Haley Barsa
August 31, 2022
IT FDN 110 B Foundations of Programming: Python
Assignment08
https://github.com/hbarsa/IntroToProg-Python-Mod08

# Using Classes to Collect and Save Product Data

## Introduction

This document presents the steps taken to collect and save product data using custom classes. Pseudo code is provided and the task is to complete the program. PyCharm is used to edit and test the script. IDLE is also used to test the script. Data classes, constructors, properties, and processing classes are discussed in the following sections.

## Data Classes

A data class is used to store data about a subject or event. Instance objects are used for data classes to create multiple copies of specific data. An instance object, also called an object, is a copy of the class with actual values.

### Constructor

A constructor is a special function which automatically runs when you create an instance object. The constructor below in Figure 1 is used to set the initial value. The self keyword must be included as a parameter to be used from an object instance. The self keyword is used for data classes and not processing classes.

```
# -- Constructor --
def __init__(self, product_name, product_price):
    self.product_name = product_name
    self.product_price = product_price
```

***Figure 1: Constructor***

### Properties

Properties are functions that manage attribute data. They interact with attribute fields to get, set, or delete data. It is fitting that two common properties are called “getters” and “setters”. The first function in Figure 2 is the getter property which is used for formatting data. The second function in Figure 2 is the setter property which is used for error handling. The setter function controls if a variable is set or a flag is raised to the user.

```
# -- Properties --
#Product Name
@property
def product_name(self):
    return str(self.__product_name_str).title() # Formatting data in title case
```

```
@product_name.setter
def product_name(self, value):
    if str(value).isnumeric() == False:
        self.__product_name_str = value
    else:
        raise Exception("Product names cannot be numbers")
```
***Figure 2: Properties***

## Processing Classes

A processing class is used to process data, like read or write data to a file. Static methods are used for processing classes so that functions can be used directly through the class, without a copy. A static method is a decorator placed at the beginning of a function to call methods directly from the class, without making an instance object. The code in Figure 3 below is used to write data to a file. The function can be called later in the code by drilling into the class and using FileProcessor.save_data_to_file(). “File Processor” is the name of the class and “save_data_to_file” is the name of the function.

```
@staticmethod
def save_data_to_file(file_name, list_of_product_objects):
    file = open(file_name, "w")
    for row in list_of_product_objects:
        file.write(str(row.product_name) + "," + str(row.product_price) + "\n")
    file.close()
```
***Figure 3: Processing Code to Save Data***

The code below in Figure 4 shows a function to input new data. An object instance is created with the text, “objP1 = Product('',0)”. The inputs from the user are saved within the object.

```
@staticmethod
def input_new_product_data():
    """  Gets name and price values to be added to the list

    :return: (string, string) with name and price
    """
    objP1 = Product('',0)
    objP1.product_name = str(input("What is the name? - ")).strip()
    objP1.product_price = float(input("What is the price? - "))
    print()  # Add an extra line for looks
    return objP1
```
***Figure 4: Processing Code to Input New Data***

## Code Demonstration

The code demonstrates how data can be gathered from the user and saved to file. See screenshot in Figure 5 of the program running.

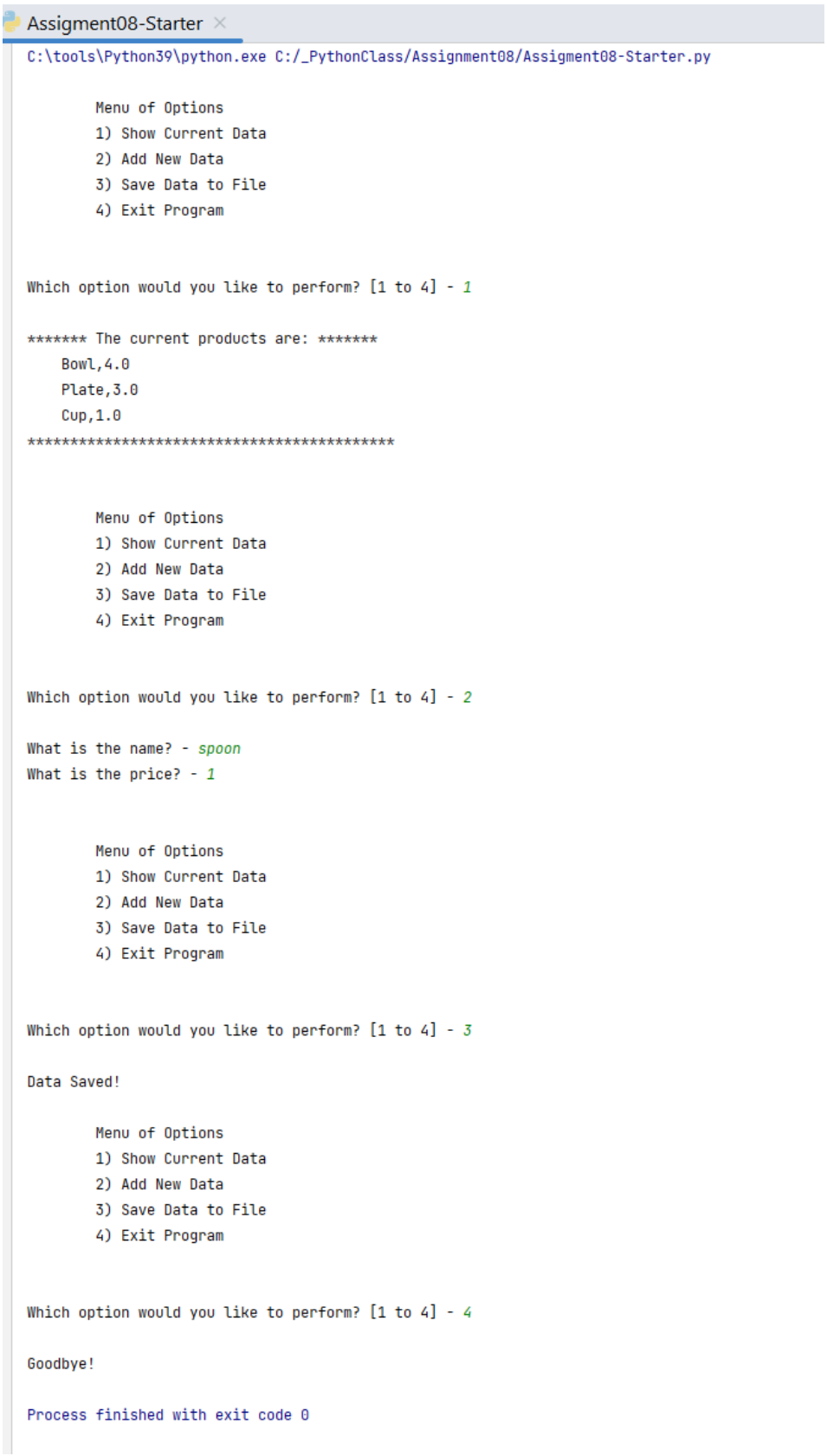


***Figure 5: Script Running From PyCharm***

Figure 6 below shows the text file contents.

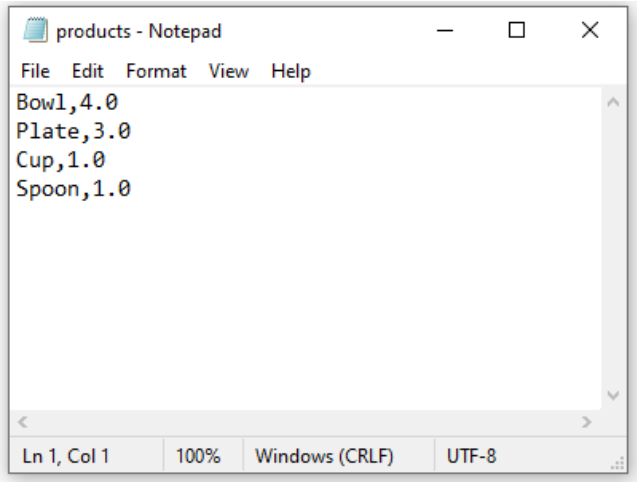


***Figure 6: Text File Contents***

Figure 7 below shows the script working in a shell window.

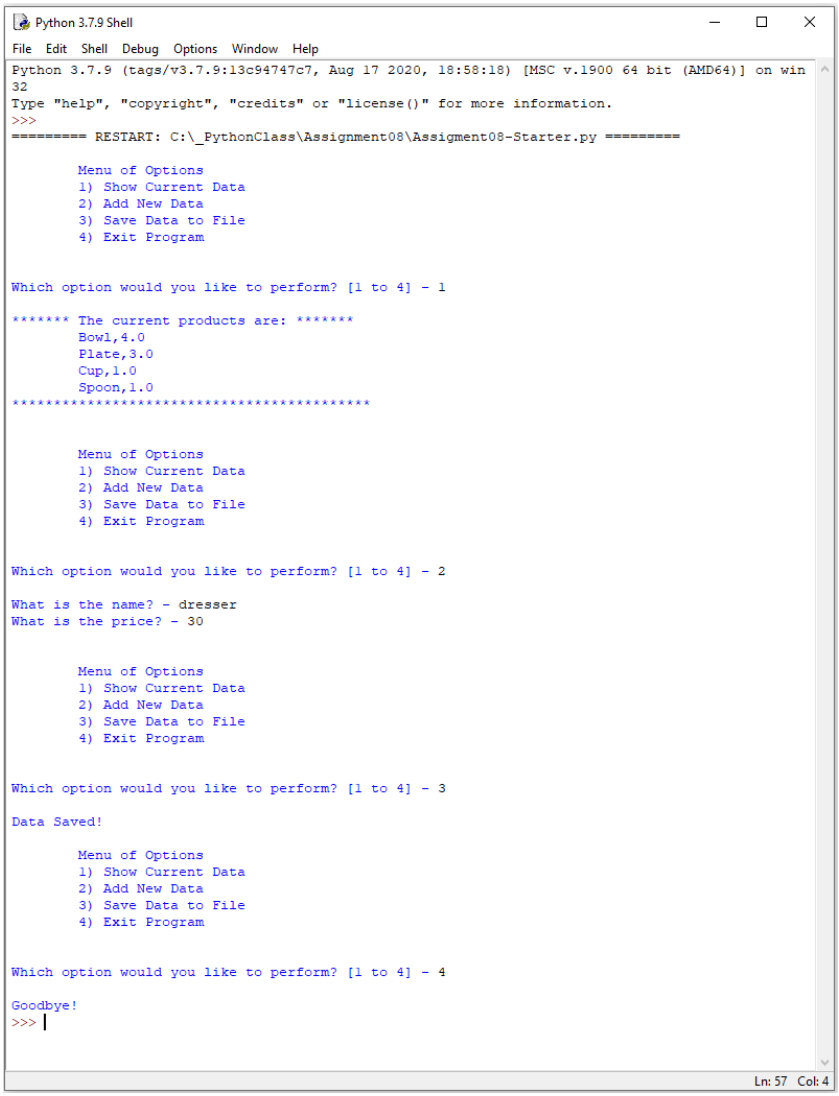


***Figure 7: Script Running From a Shell Window***

## Summary

This assignment provides the opportunity to use custom classes and the standard class pattern to collect data from a user. Both data and processing classes are used. Error handling is included to ensure the correct information is inputted from the user. Like previous assignments, a menu of options is provided to the user and several tasks can be performed.