Naga Anusha.Perali

March 6, 2022

Python Script Assignment 8

GitHub Repository Link: https://github.com/NagaAusha/ITFnd100-mod8

Objects, Classes and Attributes

Introduction

In this assignment, I'm going to explain how I create a Python Program where user can choose from the menu of options to perform the program by using some custom classes and methods.

Constructor

Constructor are functions or special methods, which will be executed when the object is created for a class. We use constructors to construct the object, to initialize the data members of the class, to initialize the resources we are going to use in the object. In Python the __init__ () method is called the constructor and is always called when an object is created.

Classes and Objects

A class is a user-defined blueprint or prototype from which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by their class) for modifying their state.

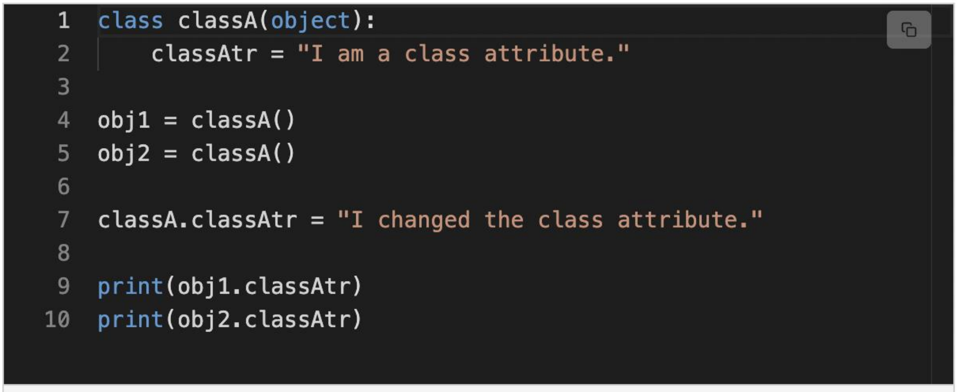
An Object is an instance of a Class. A class is like a blueprint while an instance is a copy of the class with actual values. It's not an idea anymore, it's an actual dog, like a dog of breed pug who's seven years old. You can have many dogs to create many different instances, but without the class as a guide, you would be lost, not knowing what information is required.

I found this article helpful in understanding python class and objects

https://www.geeksforgeeks.org/python-classes-and-objects/ (External link)

Class Attribute

Class attributes are variables of a class that are shared between all of its instances. They differ from instance attributes in that instance attributes are owned by one specific instance of the class only, and are not shared between instances. Check the picture below



```
class classA(object):
    classAtr = "I am a class attribute."

obj1 = classA()
obj2 = classA()

classA.classAtr = "I changed the class attribute."

print(obj1.classAtr)
print(obj2.classAtr)
```

Pic 1: showing the way to change the class attribute.

Getter and Setter

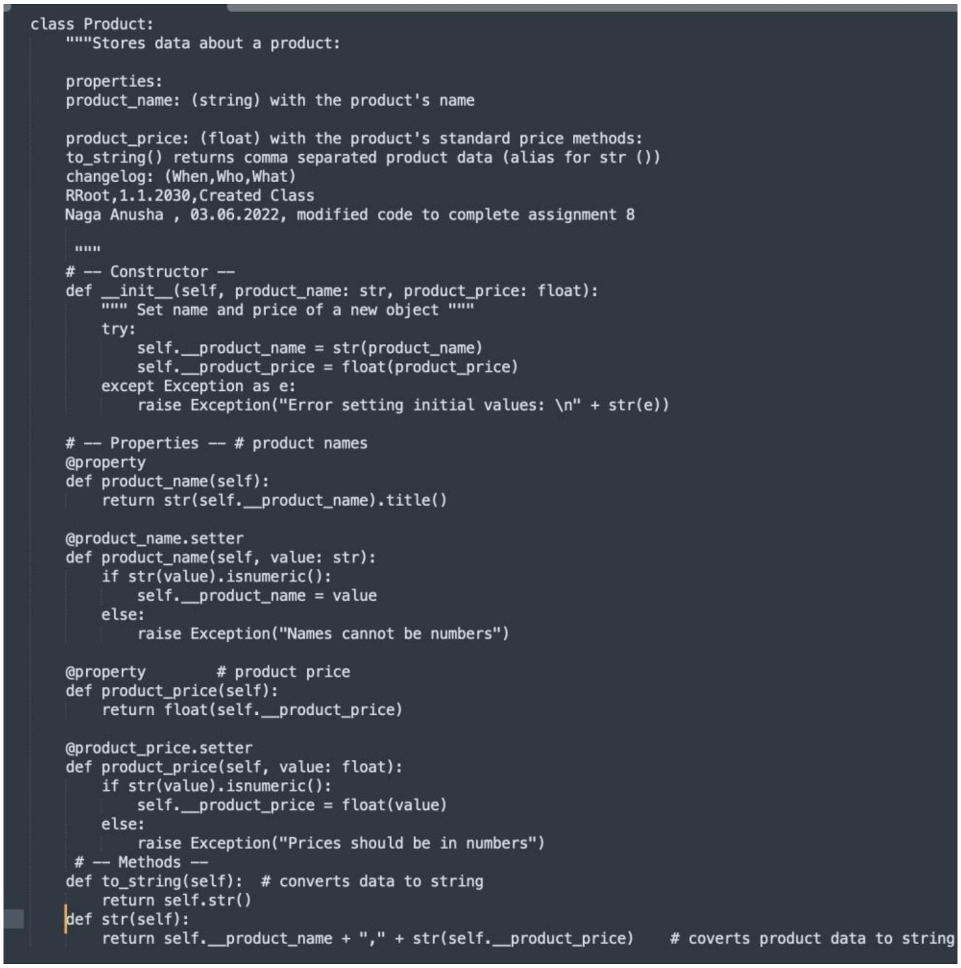
In Python, getters and setters are not the same as those in other object-oriented programming languages. Basically, the main purpose of using getters and setters in object-oriented programs is to ensure data encapsulation. Private variables in python are not actually hidden fields like in other object-oriented languages. Getters and Setters in python are often used when:

We use getters & setters to add validation logic around getting and setting a value.

To avoid direct access of a class field i.e., private variables cannot be accessed directly or modified by external user.

Class Product

To start off the program I first started by creating a constructor to set the initial values of the fields, product_name and product_price, by using the __init__ () method, I then defined the attributes inside the constructor using the self-parameter. Using the two underscores Infront of the attribute means I am using that as a private attribute. And created getter and setter property for the product_name ,the getter property is formatted and return a float variable for the product_price also includes error handling to make sure that the price is in numeric and setter property include Exception handling to get the product name as string. Next, I created a method to override the built in __str__ () method to do this I created a function to return the product name and price.

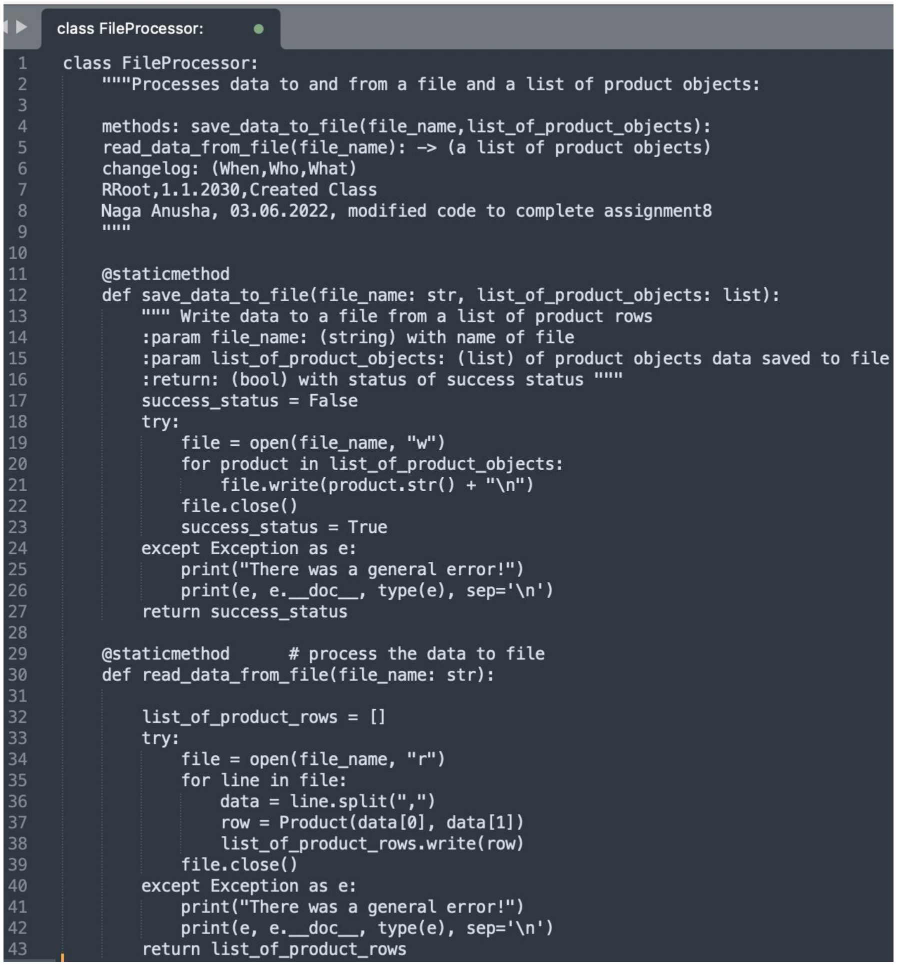


```
class Product:
    """Stores data about a product:

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

    product_price: (float) with the product's standard price methods:
    to_string() returns comma separated product data (alias for str ())
    changelog: (When,Who,What)
    RRoot,1.1.2030,Created Class
    Naga Anusha , 03.06.2022, modified code to complete assignment 8

     """
    # -- Constructor --
    def __init__(self, product_name: str, product_price: float):
        """ Set name and price of a new object """
        try:
            self.__product_name = str(product_name)
            self.__product_price = float(product_price)
        except Exception as e:
            raise Exception("Error setting initial values: \n" + str(e))

    # -- Properties -- # product names
    @property
    def product_name(self):
        return str(self.__product_name).title()

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

    @property        # product price
    def product_price(self):
        return float(self.__product_price)

    @product_price.setter
    def product_price(self, value: float):
        if str(value).isnumeric():
            self.__product_price = float(value)
        else:
            raise Exception("Prices should be in numbers")
     # -- Methods --
    def to_string(self):  # converts data to string
        return self.str()
    def str(self):
        return self.__product_name + "," + str(self.__product_price)    # coverts product data to string
```

Pic 2: Code for Class Product

Class FileProcessor

In the File Processor class, I created two static methods to read data from the file and save data to the file. The function to read data from a file used the open () function to open the file for reading (using ‘r’ mode). To read data from the file, I used a for loop to read each line of the file using the split () method, and append the index data to a list of products and used the close () method to close the file and returned the list of products. and then we open the File for writing the data by using append mode, i then used the close () method to close the file.

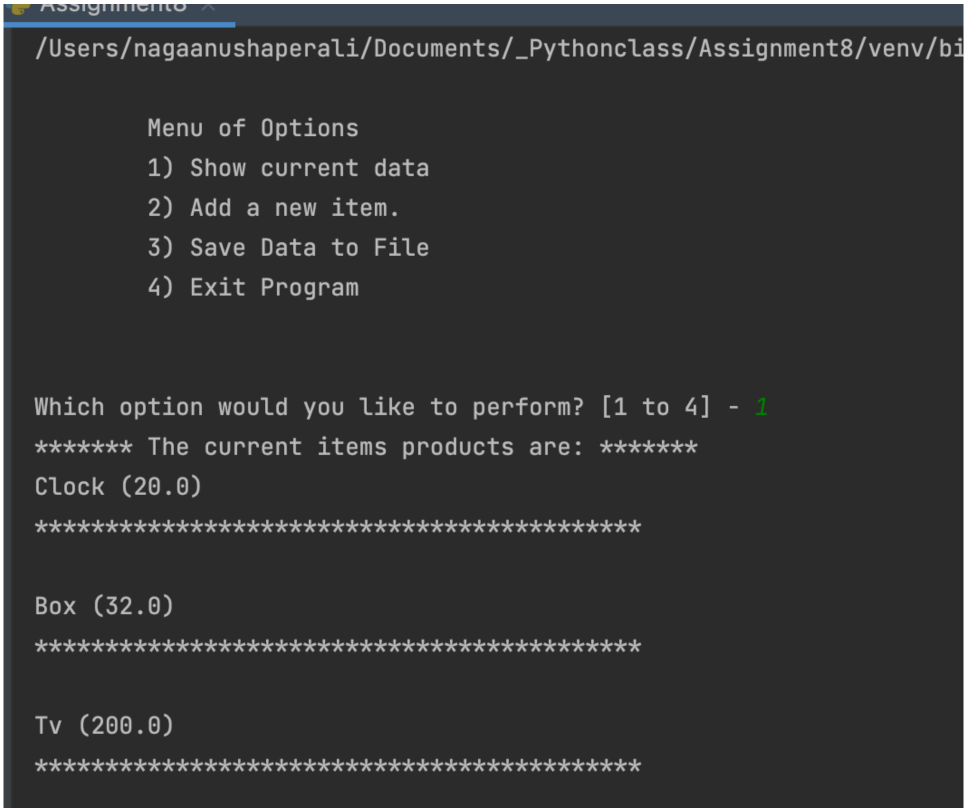


```python
class FileProcessor:
    """Processes data to and from a file and a list of product objects:

    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
    Naga Anusha, 03.06.2022, modified code to complete assignment8
    """

    @staticmethod
    def save_data_to_file(file_name: str, list_of_product_objects: list):
        """ Write data to a file from a list of product rows
        :param file_name: (string) with name of file
        :param list_of_product_objects: (list) of product objects data saved to file
        :return: (bool) with status of success status """
        success_status = False
        try:
            file = open(file_name, "w")
            for product in list_of_product_objects:
                file.write(product.str() + "\n")
            file.close()
            success_status = True
        except Exception as e:
            print("There was a general error!")
            print(e, e.__doc__, type(e), sep='\n')
        return success_status

    @staticmethod      # process the data to file
    def read_data_from_file(file_name: str):

        list_of_product_rows = []
        try:
            file = open(file_name, "r")
            for line in file:
                data = line.split(",")
                row = Product(data[0], data[1])
                list_of_product_rows.write(row)
            file.close()
        except Exception as e:
            print("There was a general error!")
            print(e, e.__doc__, type(e), sep='\n')
        return list_of_product_rows
```

Pic3: Class FileProcessor code.

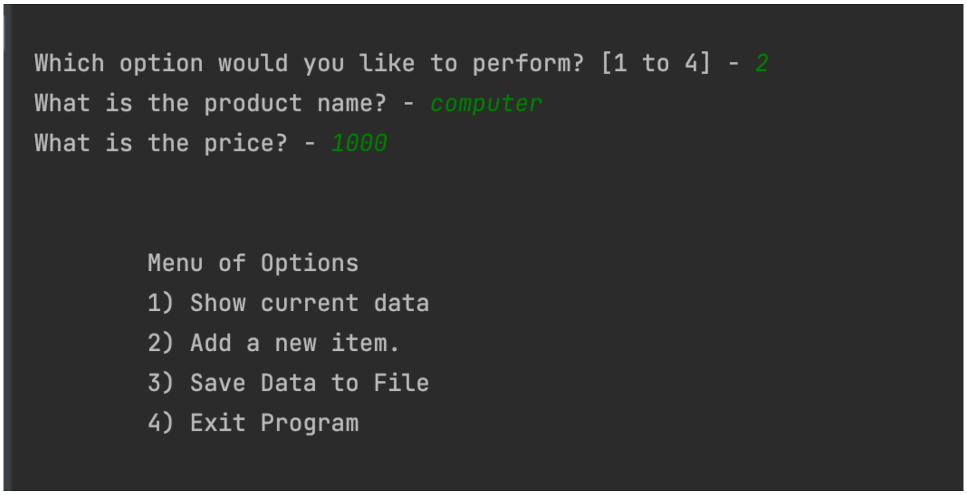
Class IO

I displayed the menu of options to the user such as Show the current data, Add the new item, save and exit the data to the file when user makes that choice. User options are shown the following images.



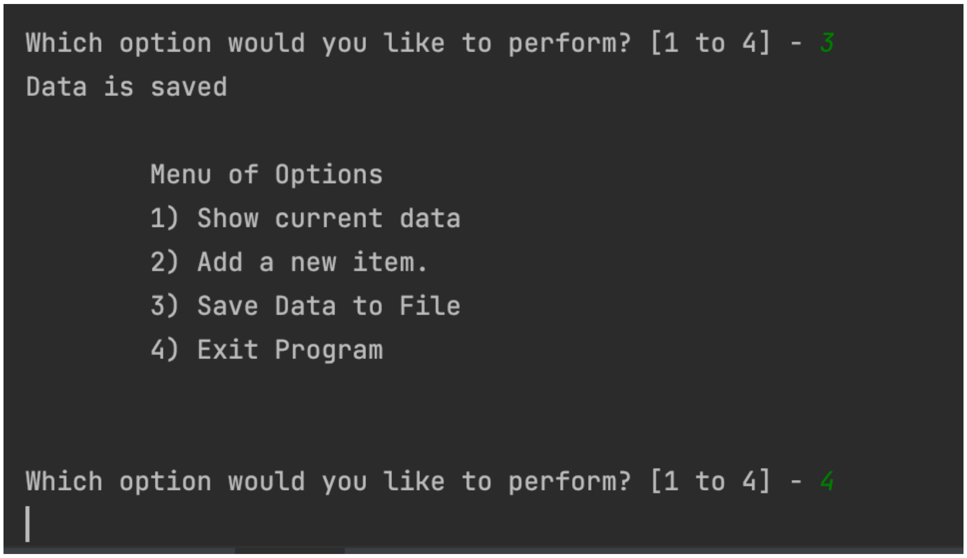
```
/Users/nagaanushaperali/Documents/_Pythonclass/Assignment8/venv/bi

        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Save Data to File
        4) Exit Program


Which option would you like to perform? [1 to 4] - 1
******* The current items products are: *******
Clock (20.0)
*******************************************

Box (32.0)
*******************************************

Tv (200.0)
*******************************************
```

Pic4: Displaying the menu of options



```
Which option would you like to perform? [1 to 4] - 2
What is the product name? - computer
What is the price? - 1000


        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Save Data to File
        4) Exit Program
```

Pic5: Adding the new item



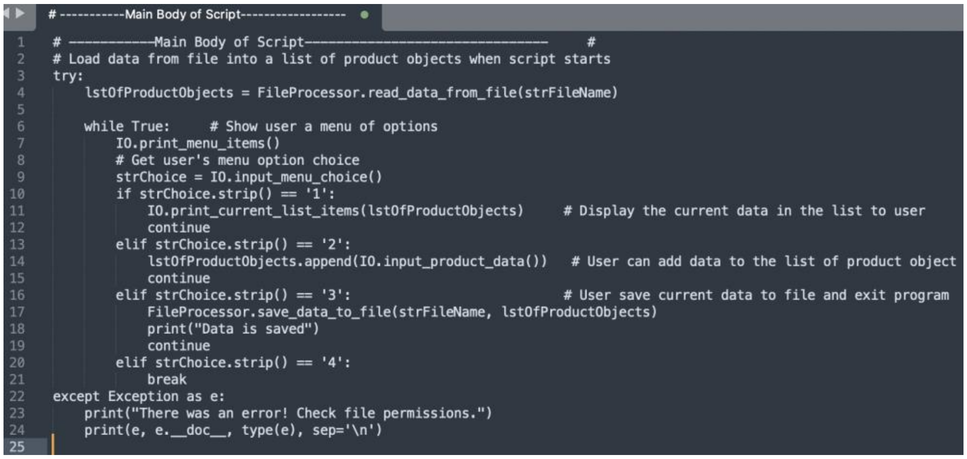
```
Which option would you like to perform? [1 to 4] - 3
Data is saved

        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Save Data to File
        4) Exit Program


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

Pic6: Saving the data and exit the program

Main Body of Script:

To start the main body of the script by calling the read_data_from_file function from the FileProcessor class and I load the data from the products.txt file into a list of product objects, by using the try-except error handling method, if the products.txt file did not exist then an error message will be displayed with the type of the error.

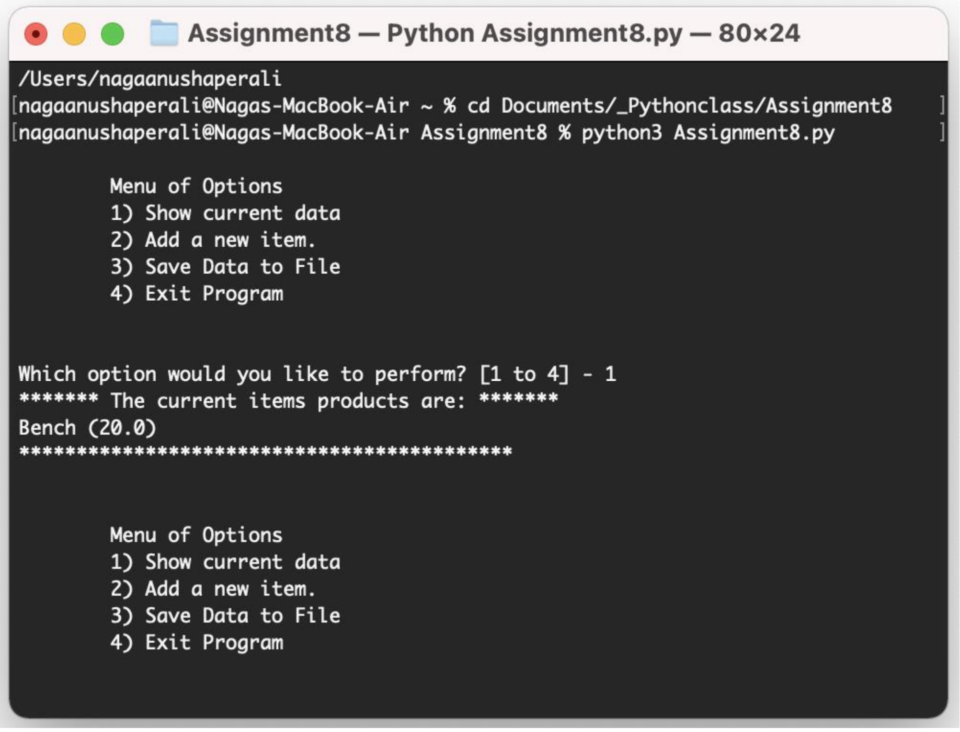
In the main body, while loop shows the menu to the user and calls the get_user_choice function from the IO class. Once the user enters a menu choice, an if elf statement is used to call each function according to user entered choice.



```
# ------------Main Body of Script---------------------------------     #
# Load data from file into a list of product objects when script starts
try:
    lstOfProductObjects = FileProcessor.read_data_from_file(strFileName)

    while True:     # Show user a menu of options
        IO.print_menu_items()
        # Get user's menu option choice
        strChoice = IO.input_menu_choice()
        if strChoice.strip() == '1':
            IO.print_current_list_items(lstOfProductObjects)     # Display the current data in the list to user
            continue
        elif strChoice.strip() == '2':
            lstOfProductObjects.append(IO.input_product_data())   # User can add data to the list of product object
            continue
        elif strChoice.strip() == '3':                          # User save current data to file and exit program
            FileProcessor.save_data_to_file(strFileName, lstOfProductObjects)
            print("Data is saved")
            continue
        elif strChoice.strip() == '4':
            break
except Exception as e:
    print("There was an error! Check file permissions.")
    print(e, e.__doc__, type(e), sep='\n')
```

Pic7: Code for Main Body of Script

I also ran the same program in the terminal window for results refer image below



```
/Users/nagaanushaperali
[nagaanushaperali@Nagas-MacBook-Air ~ % cd Documents/_Pythonclass/Assignment8    ]
[nagaanushaperali@Nagas-MacBook-Air Assignment8 % python3 Assignment8.py         ]

        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Save Data to File
        4) Exit Program


Which option would you like to perform? [1 to 4] - 1
******* The current items products are: *******
Bench (20.0)
*******************************************



        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Save Data to File
        4) Exit Program
```

Pic8: Results in the terminal window.

Summary

In this assignment by referring to module 8 video and by referring the information in the text book and some additional weblinks I learned to work with classes, objects, attributes and how where and when static methods can be used, and how the keyword self can be used. And I also learned how to create an account in GitHub Desktop and to create and upload the files, documents to repository using the GitHub desktop.