

Glenn B. Dacones

June 7, 2022

Foundation of Programming: Python

Assignment 08

[My Github Repository](#)

# Product & Price Using Classes and Objects

## Introduction

In this paper, I am integrating what we have learned in the past six weeks. Using the template, with pseudocodes provided, I will create a script using the user inputs e.g., product and price and store that information into a list, and eventually writing it to a text file. I will use classes and objects to accomplish this task. I will also create a new Github repository to store the data used in the assignment.

## Preparing the Data

Using the template, I started with the class Product to define and store the data using the function initialize\_data() (Figure 1).

```
class Product:
    """Stores data about a product:

    changelog: (When,Who,What)
    RRoot,1.1.2030,Created Class
    Glenn B. Dacones,6.6.2022,Modified code to complete assignment 8
    """

    @staticmethod
    def initialize_data(file_name=strFileName):

        lstOfProductObjects =
        [{"Product": "Television", "Price": "499.99"}, {"Product": "Table", "Price": "99.99"}]

        file = open(file_name, "w")
        for row in lstOfProductObjects:
            file.write(row["Product"] + "," + row["Price"] + "\n")
        file.close()
```

**Figure 1. An example of defining and storing the data using a function.**

## Processing the Data

In this section of the script, a class FileProcessor is provided in the template. I added several functions to complete this section. Shown below is one of the functions I created to read the data that was prepared from the previous class, Product (Figure 2).

```

@staticmethod
def read_data_from_file(file_name, list_of_rows):
    """ Reads data from a file into a list of dictionary rows

    :param file_name: (string) with name of file:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """

    list_of_rows.clear() # clear current data
    file = open(file_name, "r")
    for line in file:
        product, price = line.split(",")
        row = {"Product": product.strip(), "Price": price.strip()}
        list_of_rows.append(row)
    file.close()
    return list_of_rows

```

**Figure 2. An example of reading the data from a file using a function.**

## Presenting the Data

In this section of the script, a class IO is provided in the template. I added several functions to complete this section. Shown below is one of the functions I created to display a menu of choices to the user (Figure 3).

```

@staticmethod
def output_menu_products():
    """ Display a menu of choices to the user

    :return: nothing
    """

    print('''
    Menu of Options
    1) Add a new Product
    2) Remove an existing Product
    3) Save Data to a File
    4) Exit Program
    ''')
    print() # Add an extra line for looks

```

**Figure 3. An example of displaying a menu to the user using a function.**

## Exception Handling

In several sections of the script, I added exception handlings to catch invalid data and to prevent the script of terminating the program. Below is an exception handling to catch an invalid price for the product (Figure 4).

```

# This exception handling will catch an invalid price for the product
try:
    price = float(input("Enter a Price: "))
except:
    print("You entered an invalid price! Please try again.")
    price = float(input("Enter a Price: "))

```

**Figure 4. An example of exception handling for an invalid price.**

## Executing the Script

Shown below is the script running in PyCharm. Note the exception handling when an invalid price is entered (Figure 5).

C:\Users\Charmaine\AppData\Local\Programs\Python\Python310\python.exe  
C:/\_PythonClass/Assignment08/Assignment08-Starter.py

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

\*\*\*\*\*

#### Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 1

Enter a Product: Chair

Enter a Price: 49.99

The chair has been added to the list/Table!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

\*\*\*\*\*

#### Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 1

Enter a Product: lamp

Enter a Price: 5a

You entered an invalid price! Please try again.

Enter a Price: 19.99

The lamp has been added to the list/Table!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

Lamp (19.99)

\*\*\*\*\*

Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 2

Which product would you like to remove?: lamp

The lamp has been removed from the list/Table!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

\*\*\*\*\*

#### Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 3

Data Saved!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

\*\*\*\*\*

Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 4

Goodbye!

Process finished with exit code 0

**Figure 5. A screenshot of the script running in PyCharm.**

Shown below is the script running in command shell (Figure 6).

Microsoft Windows [Version 10.0.19044.1706]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Charmaine>cd c:/\_pythonclass/assignment08

c:\\_PythonClass\Assignment08>python.exe Assigment08-Starter.py

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

\*\*\*\*\*

Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 1

Enter a Product: chair

Enter a Price: 49.aa

You entered an invalid price! Please try again.

Enter a Price: 49.99

The chair has been added to the list/Table!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

\*\*\*\*\*

Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 1

Enter a Product: lamp

Enter a Price: 19.99

The lamp has been added to the list/Table!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

Lamp (19.99)

\*\*\*\*\*

Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 2

Which product would you like to remove?: lamp

The lamp has been removed from the list/Table!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

\*\*\*\*\*

Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

Which option would you like to perform? [1 to 4] - 3

Data Saved!

\*\*\*\*\* The current products are: \*\*\*\*\*

Television (499.99)

Table (99.99)

Chair (49.99)

\*\*\*\*\*

Menu of Options

- 1) Add a new Product
- 2) Remove an existing Product
- 3) Save Data to a File
- 4) Exit Program

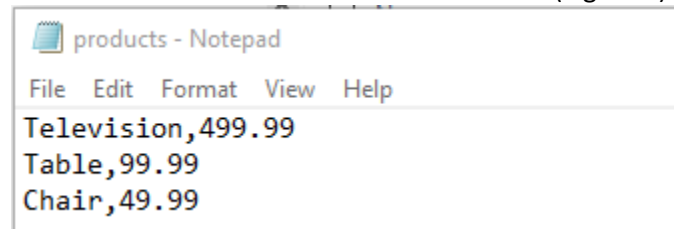
Which option would you like to perform? [1 to 4] - 4

Goodbye!

c:\\_PythonClass\Assignment08>

**Figure 6. A screenshot of the script running in the command shell.**

Shown below is the data written to a text file (Figure 7).



**Figure 7. An example of data written to a text file.**

## Summary

In this assignment, I have incorporated what I have learned so far in the past six weeks. With the provided template and pseudocodes, I was successful in adding the functions and objects in the predefined classes.