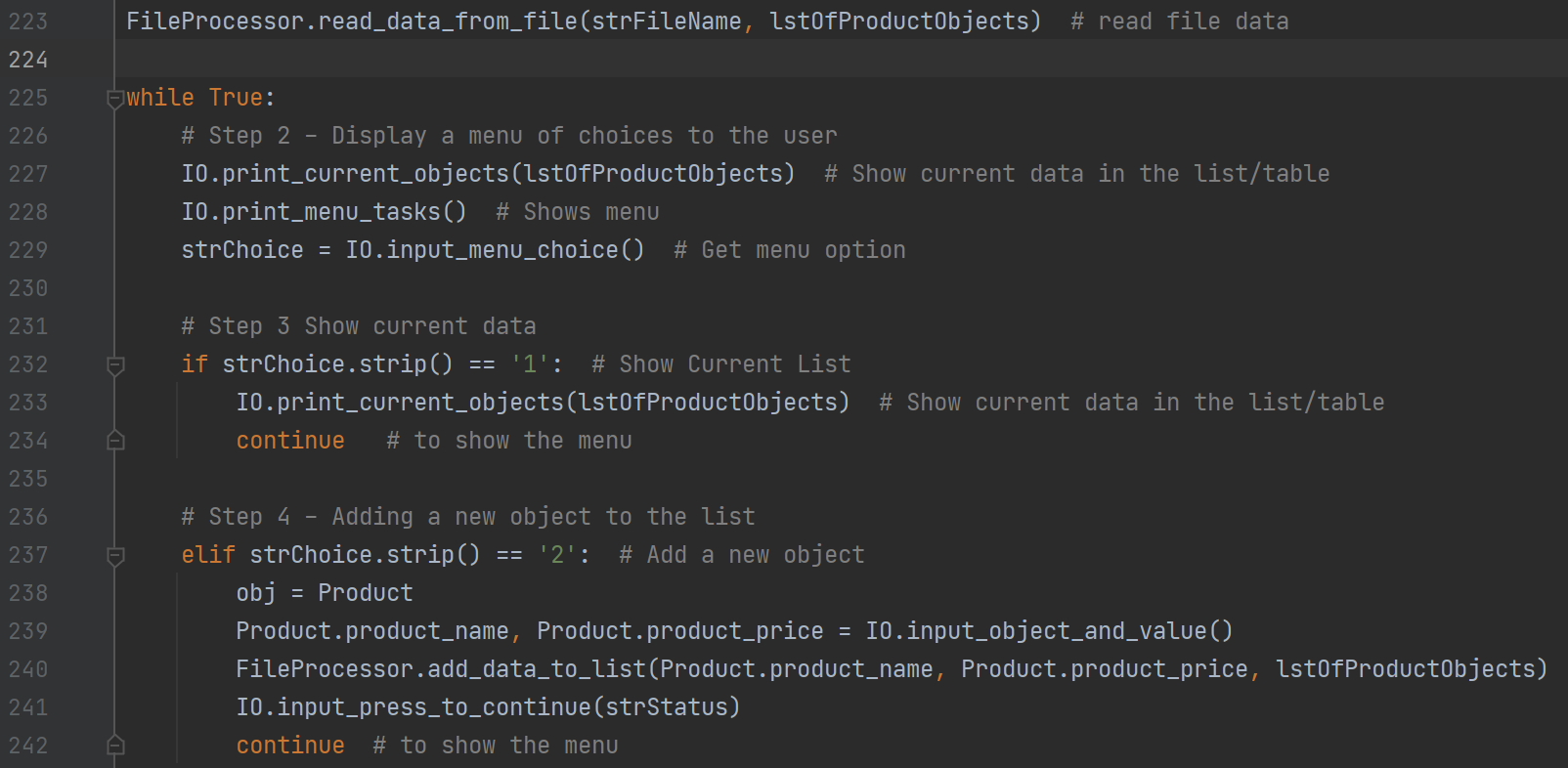
***Figure 13: Print Current Objects and Yes/No***

1. The last two methods ask for the users input on the product name/value and then hitting enter to continue the code. This is the last of the methods within the IO class. (Figure 14)



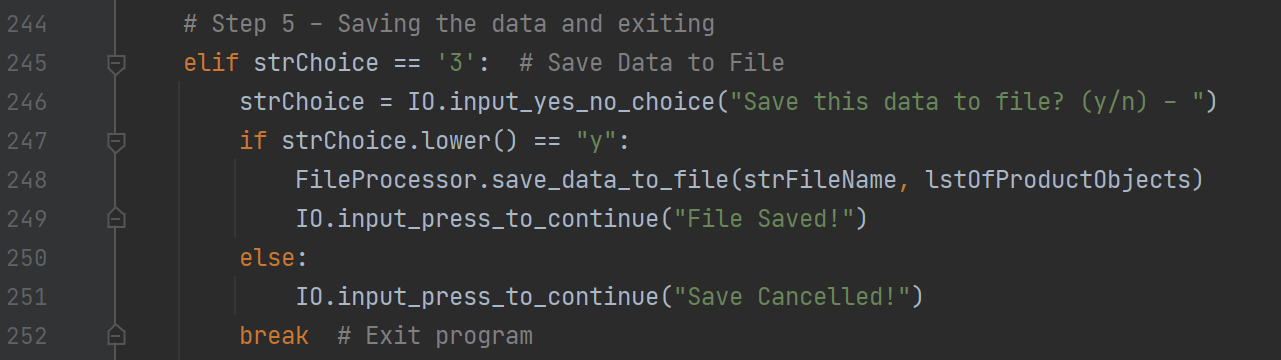
***Figure 14: Object Information and Press to Continue***

1. In the main body of the script, it first starts with using the read-data function to go through the text file and create the starting list of product objects. The menu is then printed and the user is asked to select which option to use. If they select option 1, use the IO class to print the list of product objects. If the user selects 2, collect the new product name and value and add that to the list of product objects. (Figure 15)



***Figure 15: Main Script***

1. The last bit of the main body script saves the data to the text file. The user is asked to clarify if they want the data saved to the file and if yes, the data is saved. (Figure 16)



***Figure 16: Saving the Data and Exit***

# **Summary**

In this assignment we are tasked with creating a script that uses classes to manipulate data including both the products name and value. This is similar to previous weeks in that we are working with data and saving it to a text file. This was a good method of learning elements of a class, although at this point there are several parts of classes that can get a little confusing still. If I were to do this assignment again,I would spend more time with the setter/getter elements of the product information.