### Kirsten Elghoroury

### Foundations of Programming: Python (IT FND 110B)

### Assignment 08

### March 9, 2021

GitHub Page: <https://github.com/kirstencodes/IntroToProg-Python-Mod08>

Organizing Complexities: Classes and Objects

This week I learned about how to work with classes and object classes. These are very useful for organizing data and processing. Though Python is an object-oriented language and most things are objects, this week I learned how to initialize an object class which is basically a blueprint for any number of instances of that object.

# Classes

**Classes** are a great way to organize and reuse data and functions. Typically, developers will use classes for data or processing. To make a new class, we simply use the keyword ‘class’ followed by the class name and optional parameters. Functions in classes are referred to as **methods** and data (variables/constants) in classes are referred to as **fields**. You can use classes to create new objects, like a blueprint.

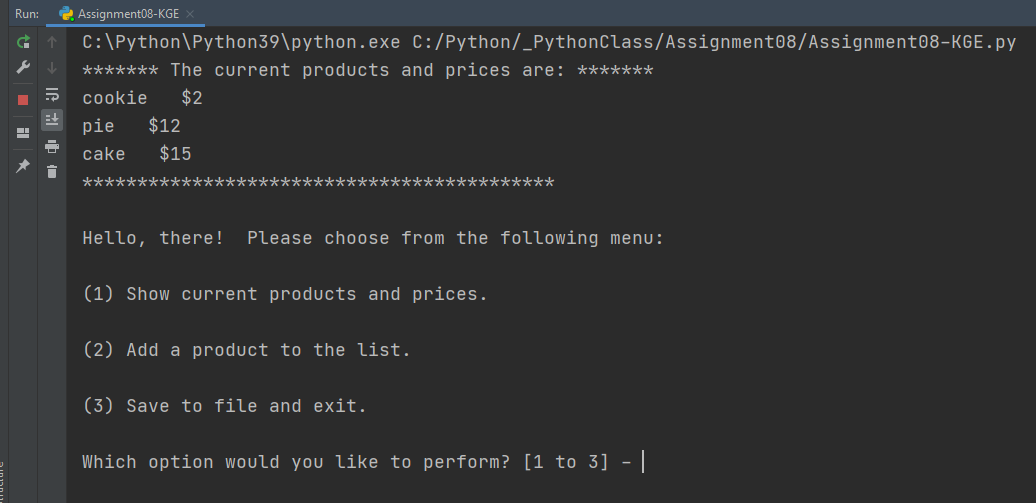
# Object Class

To create an **object** using a class, we use the **\_\_init\_\_() function** explicitly (there is always a built-in \_\_init\_\_ function in any class). This object class acts like a blueprint for any number of objects that can be defined using the same parameters. You can also create methods to manipulate or process the object in the same way. Let’s say we had an object class that we created that was a blueprint for a book. Each book could have a genre, a number of pages, a number of chapters, and the number of books sold as of a certain date. Each new book object could be defined in a similar way and you could even automatically calculate the average number of pages per chapter of each book within the class. Each book title is another instance of that object.

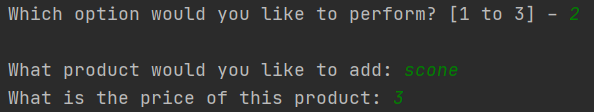
The **self** keyword is a parameter that is used when constructing an object. Even though this is used similarly to a parameter, you do not pass arguments to this parameter. It is essentially a reference to the current instance of the class.

# Assignment

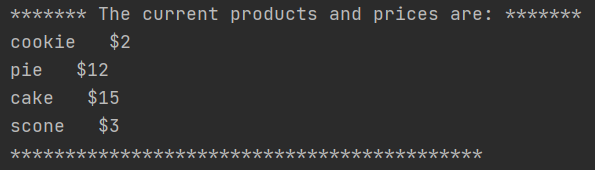
The assignment this week was to edit the starter script to create a program that has a menu of options including showing current list of products (and prices), adding to the list, and saving and exiting. The list of products was created in an object class that stores data about each instance of the object, the product and price. The list is pickled while saving to the text file when the user saves and exits.



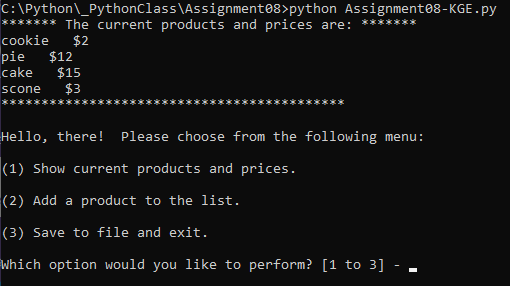
***Figure 1: In PyCharm, displaying current products (baked goods!) and displaying the menu of options.***



***Figure 2: In PyCharm, adding a product to the current products.***



***Figure 3: In PyCharm, the new current list of products.***



***Figure 4: In the command line, the program still works as expected.***

# Summary

This week, I learned how to more effectively use classes, organize my code, edit someone else’s code, and create an object class. I also used the pickle module from last week again. The main body of the code in the assignment is only four lines because most of the data and processing is handled within the fields and methods in the classes. As code gets more complex, you can see how this would be useful especially for objects that get reused in different parts of the code.