Mercedes Gonzalez Gonzalez

August 30th, 2020

IT FDN 110 A

Assignment 08

GitHub Repository: <https://github.com/mglezglez/IntroToProg-Python-Mod08>

GitHub Website: <https://mglezglez.github.io/IntroToProg-Python-Mod08/>

WORKING WITH CLASSES IN PYTHON:

# Introduction

This week, as part of Module 8 in Foundation of Programming (Python) course, I learned how to work with classes in Python. This is the first step into Object Oriented Programming. As part of this assignment, I combined the knowledge I learned in this section of the course about classes, constructors, properties and methods with data serialization and exception handling to create a script that interacts with users to create, display and save a list of products. This document contains a step-by-step description on how the script was created and tested.

# Modifying the script template

In order to implement this program, I re-used a script template provided for this assignment and I proceeded to write the code in each section as indicated in the comments in order to get the main program to work.

The script has four main components:

* The Product class
* The FileProcessor class
* The IO class
* The Main Program

The following sections briefly explain each component.

## The Product Class

The Product class contains two private attributes, which are encapsulated with getter/setter properties. The setter methods implement data validation and error handling. That is, if the value provided is valid, then the private attributes are set accordingly, otherwise, an exception is raised to indicate the value provided is not correct. This class also override the \_\_str\_\_() method in order to customize how Product objects are converted into strings.

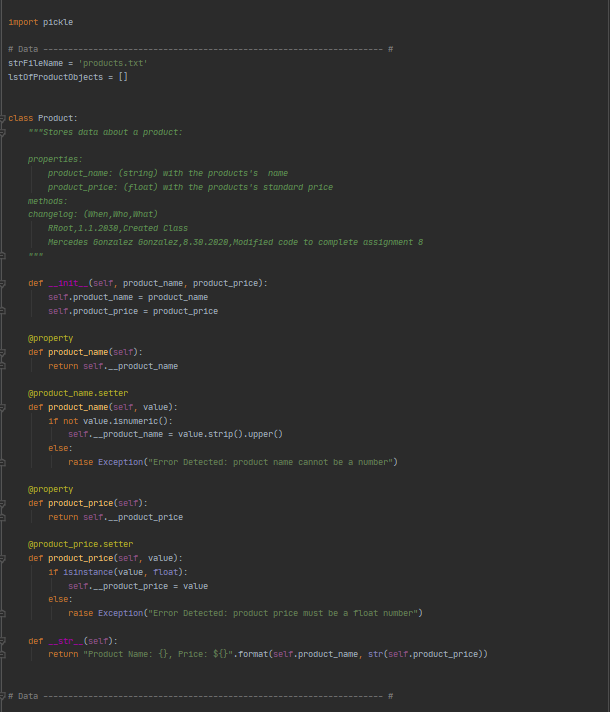


Figure 1. Product Class

## The FileProcessor Class

The FileProcessor class leverages the Pickle module in Python to serialize (save) and deserialize (read) a list of Product objects into a file. Error handling is added as well in these methods in order to make sure that any exception raised while saving or reading data is handled gracefully.

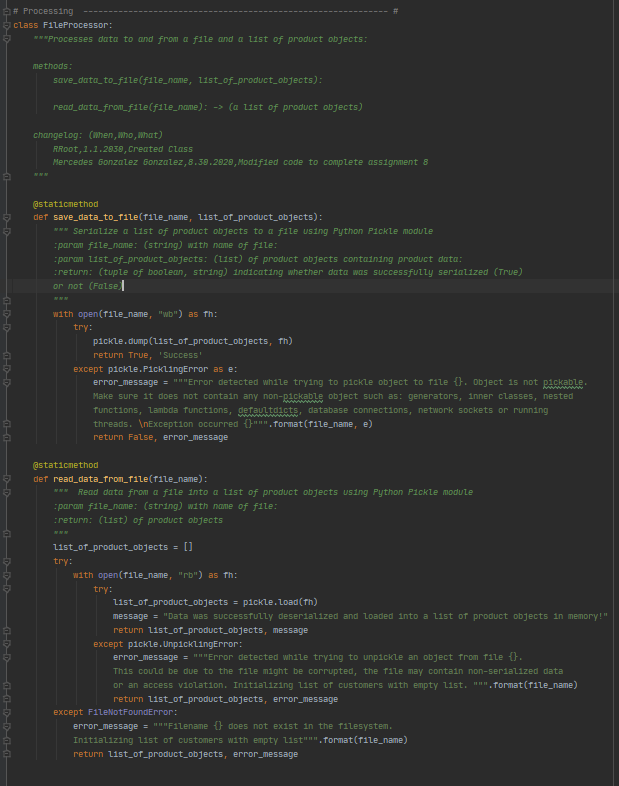
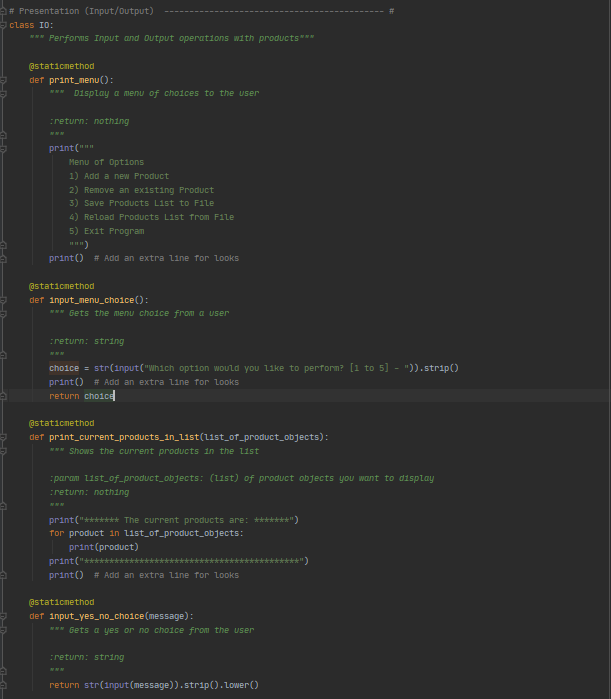


Figure 2. FileProcessor Class

## The IO Class

The IO class handles all Input/Output operations with the user, such as, printing the menu of options, the list of currently added products, asking the user to select one of the options, displaying messages on the screen and asking for confirmation to proceed, as well as handling the addition and removal of products to the list.



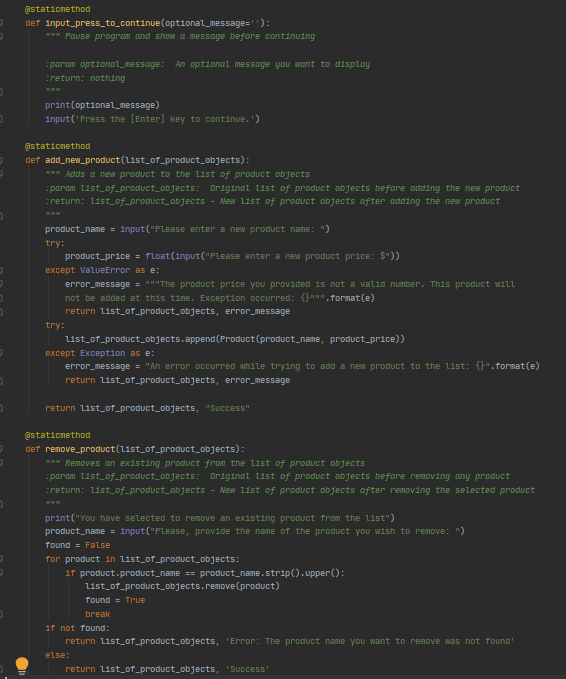
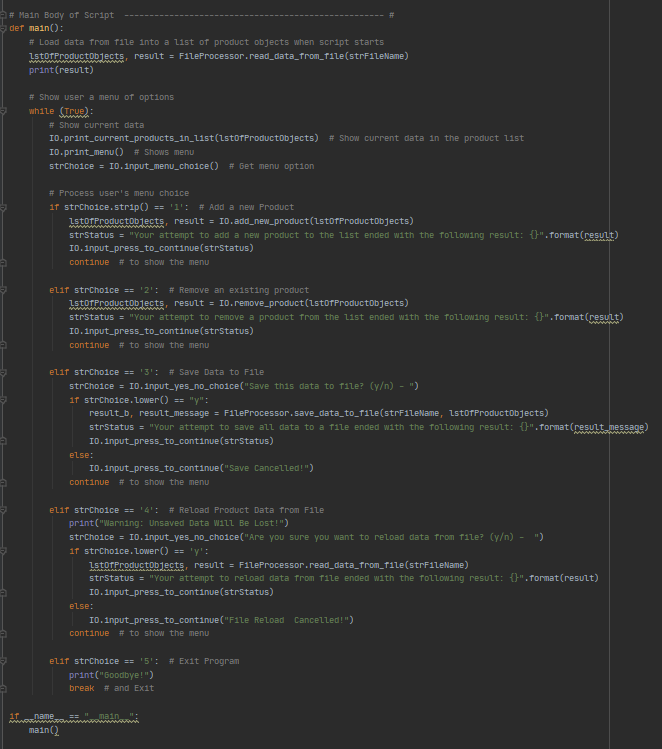


Figure 3. IO Class

## The Main Program

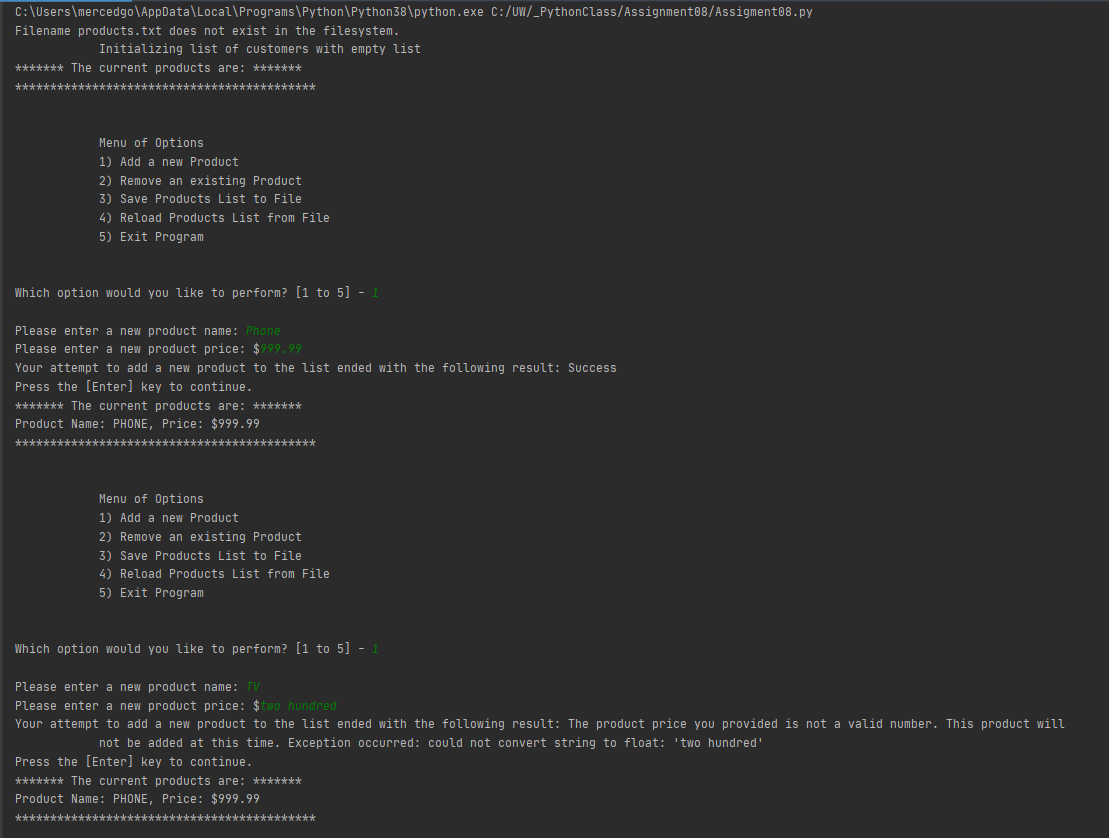
The Main program focus on a set of task that leverage the functionality already implemented in each of the classes presented above. For example, the Main program uses the static methods in the IO class to print the menu of options to the user, the list of added products and obtain a selected choice. Depending on the choice selected, it leverage the static methods in the IO class to add or remove a product, as well as the static methods in the FileProcessor class to save the list of products to a file or reload the list of products from the file back to memory. Every product is represented with an object instance from the Product class, so each product has a product name and a product price.

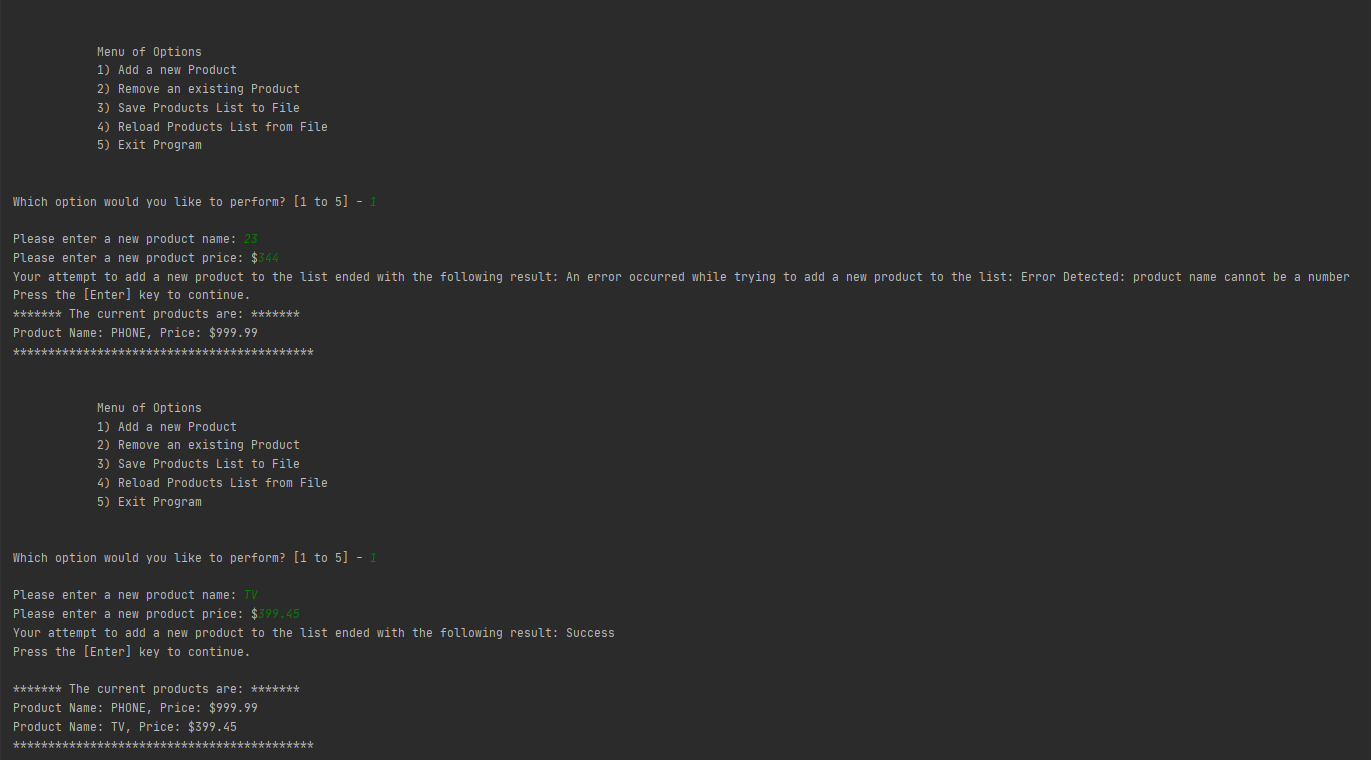


# Testing the script

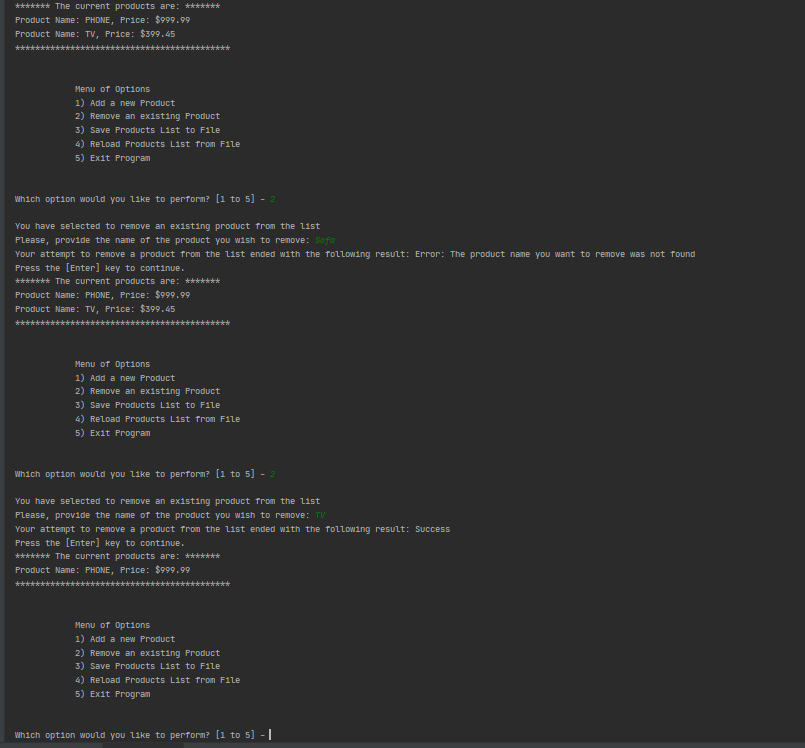
The following section contains screenshots from the program while running.

Test # 1: Adding products

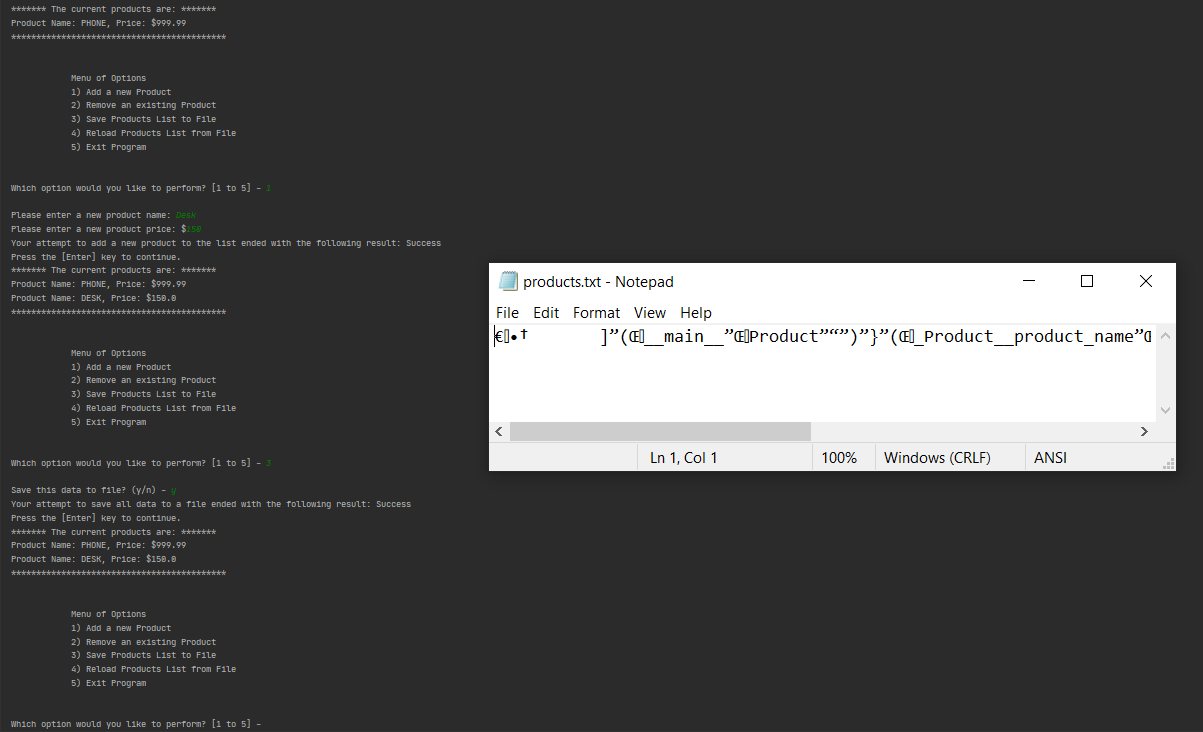




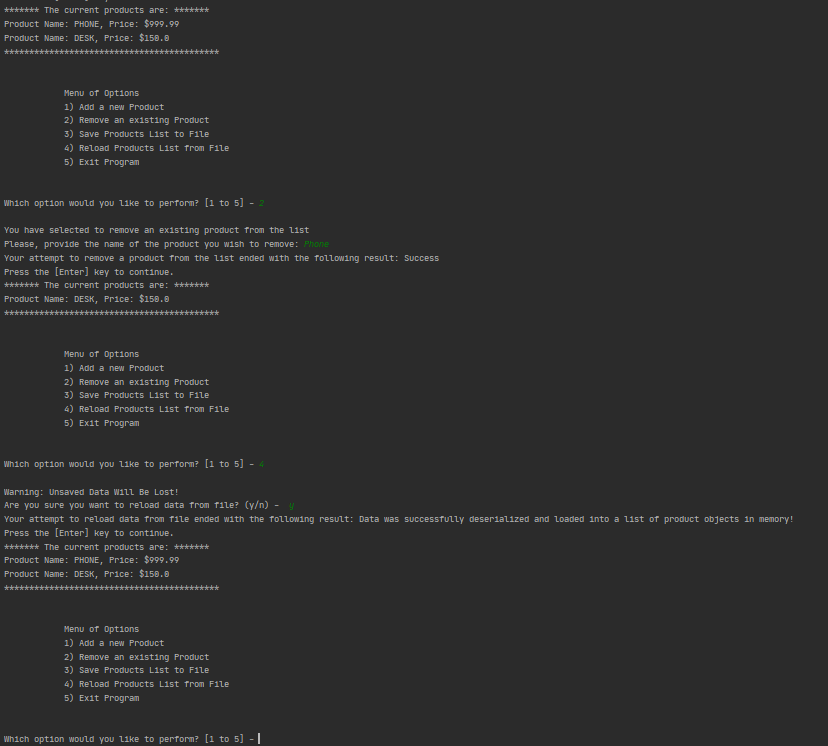
Test # 2: Removing products



Test # 3: Saving list of products to file



Test # 4: Re-loading list of products from file to memory



# Summary

This module introduced working with classes, which are one of the main components of Object-oriented programming (OOP). OOP is a programming paradigm that provides a means of structuring programs so that properties and behaviors are bundled into individual objects. Classes are used to create user-defined data structures. Classes define functions called methods, which identify the behaviors and actions that an object created from the class can perform with its data.