

**Name: Wei Wang**

**Date: 8/31/2022**

**Course: IT FDN 110 B Su 22: Foundations Of Programming: Python**

**GitHub repository:**

<https://github.com/vivian39/IntroToProg-Python-Mod08>

## **Assignment08: Objects & Classes**

### **Introduction**

This module introduces classes, the components inside of them, and how to use them. I will also download, install, and use GitHub Desktop.

### **Create a Python Script (in a Mac book) a. Create a Folder**

**a.** Create a new sub-folder called Assignment08 inside the \_PythonClass folder.

**b.** Create a new Project in PyCharm

Create a new project in PyCharm that uses the \_PythonClass\Assignment08 folder as its location

**c.** Add Code to the Script

The starter code in the file includes pseudo-code. Read and understand the pseudo-code, then add code to make the application work.

**d.** Run the Script

Run the script in BOTH PyCharm and an OS command/shell window and capture images of it working on your computer.

### **Code:**

```
# ----- #  
# Title: Assignment 08  
# Description: Working with classes  
# ChangeLog (Who,When,What):  
# RRoot,1.1.2030,Created started script  
# RRoot,1.1.2030,Added pseudo-code to start assignment 8  
# <Wei Wang>,<08.30.2022>,Modified code to complete assignment 8
```

```

# ----- #

# Data ----- #
strFileName = 'products.txt'
lstOfProductObjects = []

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

    properties:
        product_name: (string) with the product's name

        product_price: (float) with the product's standard price
    methods:
        changelog: (When, Who, What)
        RRoot, 1.1.2030, Created Class
        <Wei Wang>, <8.31.2022>, Modified code to complete assignment 8
    """

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

    # -- properties --
    # Product Name
    @property
    def product_name(self):
        return str(self.__product_name).title()

    @product_name.setter
    def product_name(self, value):
        if str(value).isnumeric() == False:
            self.__product_name = value
        else:
            raise Exception("Names cannot be numbers!")

    # Product Price
    @property
    def product_price(self):
        return str(self.__product_price).title()

```

```

@product_price.setter
def product_price(self, value):
    try:
        self.__product_price = float(value)
    except Exception as e:
        raise ("Price must be numbers! \n\t" + e.__str__().title())

# -- Methods --
def show(self):
    print("product name: " + self.product_name + ", " + "product price: " +
self.product_price)

# Processing -----

class FileProcessor:
    """Processes data to and from a file and a list of product objects:

    methods:
        save_data_to_file(file_name, list_of_product_objects):

        read_data_from_file(file_name): -> (a list of product objects)

    changelog: (When, Who, What)
        RRoot, 1.1.2030, Created Class
        <Wei Wang>, <08.31.2022>, Modified code to complete assignment 8
    """

    @staticmethod
    def save_data_to_file(file_name, list_of_product_objects):
        with open(file_name, "w") as file:
            for product in list_of_product_objects:
                file.write(product.product_name + ", " + product.product_price + "\n")

    @staticmethod
    def read_data_from_file(file_name):
        list_of_product_objects = []
        with open(file_name, "r") as file:
            lines = file.readlines()
            for line in lines:
                list_of_product_objects.append(Product(line.split(",")[0].strip(),
line.split(",")[1].strip()))
        return list_of_product_objects

```

```

class IO:
    """ Handle input and output:

    methods:
        show_menu():
        get_menu_input(file_name): -> (file_name)
        show_current_data(file_name):
        show_current_data_from_file(file_name): -> (file_name)
        get_menu_input():
        bye():
    """

    # TODO: Add code to show menu to user
    def show_menu():
        print("""
        Menu of Options
        1) Add Data
        2) Show current data
        3) Load data from file
        4) Save data to file
        5) Exit Program
        """)

    @staticmethod
    def get_menu_input():
        return str(input("Which option would you like to perform? [1 to 5] -")).strip()

    @staticmethod
    def show_current_data_from_file(file_name):
        for product in FileProcessor.read_data_from_file(file_name):
            product.show()

    @staticmethod
    def show_current_data():
        for product in listOfProductObjects:
            product.show()

    @staticmethod
    def input_product_data():
        product_name = input("Input product name: ")
        product_price = input("Input product price: ")
        return Product(product_name, product_price)

```

```
Run: Assignment08
/Users/wei/Documents/_PythonClass/Assignment08/bin/python /Users/wei/Documents/_PythonClass/Assignment08/Assignment08.py

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -1
Input product name: chocolate cookies
Input product price: 10.5

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -2
product name: Product1, product price: 10.5
product name: Chocolate Cookies, product price: 10.5

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -3
product name: Product1, product price: 10.5
```

```
@staticmethod
def bye():
    print("bye")
```

```
# Main Body of Script ----- #
```

```
# Load data from file into a list of product objects when script starts
lstOfProductObjects = FileProcessor.read_data_from_file(strFileName)
```

```
# Show user a menu of options
while True:
    IO.show_menu()
    strChoice = IO.get_menu_input()
    match strChoice:
        case "1":
            lstOfProductObjects.append(IO.input_product_data())
            pass
        case "2":
            IO.show_current_data()
```

```
    pass
case "3":
    IO.show_current_data_from_file(strFileName)
    pass
case "4":
    FileProcessor.save_data_to_file(strFileName, lstOfProductObjects)
    pass
case "5":
    IO.bye()
    break
```

### **Run the code**

Run the script both in PyCharm and an OS command/shell window and capture images of it working on my computer

This is the first screen shot of the script running in PyCharm(Figure 1)

Figure 1: the first screen shot of the script running in PyCharm

This is the second screen shot of the script running in PyCharm(Figure 2)

```
Run: Assignment08 x
product name: Product1, product price: 10.5
product name: Chocolate Cookies, product price: 10.5

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -3
product name: Product1, product price: 10.5

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -4

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -5
bye

Process finished with exit code 0
```

Figure 2: the second screen shot of the script running in PyCharm

This is the screen shot of the script running in IDLE(Figure 3)

```
Python 3.10.5 (v3.10.5:f377153967, Jun 6 2022, 12:36:10) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: /Users/wei/Documents/_PythonClass/Assignment08/Assignment08.py ====

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -1
Input product name: chocolate cookies
Input product price: 10.5

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -2
product name: Chocolate Cookies, product price: 10.5

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -3

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -4

Menu of Options
1) Add Data
2) Show current data
3) Load data from file
4) Save data to file
5) Exit Program

Which option would you like to perform? [1 to 5] -5
bye
>>> |
```

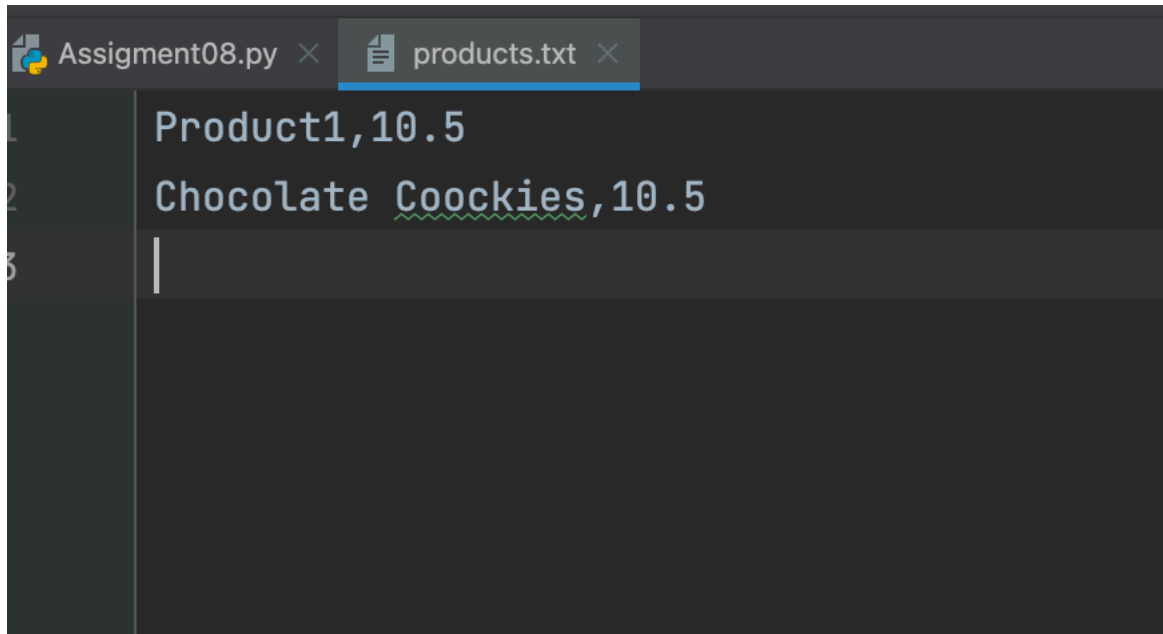
screen shot of the script running in IDLE

Figure 3: the



## Verify that it Worked

open the text editor to verify if the data was there(Figure 4).

A screenshot of a text editor window with two tabs: 'Assignment08.py' and 'products.txt'. The 'products.txt' tab is active and shows two lines of text: 'Product1,10.5' on line 1 and 'Chocolate Cookies,10.5' on line 2. The word 'Cookies' is underlined with a green squiggly line, indicating a spelling correction. Line 3 is empty with a cursor at the start. The editor has a dark theme and a line number margin on the left.

```
1 Product1,10.5
2 Chocolate Cookies,10.5
3 |
```

Figure 4: Verifying that the file has data

## Summary

In this module, I got to learn classes and the components inside it.