Dominic Ferenczy  
June 1, 2021  
Foundations of Programming: Python  
Assignment 08  
[uwferenczy/IntroToProg-Python-Mod08 (github.com)](https://github.com/uwferenczy/IntroToProg-Python-Mod08)

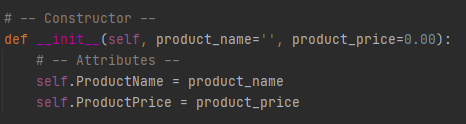
Classes and Objects

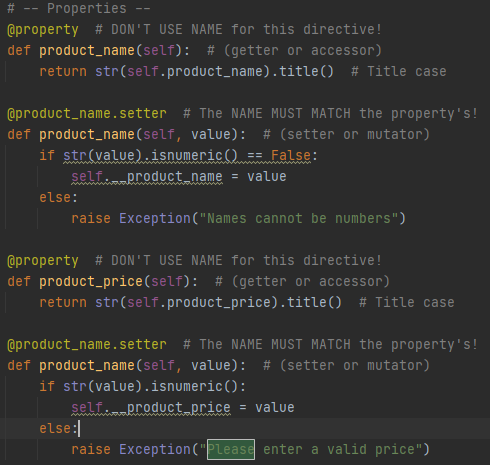
# Introduction

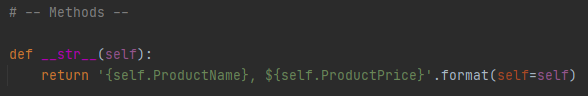
This document details how I applied what I have learned in module 8 of the Introduction to Programming class to improve the readability and portability of my code using classes and objects.

# Classes

Classes can be used to define object data structure and behavior to quickly generate objects using the constructor definition. The constructor identifies what attributes are required to initialize a new object of the class and the default values of each if none are provided. For this assignment, I used product\_name and product\_price, setting the default value of name to a blank value and the default product price to $0. Classes use self to refer to data that exists in the object instances but not directly in the class definition. The resulting code can be seen in figure 1.

  
Figure 1: PyCharm showing the class constructor  
  
After defining the constructor, I created setter and getter properties using the variables that are passed into the class (product\_name and product\_price). The getter property defines how the class behaves when the corresponding property is accessed to read / get the data from an object instance. In the case of this application, I used it to return the variables in title case. The setter property defines how the class behaves when the corresponding property is accessed to write / update the data. In the case of my application, I used it to check if the variable was numeric or not and raise an exception if needed. The resulting code can be seen in figure 2.

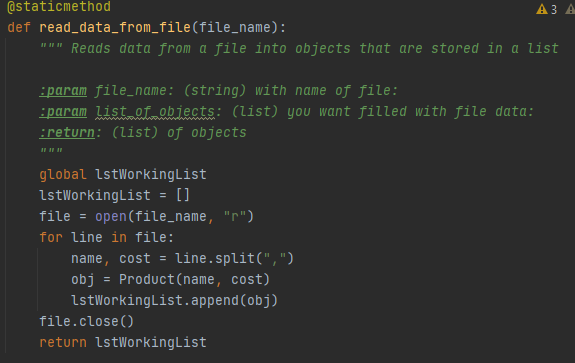
  
Figure 2: PyCharm showing the properties of Product Class  
  
The \_\_str\_\_() method can be updated to modify and define how the object string displays when it is called. In the case of this application, I separated the Product name and cost with a comma and added a dollar sign in front of the cost. The resulting code can be seen in figure 3:

  
Figure 3: Pycharm showing \_\_str\_\_ method definition

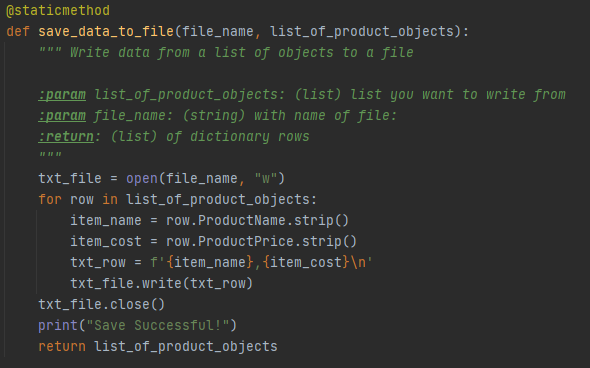
I called this method later in the application during the print option. The starter file for this assignment also included a FileProcessor class. I leveraged the code I have written in past assignments to create the read\_data\_from\_file and save\_data\_to\_file methods and updated them to utilize objects using the Product method.

# Objects

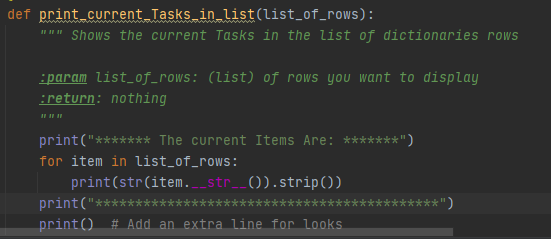
Objects are copies of classes using their constructors. They are useful when you need to duplicate and enforce data structure. In this case, I created a new object for each line in the text file by parsing the data into variables for name and cost and passing them into each object. I then appended the objects into a list and closed the file. The resulting code can be seen in figure 4.

  
***Figure 4: PyCharm showing objects being initialized using the Product Class***

I updated the save\_data\_to\_file method to access the getter function of each object in the list and write them into rows in the data file. The resulting code can be seen in Figure 5.

  
Figure 5: PyCharm showing Objects being written to file

I also updated the print\_current\_Tasks\_in\_list method to use the \_\_str\_\_() method from the Product class by printing it for each object in the list and stripping the break lines from them (to avoid duplicate lines). The resulting code can be seen in Figure 6.

  
Figure 6: PyCharm showing the \_\_str\_\_() method being called from the IO class

# Summary

This document captured what I learned in module 8 of the class. I documented the steps I took to use classes to use pickling in a Python script, how to improve the readability and portability of my code using classes and objects.