Shruthi Srivatsan

Dec 5, 2020

Assignment 08

Python Script with Classes

**Introduction:**

In Assignment 8, I was given a script that uses classes and I modified it using the pseudocode that was in the script. Each of the attributes have a dot notation and I have gone in detail why or how I wrote the following script. There are previous concepts that are addressed in this Assignment.I am going to go over the parts that I was asked to write code for rather than discuss all the pre-written parts of the assignment.

First I had to define my variables. I defined the file that was going to be created, the list of product obkects, input product, price and str choice and status. In the next section for the class product I was able to define the properties for class product for the product\_name and product\_price. Doing so will allow you to use these further along in your code as it has a definition to refer back to. This was already given in the pre written script and is shown in Figure 1.

Text

Description automatically generated

Figure 1: Pre- Written Script and Definitions

The class file processor part of the script was to define the functions that will process data to and from a file and list the product objects. The methods for that were defined under that section, which can be seen in Figure 2. This was already pre-written but again an important part of the script so I am including it in this write up.

Graphical user interface, text

Description automatically generated

Figure 2. FileProcessor – Pre-Written Script

Now in this next section I have a static method for read\_ data\_ from\_ file, write \_data\_ to\_ file, and add\_ object\_ to\_ list. This can be seen in detail in figure 3 however if I want to summarize what each of these do then I can start with Read\_data\_ from\_file. This function basically reads the data from the file into a list of objects I will start with a try except clause and have code that will list the product objects and open the file name that is defined in the definition section above and read the file. I also have a loop-> for line in file-> that functions when we are splitting each line and assigning each field to the variables and basically having it in the format of item, price. Write\_data\_to\_file Will basically open the file and write in it and for every object in the list it will concatenate the object's name and then list and write it to the file. Once that is done the file will close. Add\_object\_to\_list basically uses the item and price parameters to create the object and add it to the list. The code that I wrote will basically list the product objects in terms of the item and the price and basically return a message saying that the item was added to the user. This code can be seen in Figure 3 below.

Text

Description automatically generated

Text

Description automatically generated

Figure 3. Showing all @staticmethod functions for adding object to list, writing and reading data

In the next section of the code I used this class IO: to basically name out the methods that I will be using in the following section that included: print\_menu\_products():,input\_menu\_choice():input\_yes\_no\_choice(),get\_current\_data\_from\_list(), input\_product\_data(),input\_press\_to\_continue(optional\_message=().

I used this section to print out the menu options with my first static method function. then I put it static method statement for inputing the menu choice, user choice of yes or no, getting current data from the list, and then in putting new data to the list, and pressing enter to continue The code for these can be found in Figure 4.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Figure 4: Menu options staticmethod code

After this section we come to the main body of the script where I wrote code in order to load data from a file. once the computer loads the file that you wanted to read from which was previously defined in your code above. once the file opens it should then show the user a menu of options an get users selection if the user selects one then it should show the user the current data from the list. If the user selects two it should let the user add data to the list of products it would prompt the user to input a product and the price and then click continue. If the user selects 3 then it should immediately ask a question to the user stating whether or not they want to save this data to the file yes or no. Once the user selects Y for yes then it should give a successful goodbye indicating that the data was saved. If there are issues with the data being saved then there is an else statement in the code that should display save cancelled and loop back to the menu so that the user can select any of the options from the menu again. The last option would be for user to select for which would automatically print a goodbye statement and end the code. These statements were all written in the format just like strChoice.strip(). You will be able to see the detailed code in figure 5.

Text

Description automatically generated

Text

Description automatically generated

Figure 5 Providing user options to select from in menu

Finally, I will show you how this code looks when you run it in py charm and in terminal. figure 6 will show you how it shows up in py charm and figure 7 will show you how it shows up in terminal.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Figure 6. Running code in pycharm

Text

Description automatically generated

Figure 7. Running the file in Terminal

**Conclusion:**

In conclusion I was able to focus on learning about classes and creating objects from them. I was able to write functional code using attributes and constructors and I was able to successfully complete this assignment. Although this uses a similar menu to previous assignments the concept of building a working code using classes was new and challenging. Using the class videos and notes I was able to complete this successfully.