**Course:** Foundations of Programming, Python

**Assignment:** 07

**Name:**  Pickling and Exception Handling in Python Script

**Description:** CreatingPython script that stores data in a binary file with handling errors

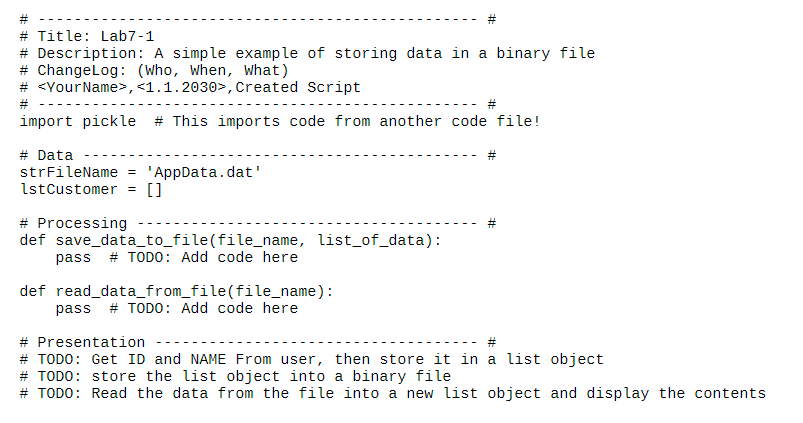
**Created:** 05/30/2020 by Kate Golenkova

**Introduction**

In this assignment I will show how I created a new python script that askes user to add new data (ID and Name), read and stores data in a binary file with handling errors. This script works with binary file **AppData.dat** to read the data from it and save new data. Also, python script is handling errors without halting the execution abruptly. I will show results of tests run in both PyCharm and Command Line.

**Pseudo-Code of Script**

To create new script for Assignment 07 I used simple template **Lab7-1\_starter.py**, that you can see in **Figure 1.** below:

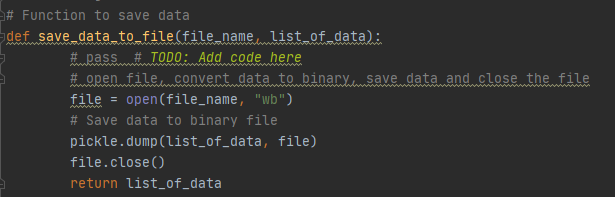


**Figure 1.** Simple Template

I downloaded this file and used it as pseudo-code to create my script. Of course, I added more functions to get user input, read and write binary file and display the Menu, to make it more user friendly.

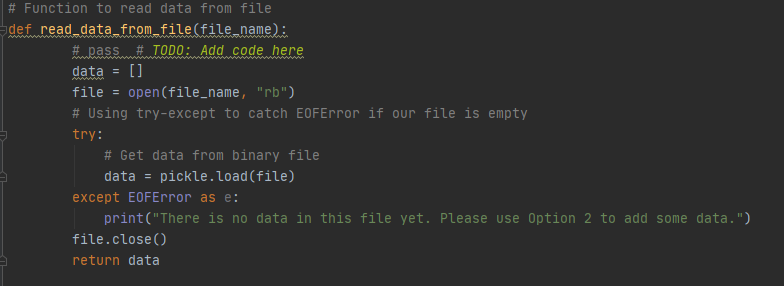
**Python Script with Pickling and Exceptions Handling**

As soon the task for this assignment was to create script that can work with binary file and handle errors without interruptions, I started with functions ***save\_data\_to\_file*** and ***read\_data\_from\_file***. In **Figure 2.** you can see my code for saving data to binary file with using pickle:



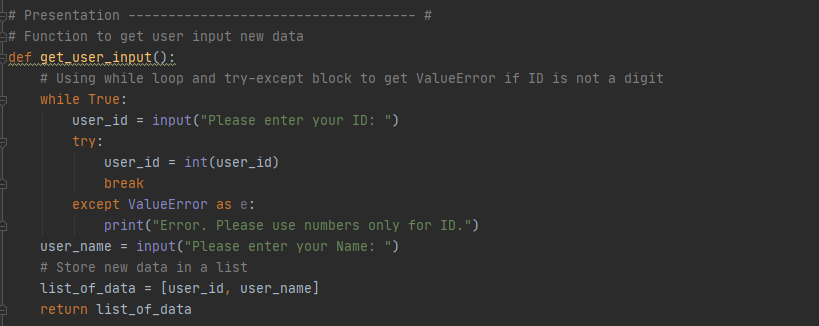
**Figure 2.** Code for saving data to a binary file.

To read data from binary file **AppData.dat** I created function, that you can find in **Figure 3.** Also, in this function I added error handling in case binary file is empty. I found the needed error in list of Python’s exception classes here: <https://docs.python.org/3/library/exceptions.html#bltin-exceptions>.



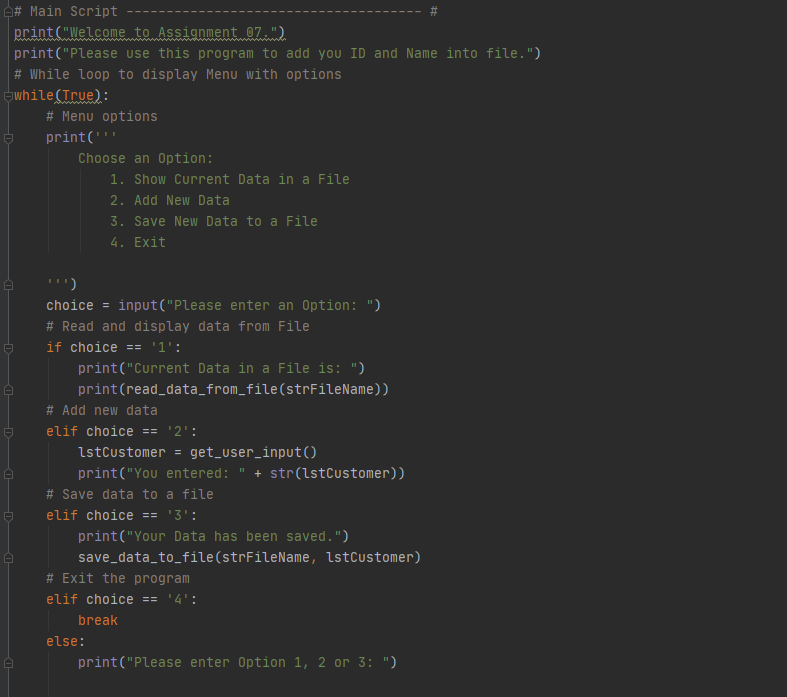
**Figure 3.** Code for reading data from binary file.

After two main functions have been created, I added another one to Presentation part to get user input new data. As soon my program asks for user ID and Name, there is a chance someone can put letters or special signs for ID. To exclude this, I added **try-except** block to handle **ValueError** without interruptions. Please check **Figure 4.** on page 3 to see the code for ***get\_user\_input*** function.



**Figure 4.** Code to get user input new data.

Then I added main body of my script with Menu so user can choose an option. Here I used all my functions with proper arguments. Please see **Figure 5.** below:

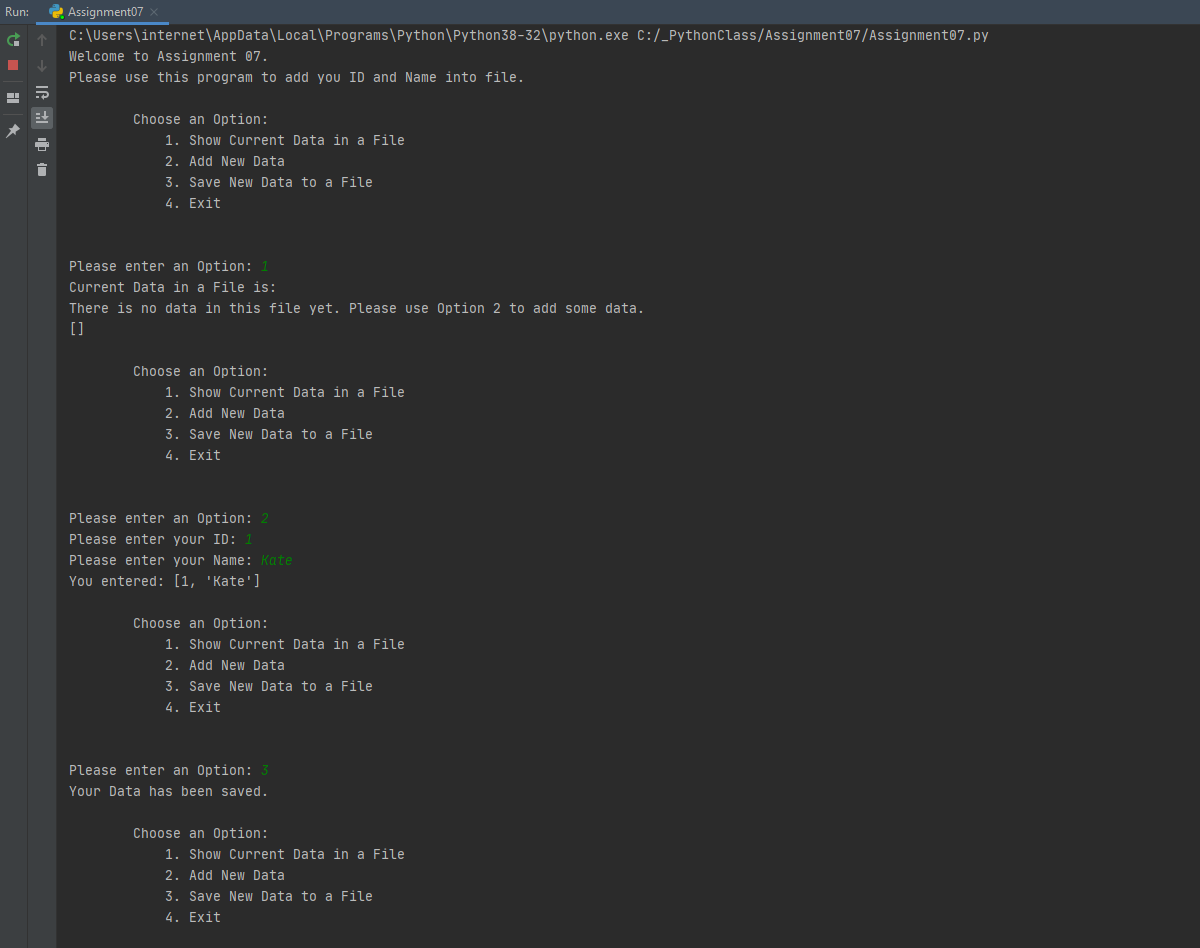


**Figure 5.** Menu with options to read/save/add new data.

So, I created the python script that asks user to choose an option from the Menu to read the file, add new data or save new data to file, or just exit the program. In next part I will show how the program has been tested in PyCharm and CMD environments.

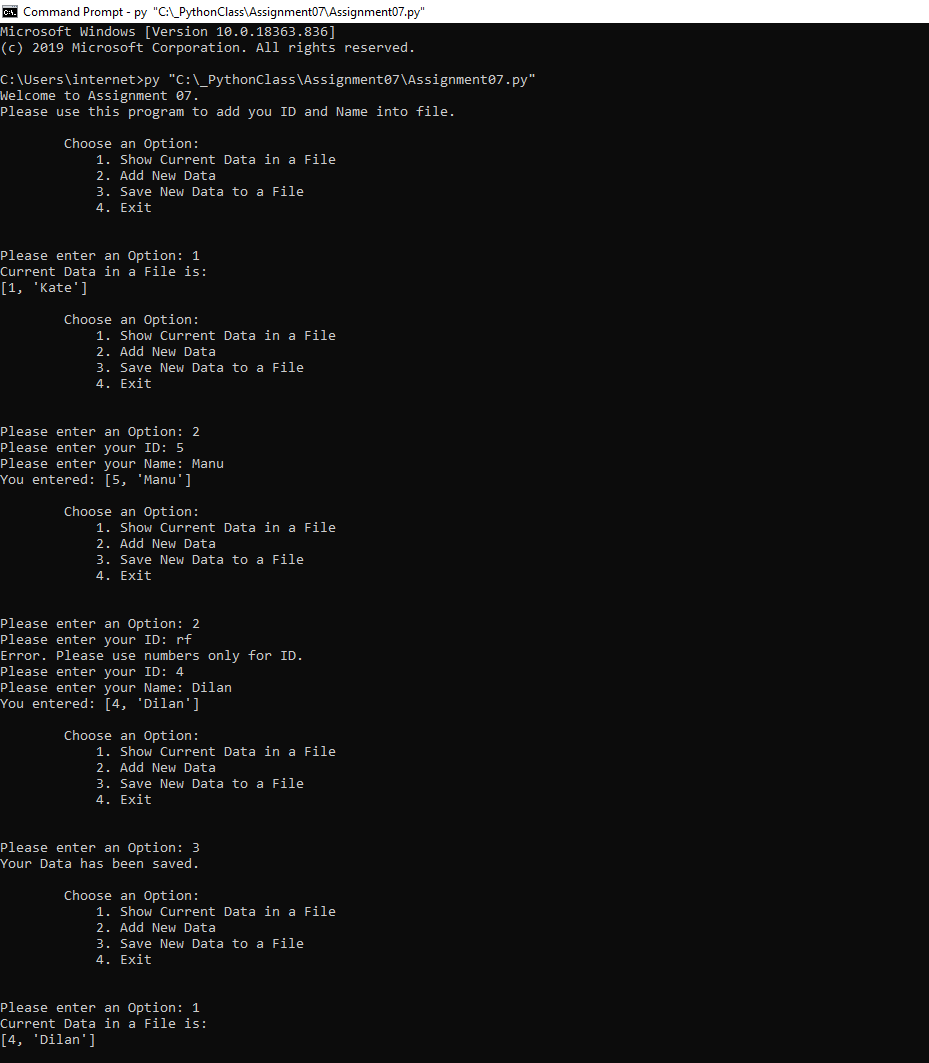
**Tests of Python Script in PyCharm and CMD**

I created the empty binary file **AppData.dat** and saved it to the **Assighnment07** folder, where my new python script has been saved already. After all components were ready, I tested the code in PyCharm (see **Figure 6.**) and in Command line (see **Figure 7.**) and checked data in the file.



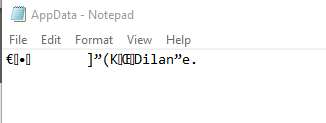
**Figure 6.** Successful test of code in PyCharm.

As you can see in **Figure 6.** I checked current data in the file and got the information that file is empty instead of interruption of code with error. I added new data and saved it to binary file. Then I run test in CMD and checked current data. As you can see in **Figure 7.** my script read data from binary file. Then I added new data and tested input ID with letters, not digits. Finally, I saved new data to a file.



**Figure 7.** Successful test of code in Command Line.

After both tests in PyCharm and CMD I checked my binary file and got the next output that you can find in **Figure 8.** On page 6.



**Figure 8.** The data in binary file AppData.dat after tests run.

**Summary**

I read the next chapter in the book and Programing Notes for Module07, I also watched all the videos and used external resources to understand how to work with binary files with pickle. I found several external resources to read about this, and used this one <https://wiki.python.org/moin/UsingPickle> to understand it more. Also, I learned how to trap errors with try-except blocks. I wrote new python script and my successful tests showed I created working script to read data from binary file and to write to it with handling errors without halting the execution abruptly.