Kevin Scales

Mar 11, 2021

Introduction to Programming (Python)

Assignment 8

Working with Objects

# Introduction

The main topics for this assignment are objects and the classes that come from them. In many ways, at least at this point, objects don’t introduce much functionality that can’t be done with functions and the various data types we have already encountered, but they offer a ‘natural’ way to look at things that is conceptually straightforward, even if the syntax is at first different and challenging. The typical example is that you could have an object of the Human class, i.e. a person, and can obviously have more than one of them, each with unique information of the same repeated sorts of data.

# The Python Code

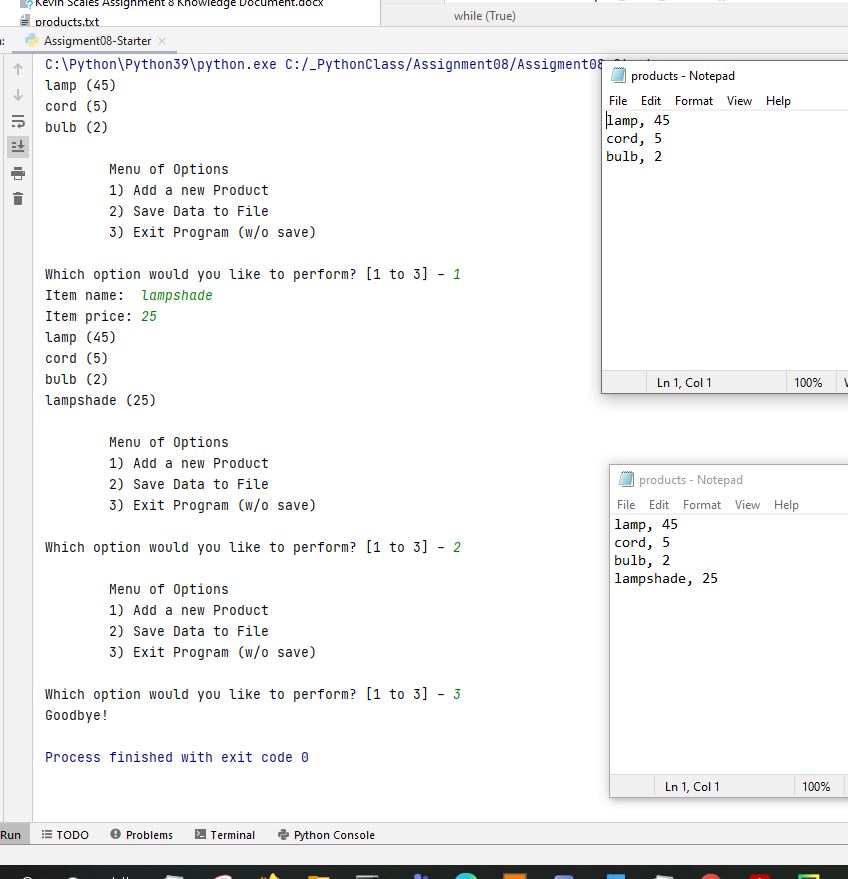
We used three classes, two being objects, each with two or more defined functions. In many ways the overall structure was similar to that last seen in assignment 6, being a menu of choices and ways to define functions to open and close files for reading and writing, add data, display data, and with exception handling in case of errors. Meaningful names, like IO, FileProcessor, and Product continue to aid in program readability.

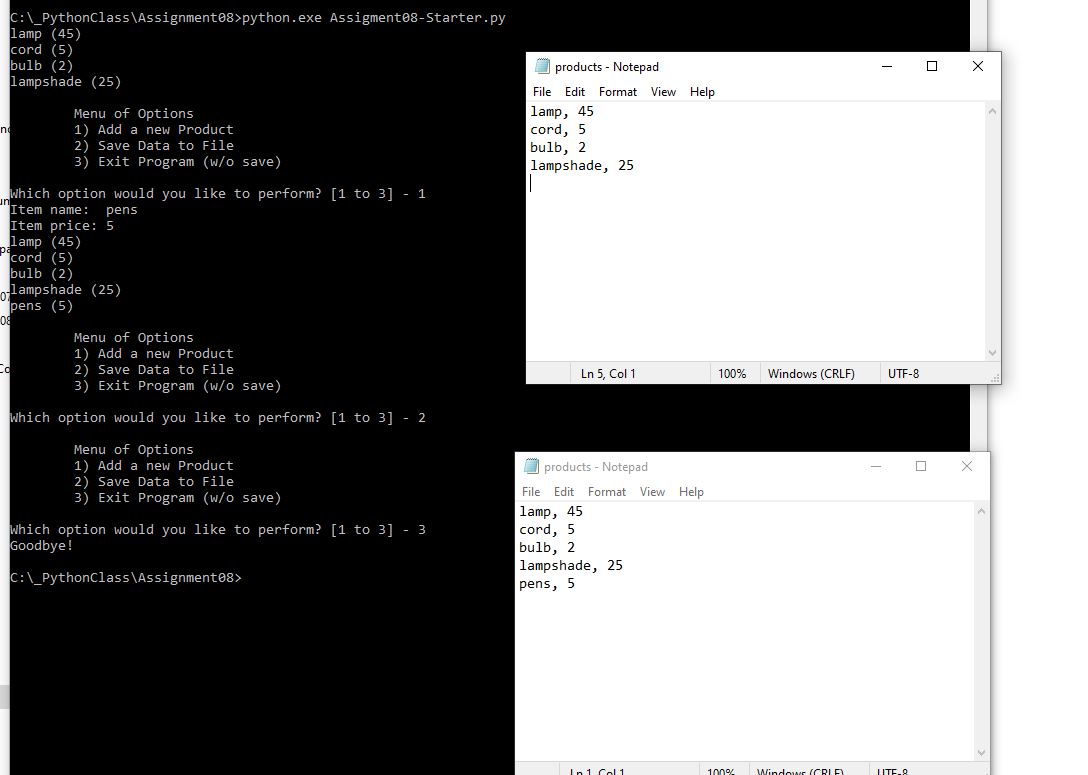
The Product class, which is an Object, displays some of the new concepts, with an \_\_init\_\_ defined to give new objects the basic information for that object. This is the constructor method (or initialization method). We can do what we want with this method, but as it is called immediately, it should be something the object can actually do immediately. The self parameter is also introduced here as a way of referring to an object by that object.

Static methods are more fully explained in this section. They don’t contain a self in the parameter list. More specifically, it need not refer to an object of the class type, just a property of the class. With the example above of Humans, a static method might tell us the population of Humans that exist, but it is not a property of a particular human. The same is the case with Python objects, there are individual properties (even if all instances must have some value for it that could be a unique value), and there are aggregate properties like population.

# Results

The combinations of this program are not extensive, and can be captured in two screenshots, one for PyCharm and one for the console window. In order:





# Conclusion

The greater lesson of the chapter this time were not just how to use objects but how objects and classes and their associated properties add to the capabilities we’ve seen while also duplicating many of them. This is probably why object-oriented programming is a thing; we can use objects for routine operations but also to open up new capabilities.

As a meta-lesson, not really part of the assignment, I re-learned the value of re-using code, of keeping it simple and incremental, and not overthinking things. Though there is a lot of new material, I’m reminded that when I programmed professionally (in C, without an object to be seen anywhere), I didn’t try to re-invent the wheel each time, and if I’d followed along the guides created in the last several assignments, things probably would have been smoother.