Hwai-Jiang Jong

May 31, 2021

Foundations of Programming: Python

Assignment 08

https://github.com/uwp-h2021/IntroToProg-Python-Mod08

Product Object Script

# Introduction

This report documents my learning concerning classes and objects, which is demonstrated by a Python script I created for Assignment08. It contains three sections (excluding this intreoduction):

1. Script Implementation;
2. GitHub Publication using GitHub Desktop;
3. Summary

# The Product Object Script

## Script Planning

In this script, I am demonstrating a Python script to operate on a class of product object data. The following are the steps used to demonstrate the working code, which was then written according to the pseudo-code in Assignment08-Starter.py. Before the run, I created a text data file that contains two lines of product name and price. The text data file name is “*products.txt*” stored in the hard drive, and its content is shown in Figure 1.

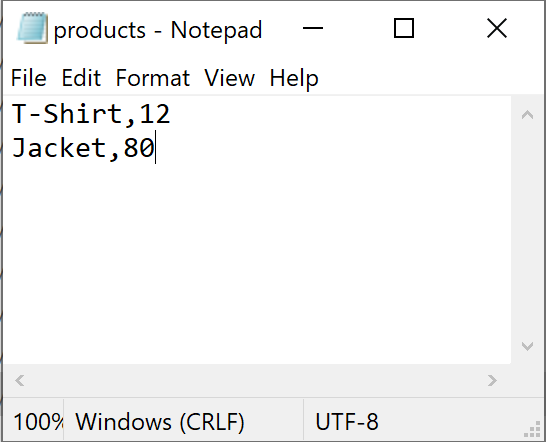


Figure 1 The “products.txt” With Its Content

The steps in this demonstration are shown below.

*Load the data from “products.txt” into a list of product objects.*

*While True:*

*{*

*Display menu of options to user to choose from.*

*Receive user choice between options 1-4:*

* + *If option 1: Display current product data*
  + *If option 2: Receive and add user input data into the list of product objects*
  + *If option 3: Save data into file*

*Use Exception to handle incompatible type of input data*

* + *If option 4: Exit the program, break*
  + *Else: Exception handling to let user re-enter choice*

*}*

In the test run, I added two more product data. They were (‘hat’, ‘5’) and (‘shoes’, ‘60’). For user readability purposes, I added a dollar sign ($) in front of the price value when the option 1 was chosen. However, the actual stored price data were float type in the memory and text file.

## Writing the Script in PyCharm

I created a new project folder “C:\\_PythonClass\Assignment08\”, in which the script “Assignment08.py” was created and tested. The script was completed with its various sections described in detail as follows.

1. Change History: Figure 2 shows the history in creating the script.

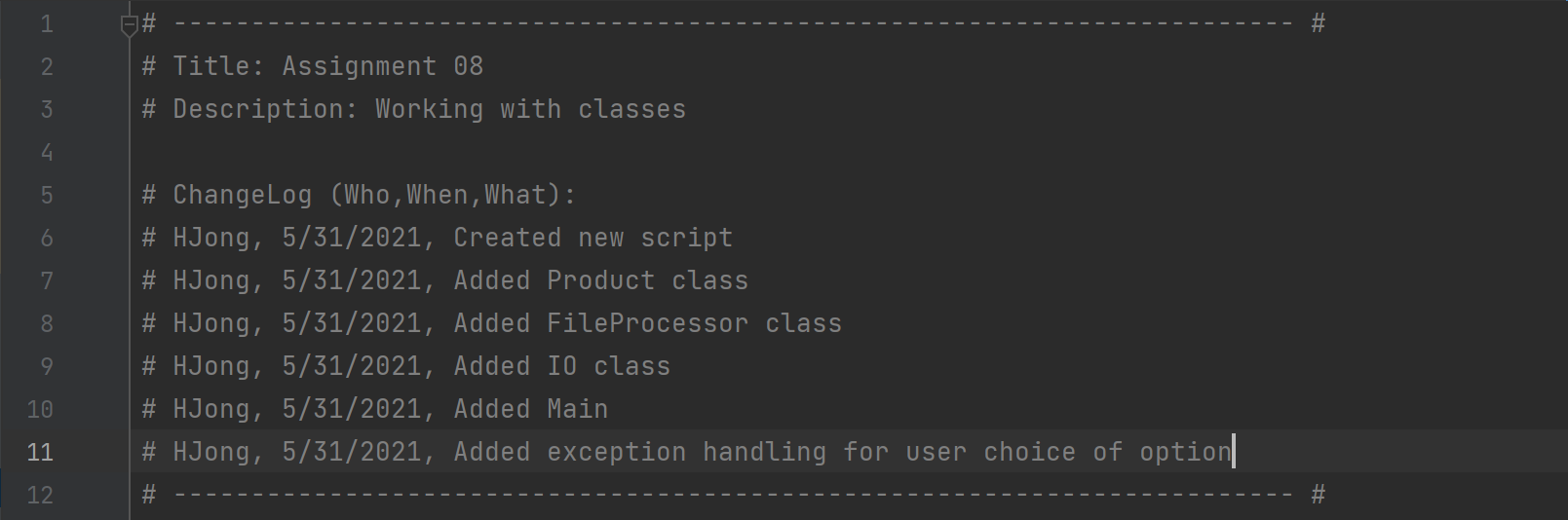


Figure 2 Change Log of the Script

1. Data definition: Figure 3 shows the variables in the script. The variables are very simple because of the encapsulation of data using class objects.

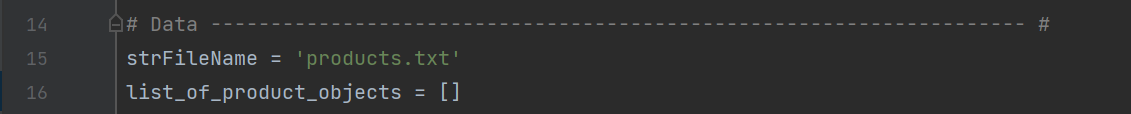


Figure 3 Data Declaration Section of the Script

1. Class Product – This section contains a class *Product* which consists of *constructor*, *property getters*, and *property setters* for product name and product price. See Figure 4.



Figure 4 The Class Product

1. Class *FileProcessor* – This section contains the data processing part of the code, which consists of *save\_data\_to\_file() and read\_data\_from\_file()* methods, as shown in Figure 5.

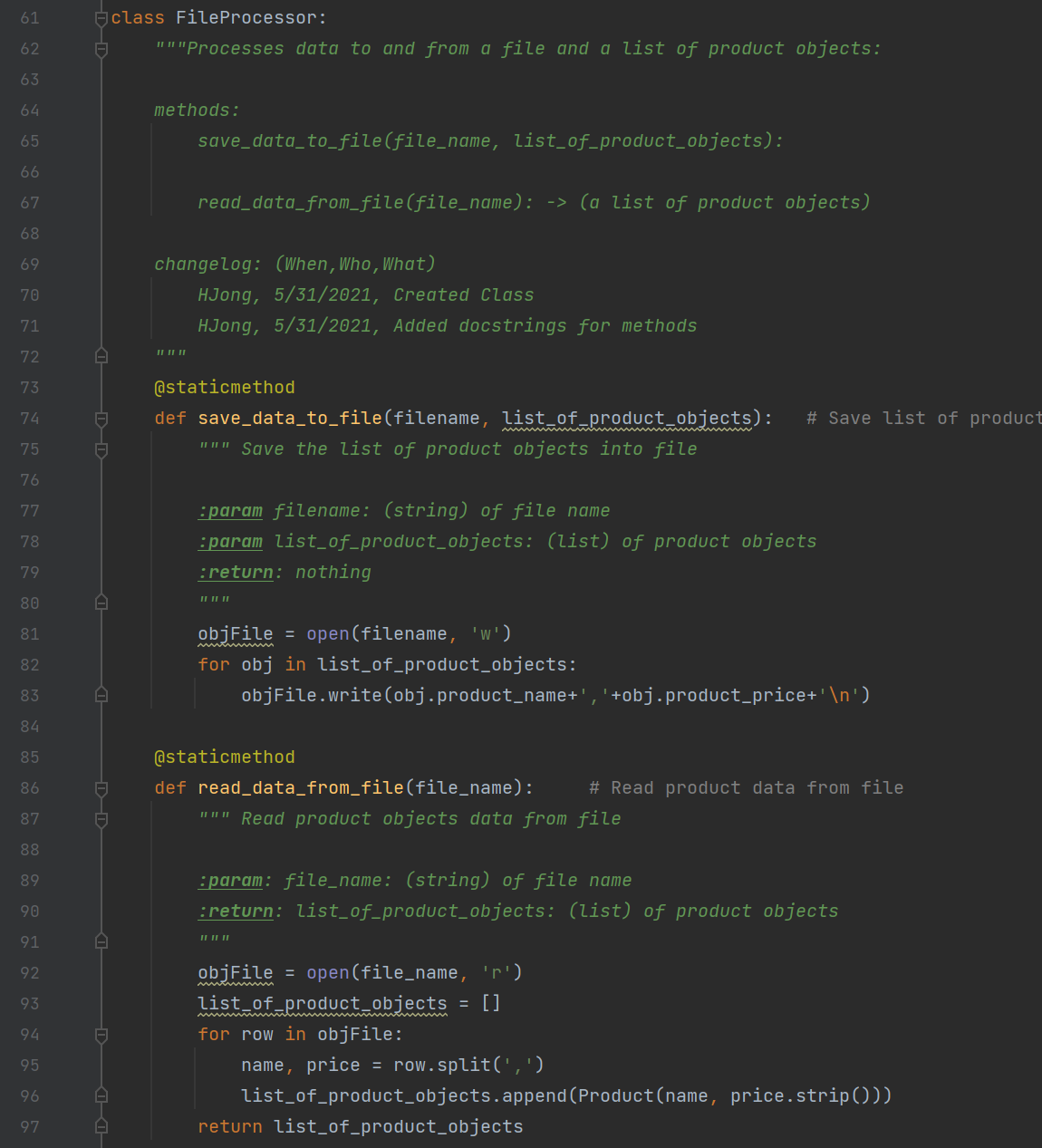


Figure 5 The Class FileProcessor for Presentation

1. Class *IO*: This section contains class *IO* that handles the presentation part of the code. The *show\_menu()* and *get\_user\_choice()* methodsare shown in Figure 6.

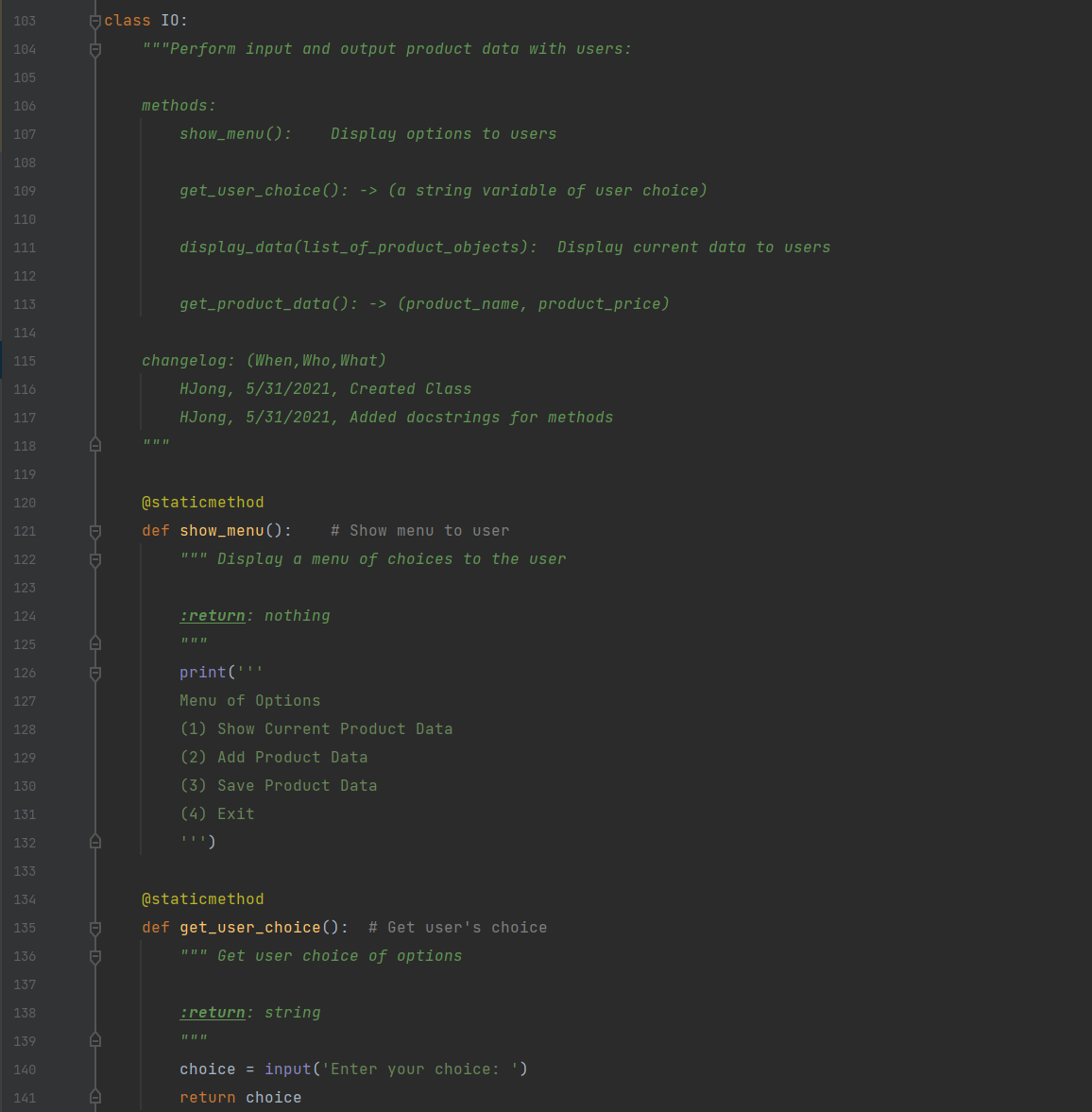


Figure 6 Class IO: show\_menu() and get\_user\_choice() Methods

The *display\_data()* and *get\_product\_data()* methods are shown in Figure 7

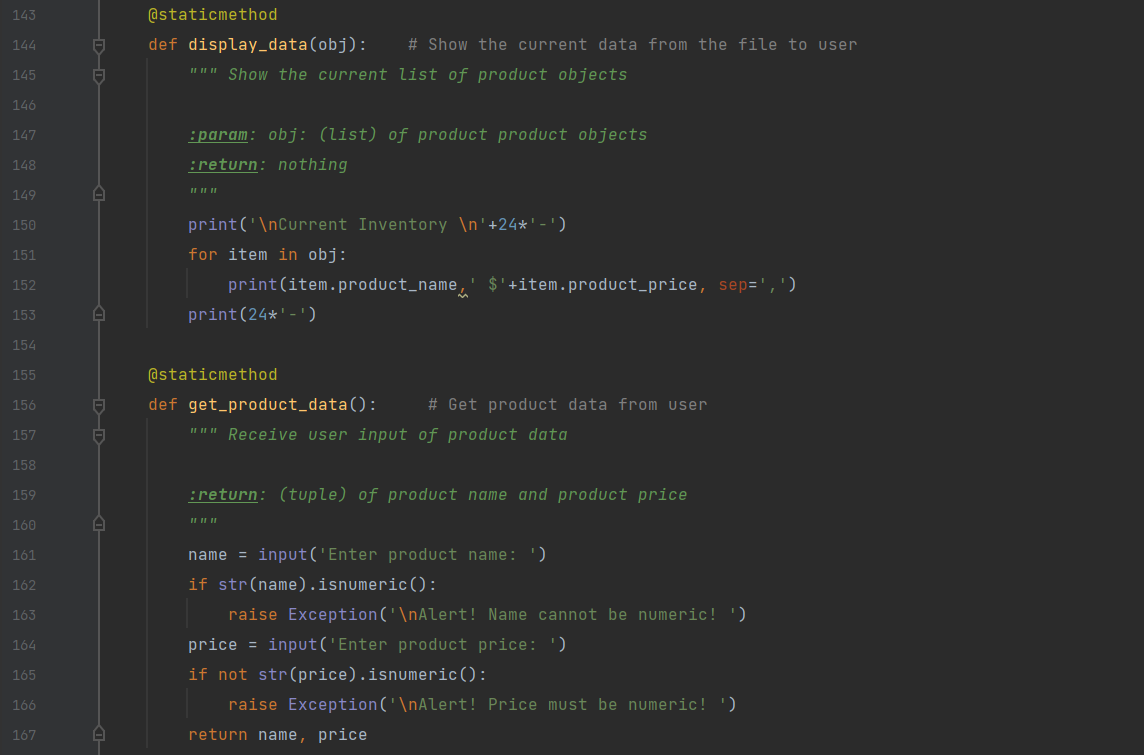


Figure 7 Class IO: display\_data() and get\_product\_data() Methods

1. Main program – The main program executes the steps described in the “Script Planning” section above. It first loads the data from “*product.txt*” into the list of product objects, show the menu of options, receives and processes user choice through options 1-3, and exited the program if option 4 is selected. Figure 8 shows the implementation of the main program. The *read\_data\_from\_file()* and *write\_data\_to\_file()* methods in the class *FileProcessor* handles how the product objects are accessed and processed.

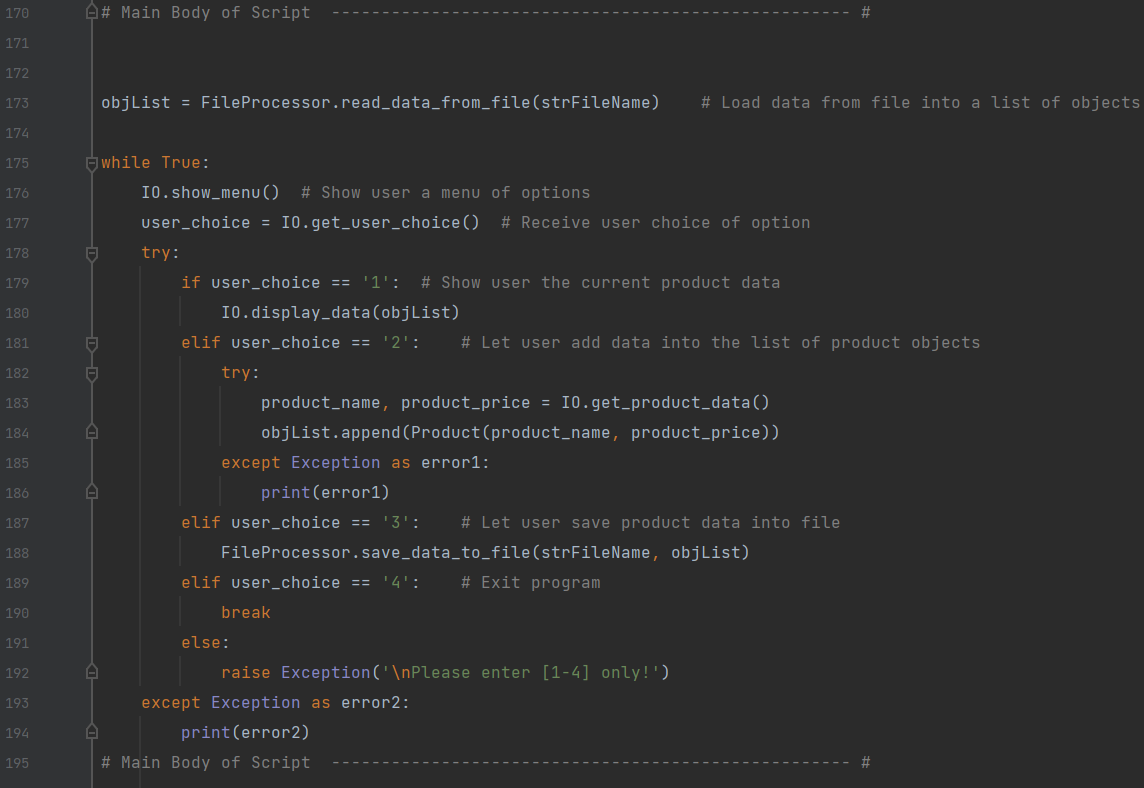


Figure 8 Main Program that Processes User Choice Options

## Executing the Script in PyCharm

In this section, I am demonstrating the script in performing actions described in the “Script Planning” section. As mentioned in the script description, the *read\_data\_from\_file()* and *write\_data\_to\_file()* in the class *FileProcessor* are the two keys methods by which the product objects were accessed and processed. I also demonstrated the exception handling when the user input of the price of the shoes contained a dollar sign ($). It triggered an error message, and the program alerted the user that only float type for price value was accepted. The run results are shown in Figure 9.

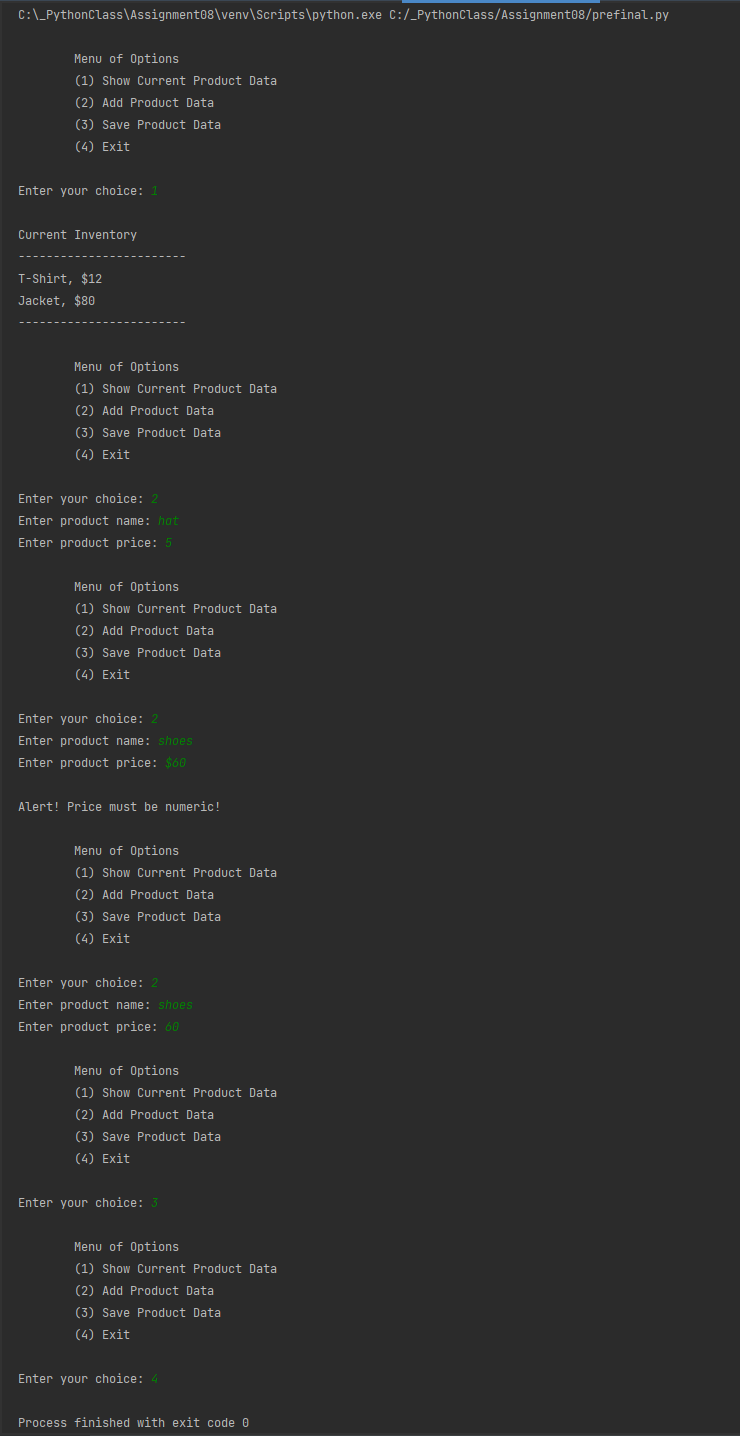


Figure 9 Run Results in PyCharm

## Checking the Data File After PyCharm Run

Finally, I opened “*products.txt”* to verify the product data were added as intended, which is shown in Figure 10.

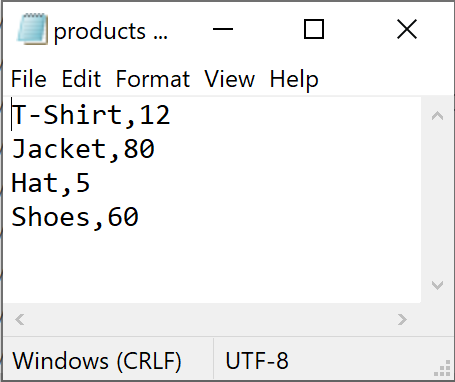


Figure 10 Product Data in “products.txt” After the Run in PyCharm

## Executing the Script in Windows Command Prompt

I went to “C:\\_PythonClass\Assignment08\” and run Assignment08.py in Windows command prompt. Same steps as the PyCharm run were followed, but I only added one more item (‘eye glasses’, ‘115’). This time, I also demonstrated a successful exception handling when user input a numeric product name. The results are shown in the screenshot in Figure 11.

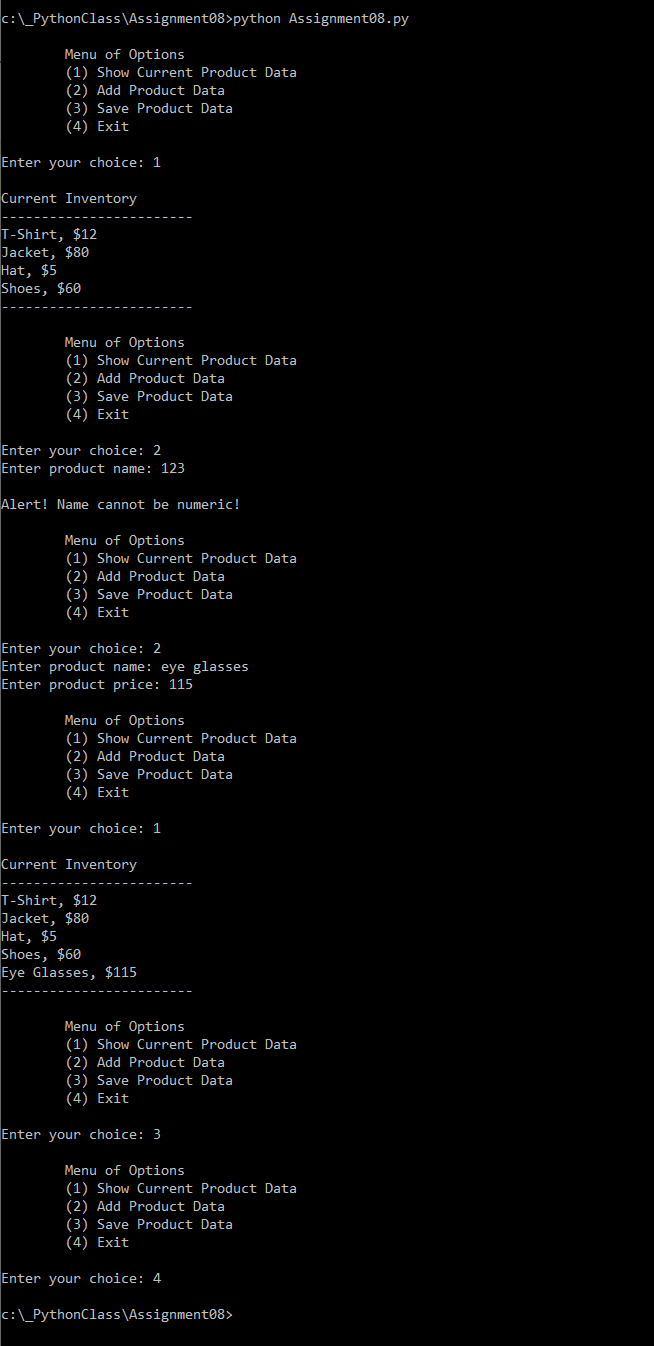


Figure 11 Run Results in Windows Command Prompt

## Checking the Data File After Windows Command Run

After the Windows command run, the file “*products.txt*” in the working folder had five items, as shown in Figure 12. The script was thus validated in both PyCharm and Windows command.

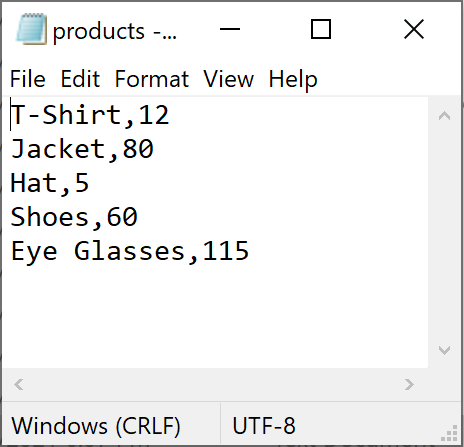


Figure 12 Product Data in “products.txt” After the Run in Windows Command

# GitHub Repository

I dropped the assignment files into the local repository folder as shown in Figure 13.

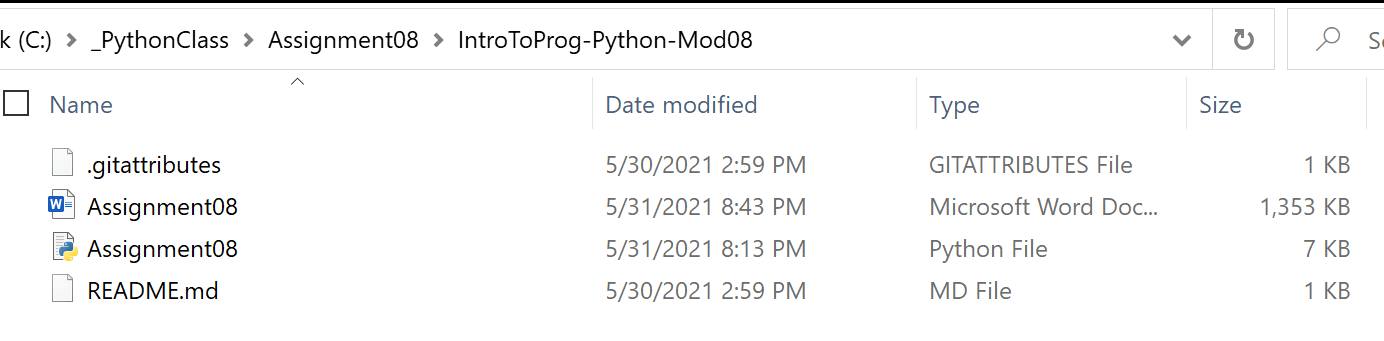


Figure 13 Assignment Files Copied into Local Repository Folder

I used GitHub Desktop to upload the files to the GitHub repository, as shown in Figure 14.

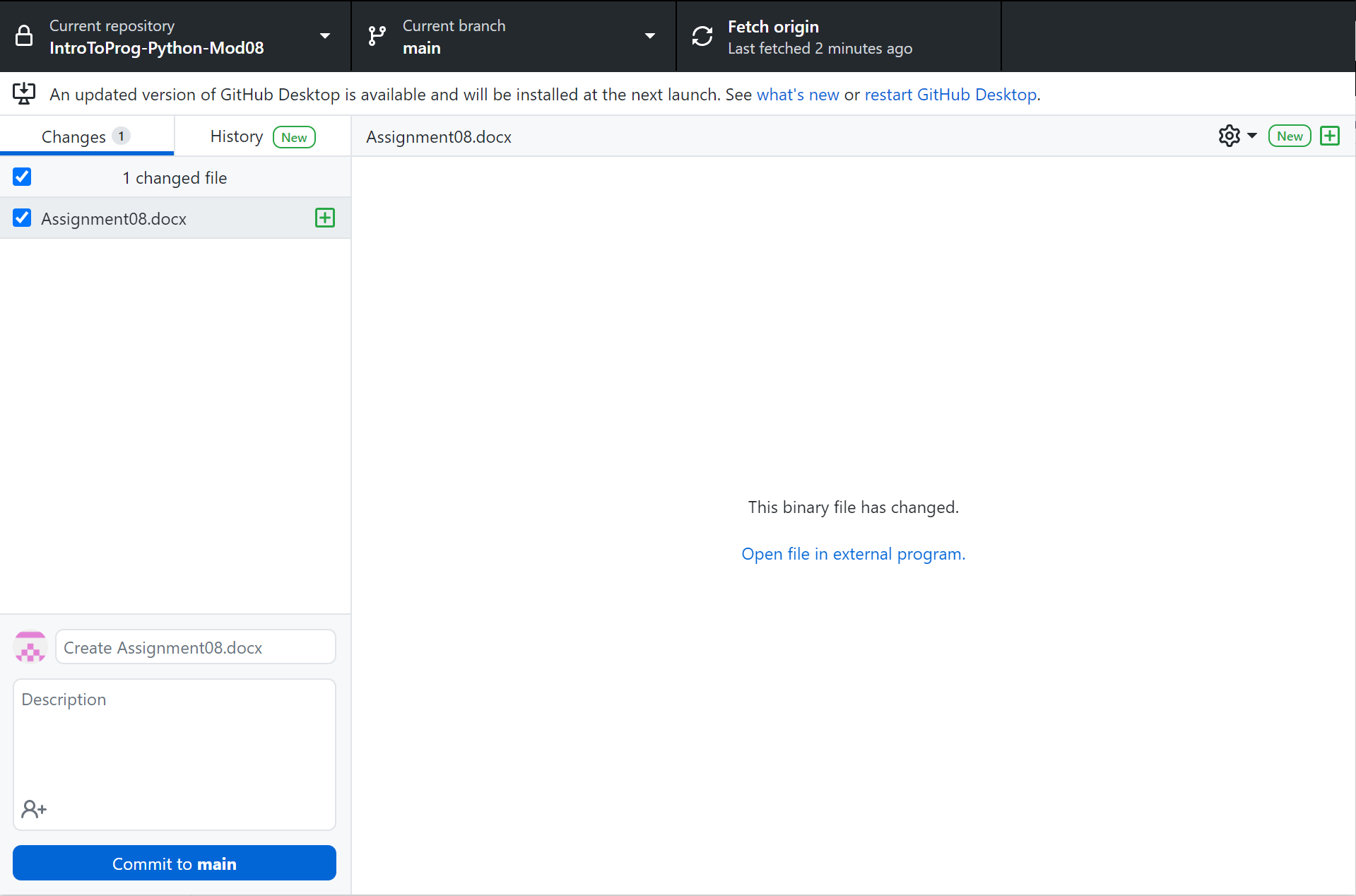


Figure 14 Uploading Assignment Files to GitHub.com

# Summary

This report summarizes the steps I took to implement the script and demonstrate it by test running the code in PyCharm and Windows command. This module has been the most difficult part of this course so far. In particular, the concept of class versus object, field versus attribute, property setter versus property getter, and private attribute versus public attribute requires much practice of many short scripts to solidify the learning. The practice of these short codes to try out the behavior of class objects have shown to be very helpful.