Nicolas Bolinaga

8/12/2019

Introduction to Python

Assignment05

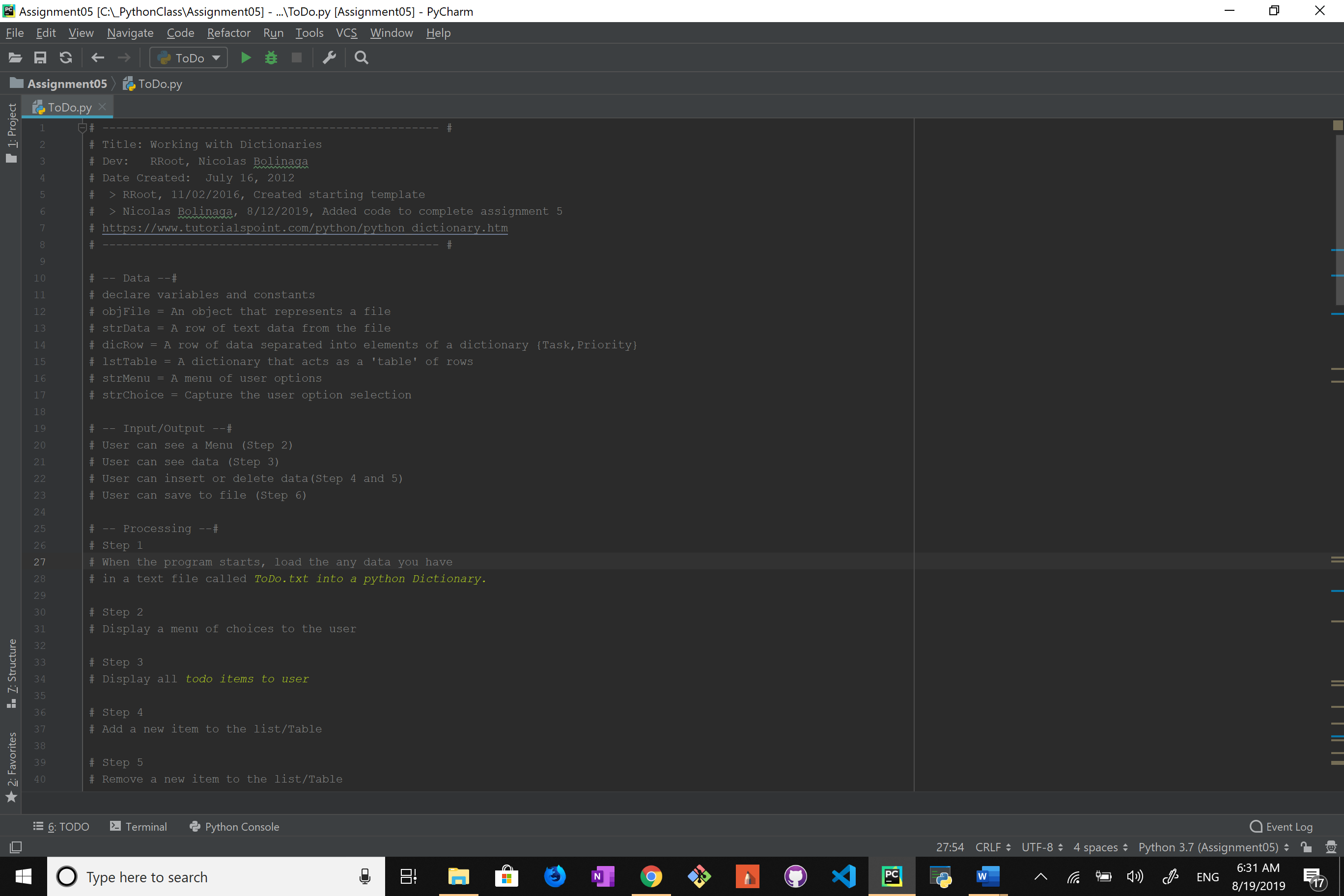
Python Dictionaries and Lists

This week’s assignment is based on something new we learned last class, Dictionaries and Lists. The Goal is to create a program that can read and write, from/to a text file, tasks and their priority level. The information must be stored in dictionaries and those dictionaries stored on a list. But first we must define what these are;

Python Dictionary: *“in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair.” – John Sturtz (2018). Dictionaries in Python.* [*https://realpython.com/python-dicts/*](https://realpython.com/python-dicts/)

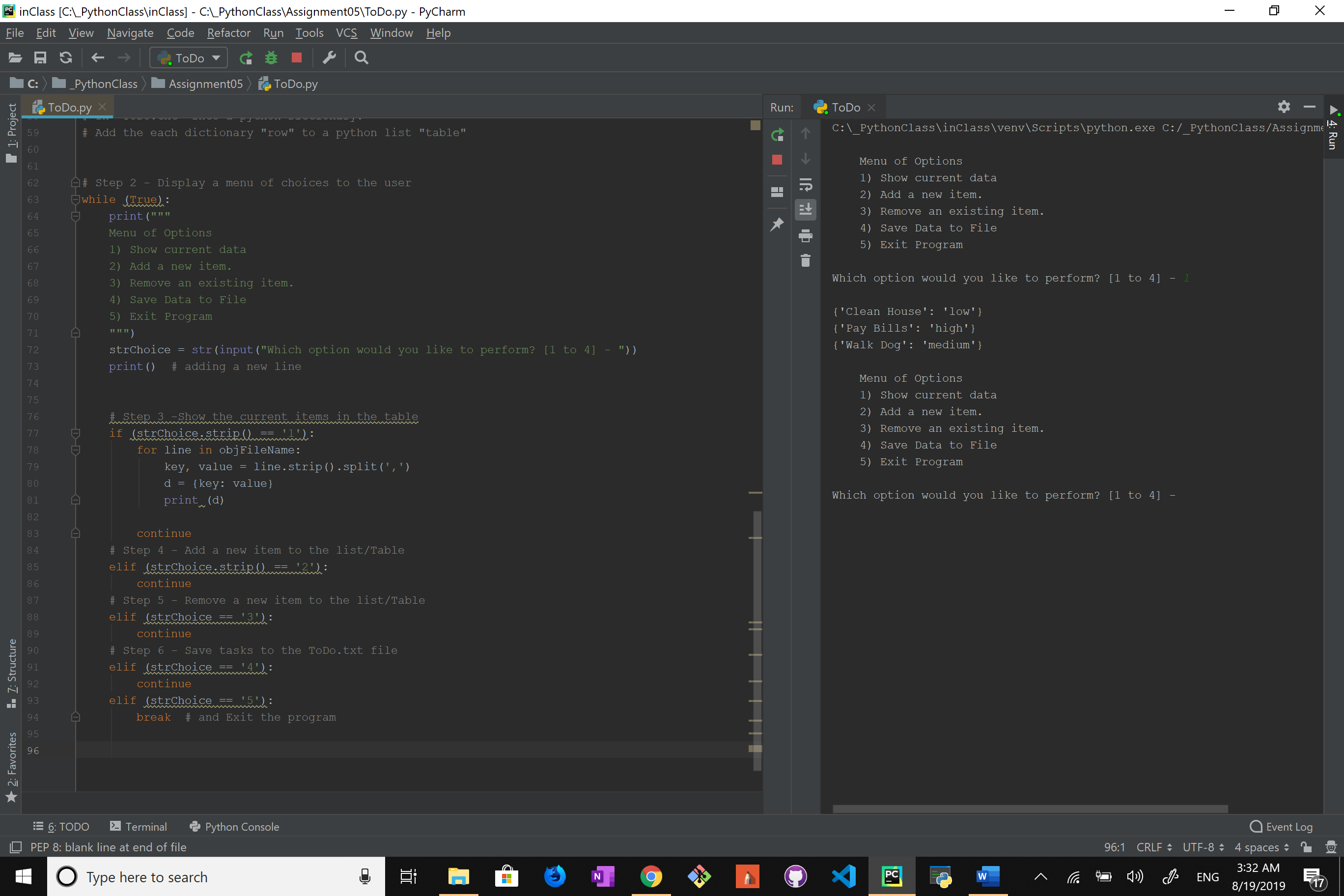
Python List: *“In short, a list is a collection of arbitrary objects, somewhat akin to an array in many other programming languages but more flexible. Lists are defined in Python by enclosing a comma-separated sequence of objects in square brackets ([])– John Sturtz (2018). Lists and Tuples in Python.* [*https://realpython.com/python-lists-tuples/*](https://realpython.com/python-lists-tuples/)

To start off we already had a document created for us with pseudo code and the menu already set, I simply added some comments to show my contribution and created the text file where the data will be stored.

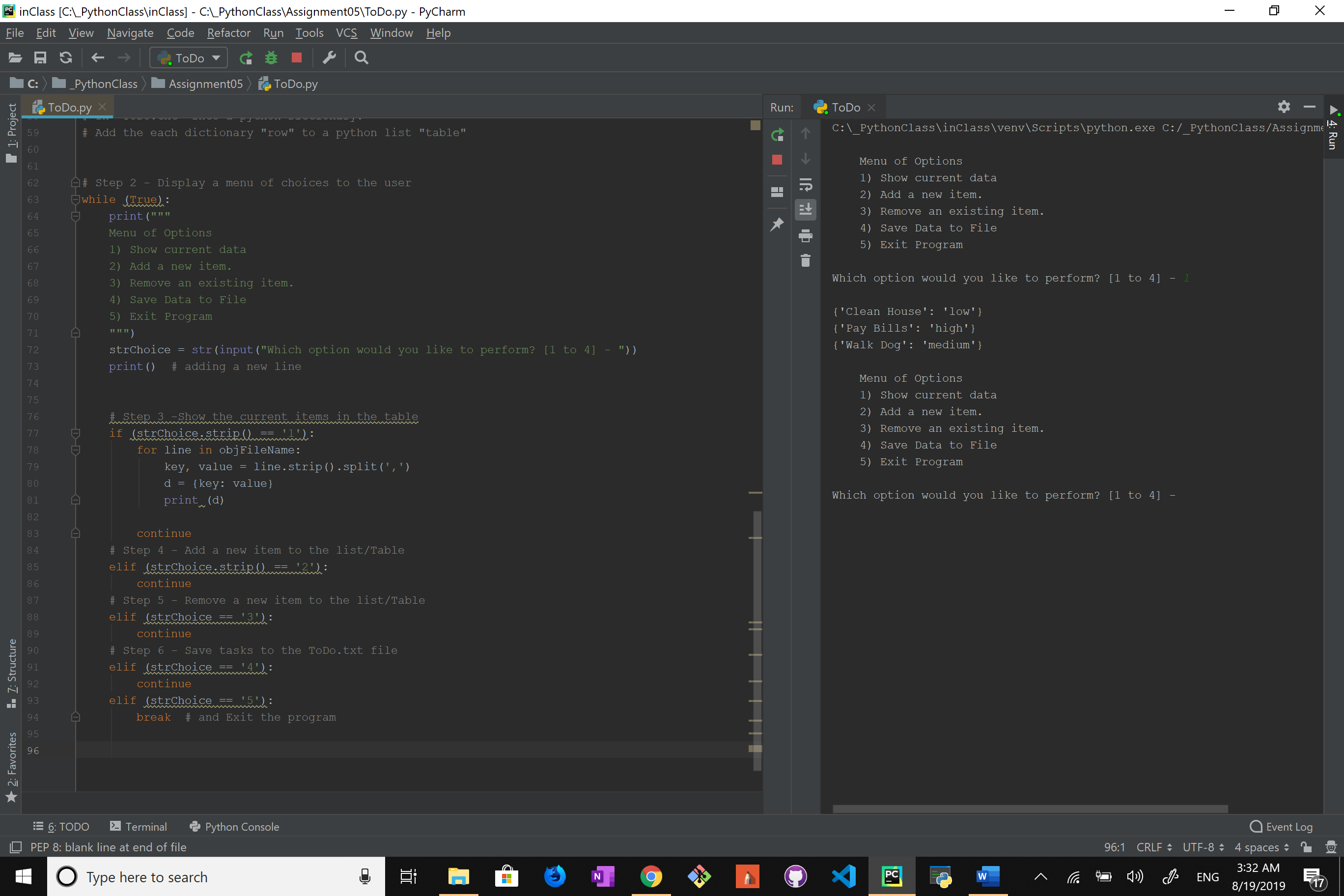
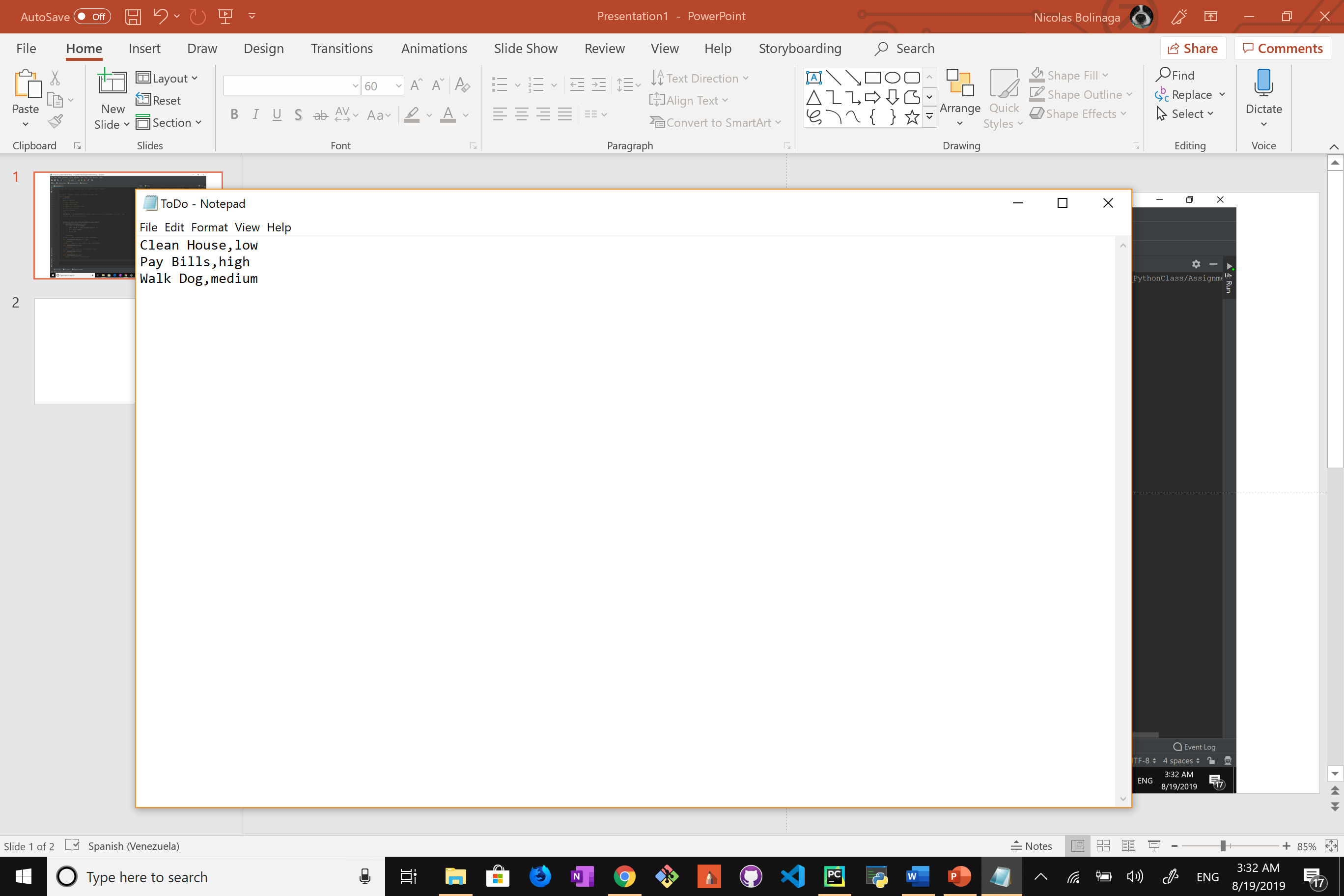


Next I dived right into coding by creating the first feature of the program, which is being able to read data from the .txt file. I also went ahead and made each row of the text file a dictionary.

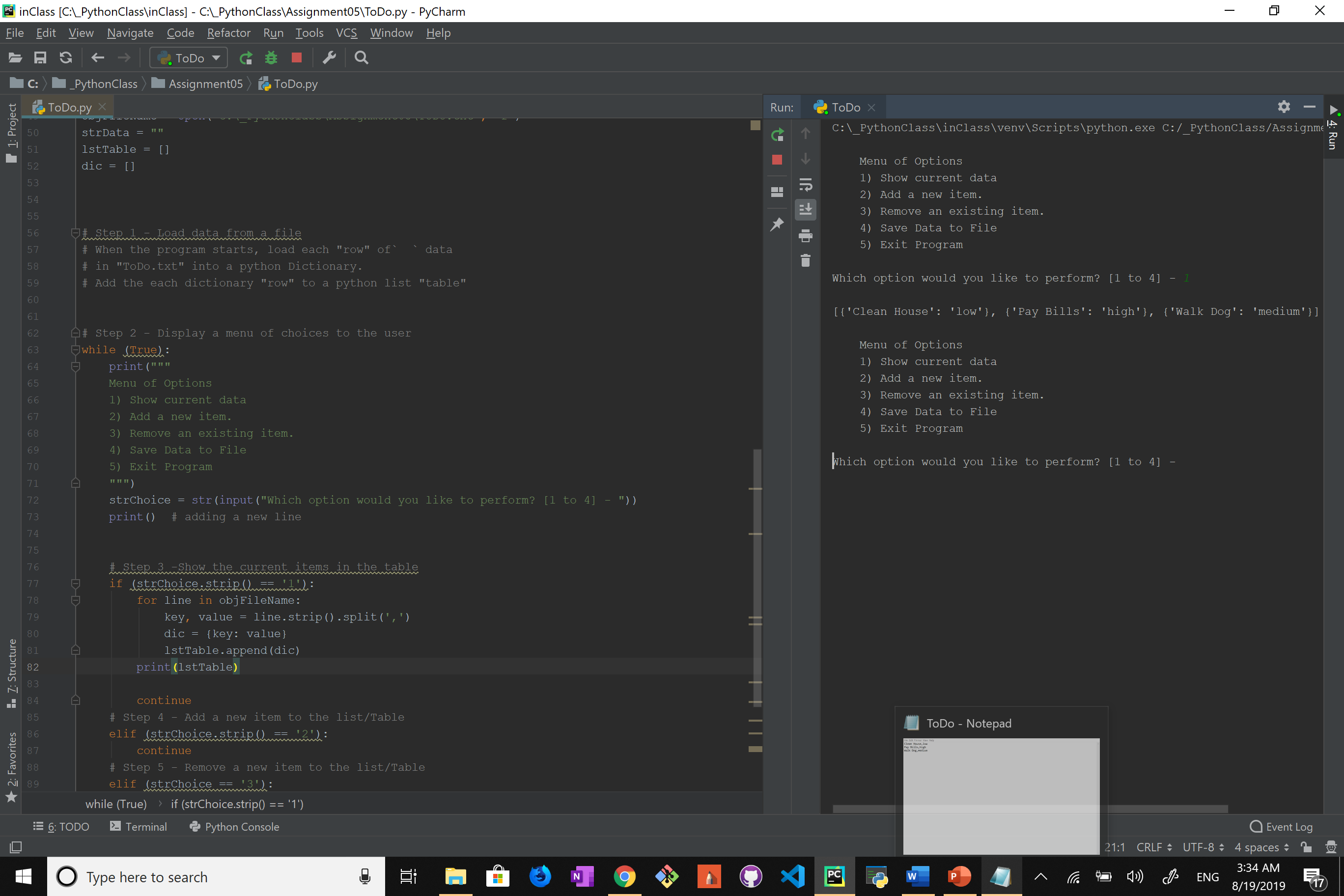
Here objFileName is the .txt file.



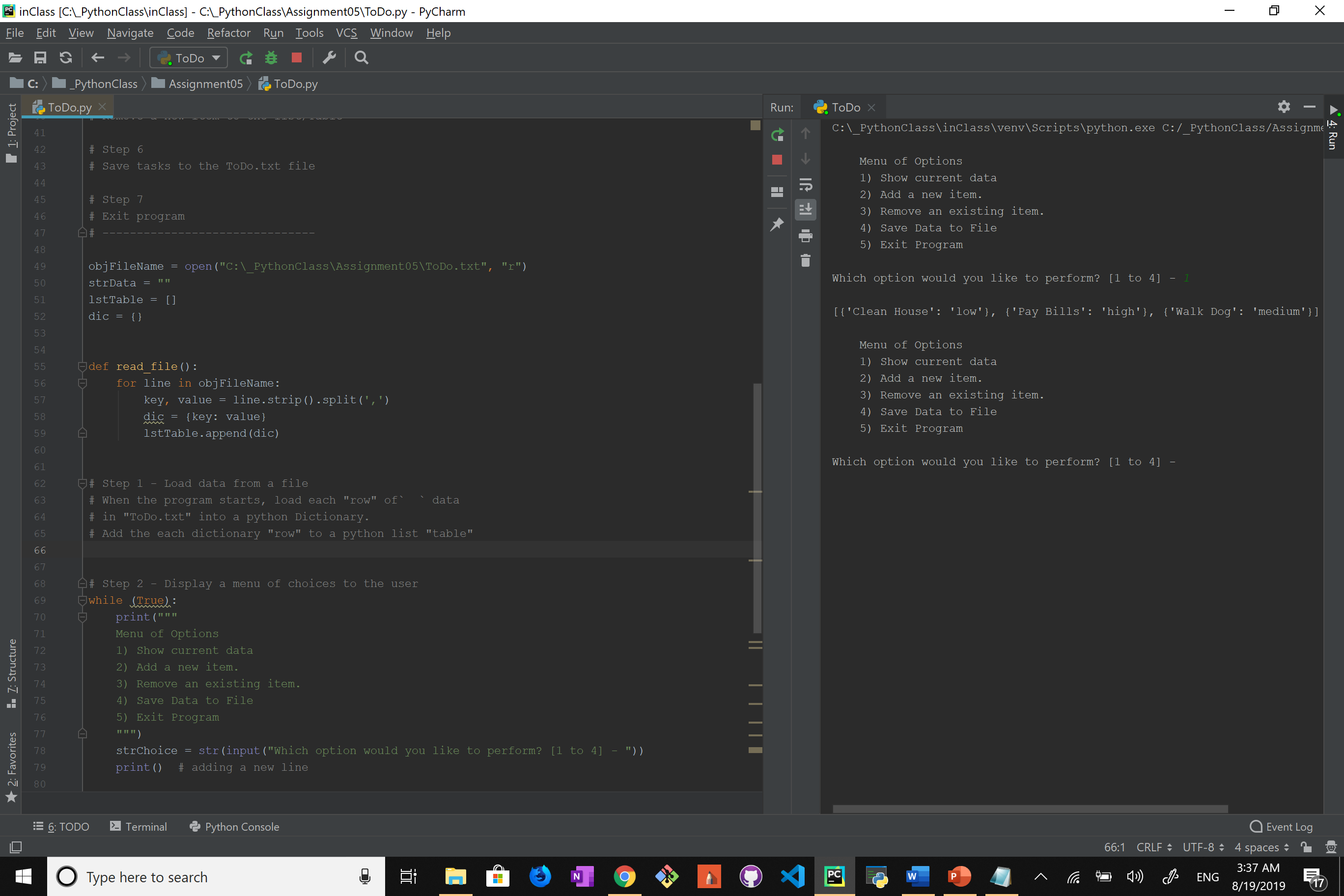
And this is the result:



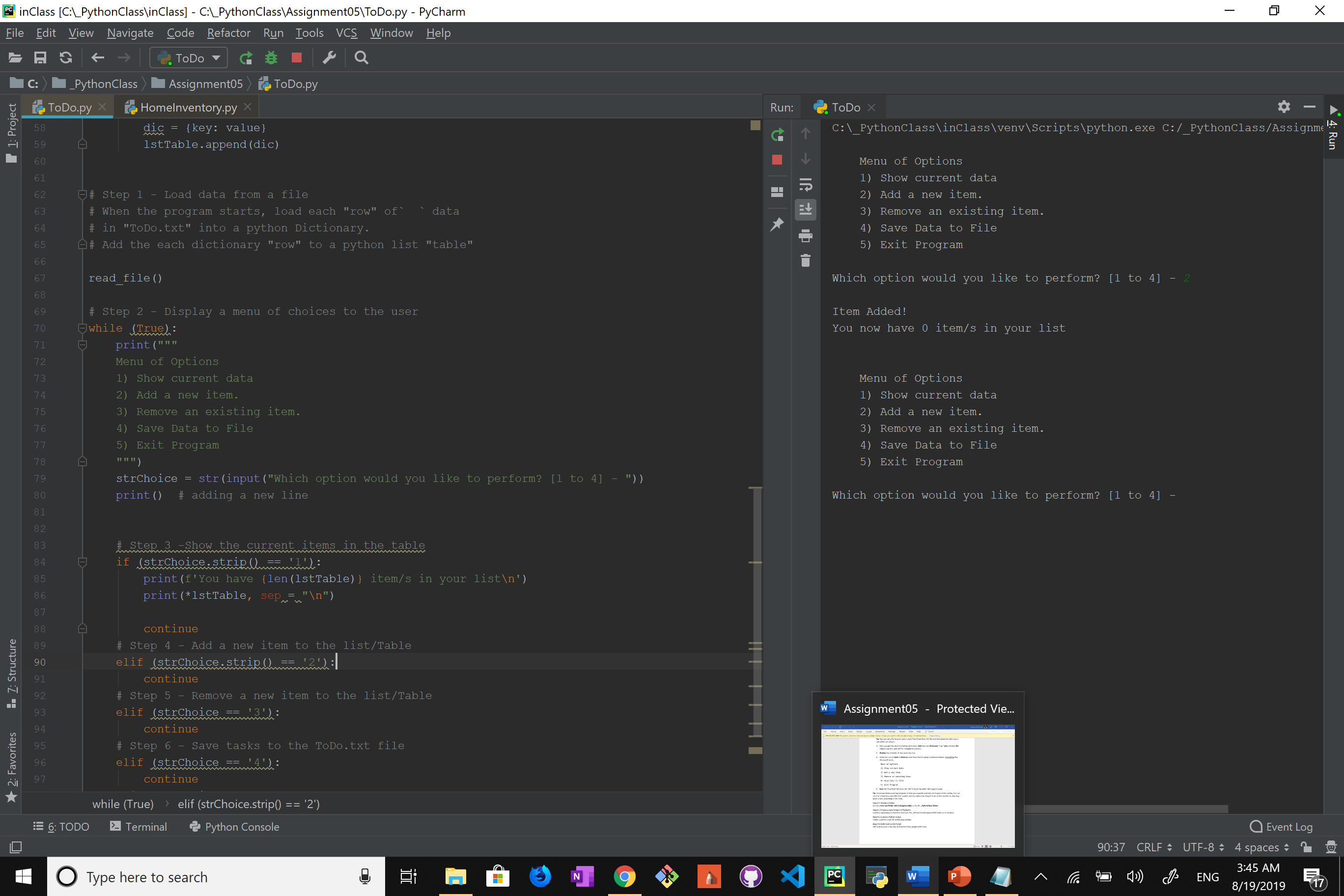
Now I had to append those dictionaries into a list.



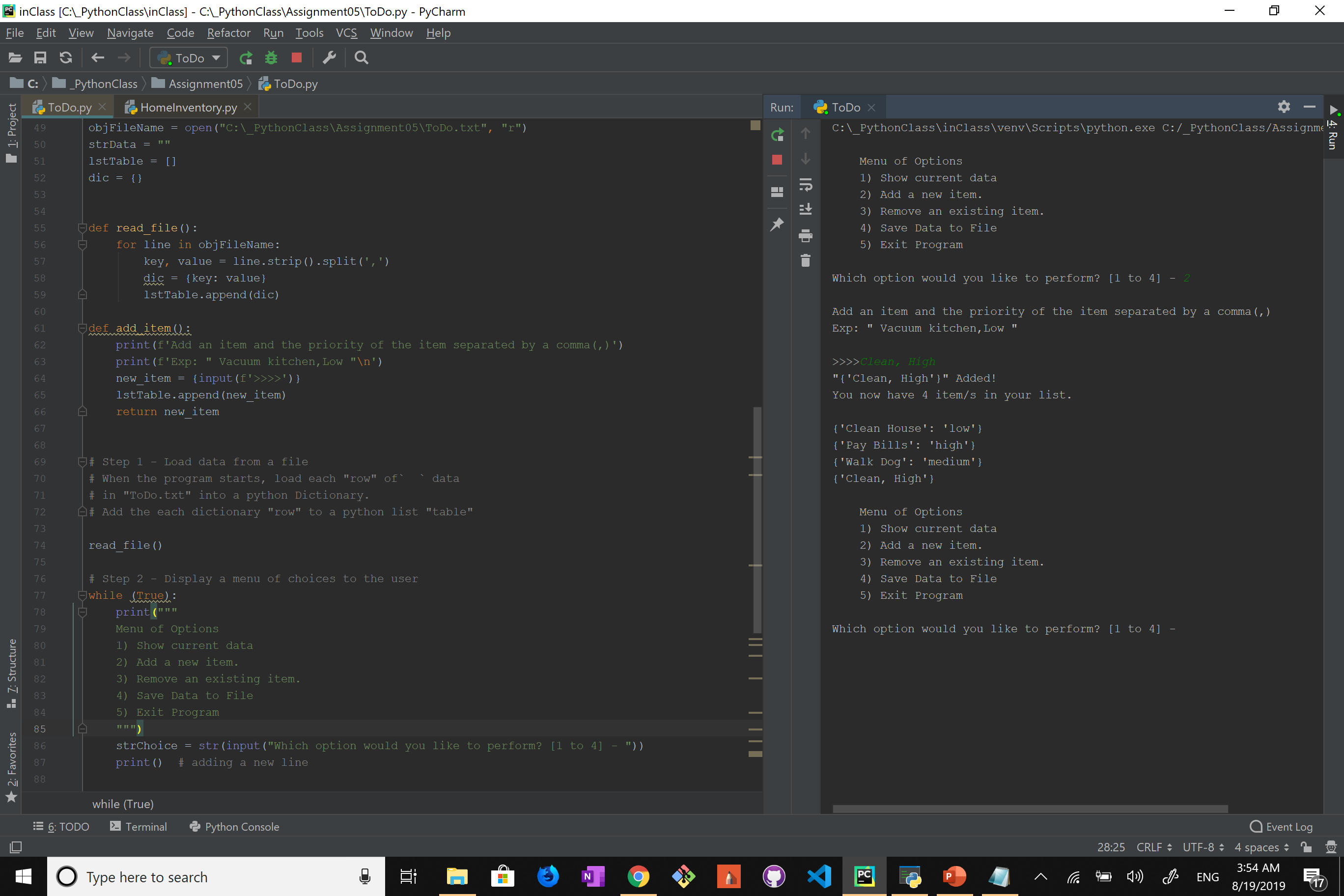
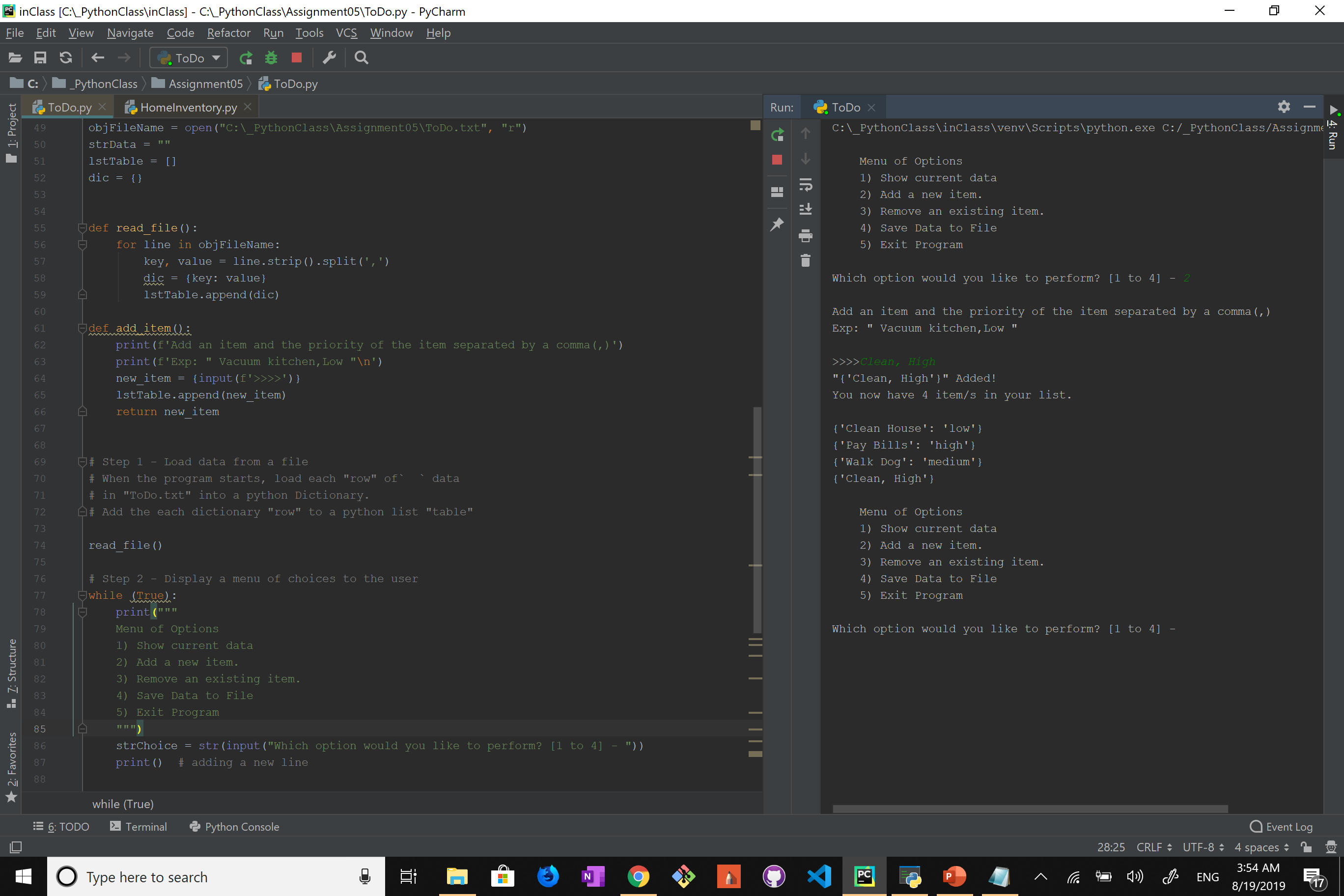
With this code we had the first part of the program done (some changes are going to be made later, but for now it works) I decided to move this part into a function to make it cleaner and just in case we need to do this various times we can just call the function.



I added one line of code to make it more readable when printing the list to the console. I Also took the function call out of the “if” statement, so it’s called every time the program is opened and there is no need to wait for the user input to read the data.

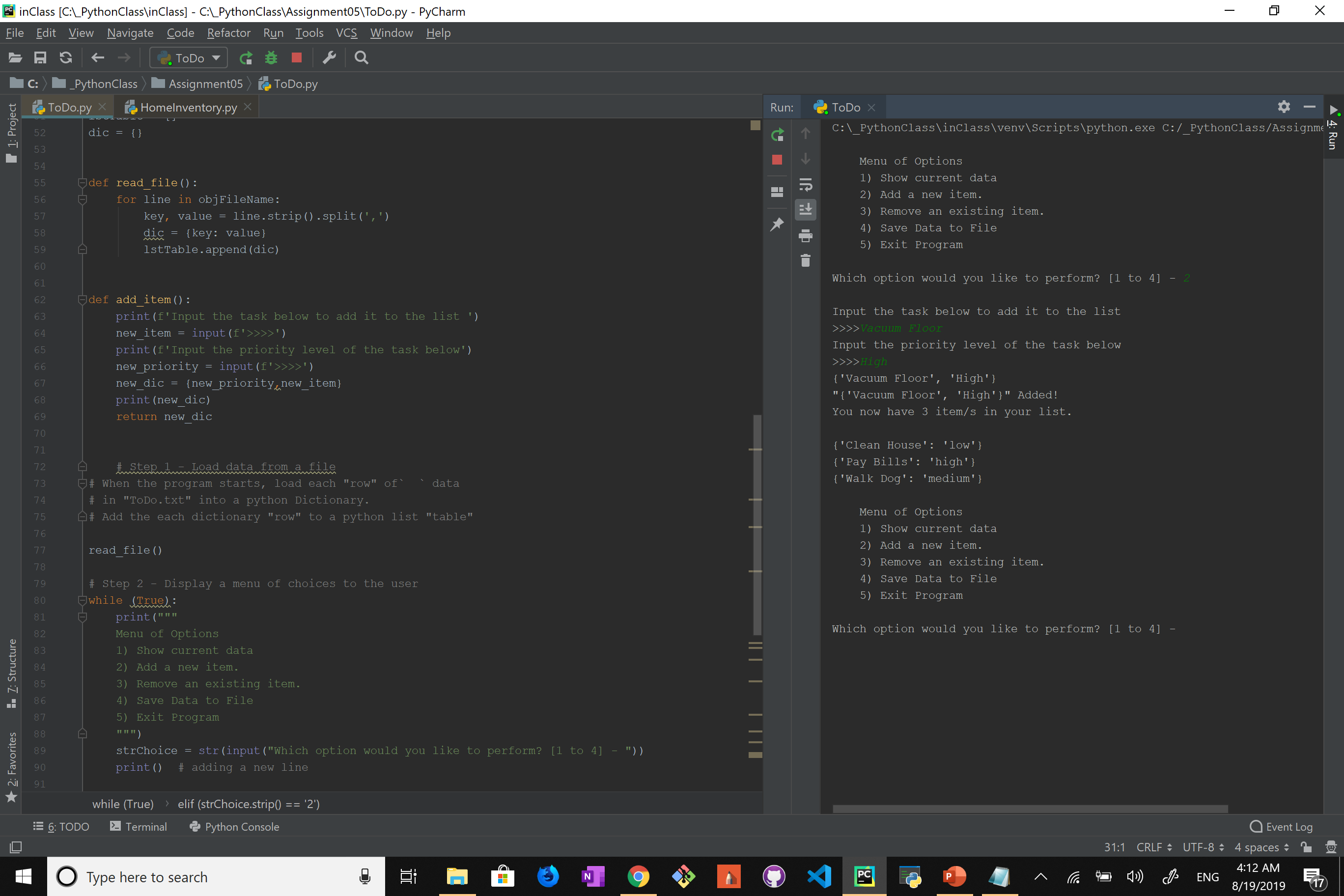
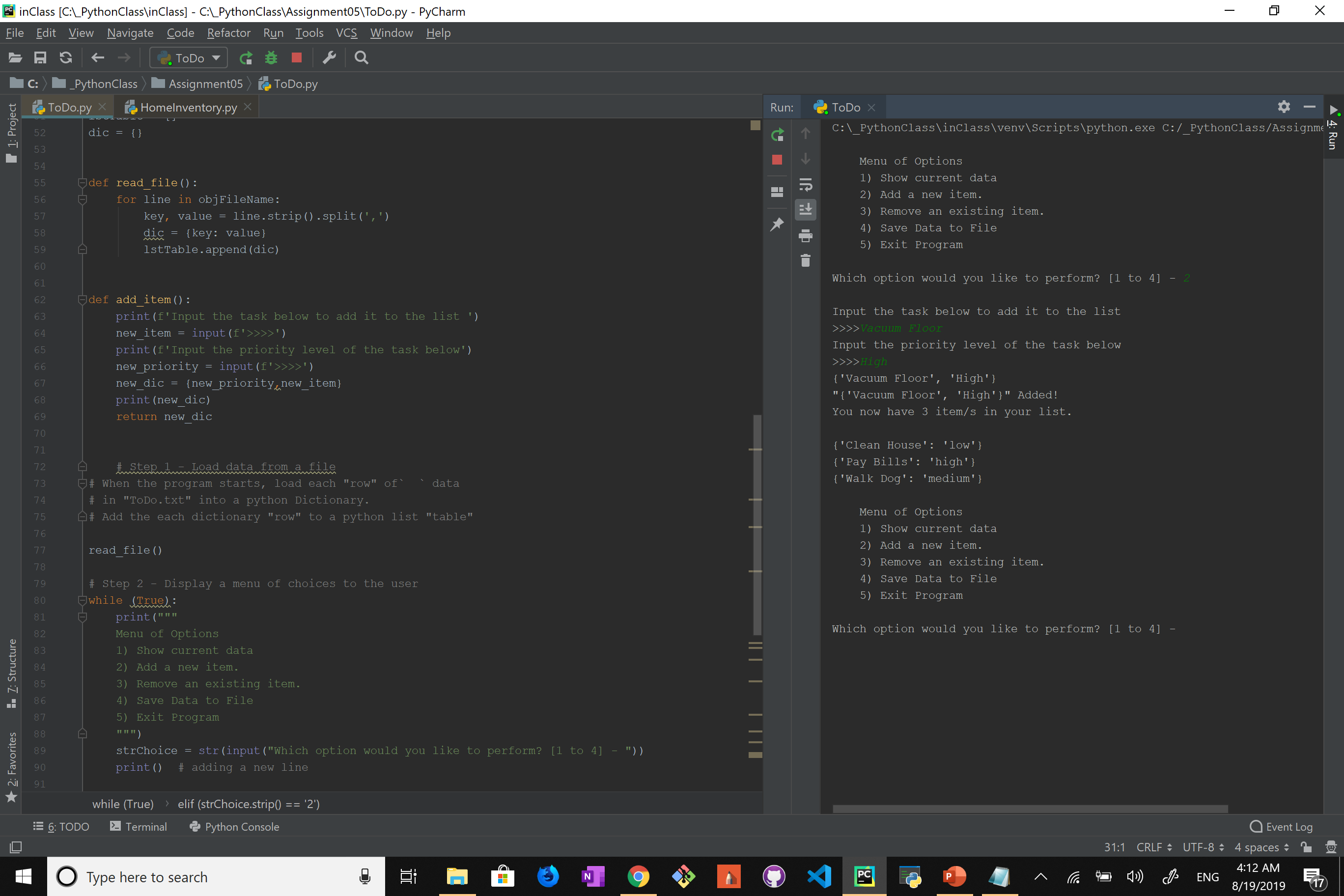


Now it was time to move to the second feature of the program and that is to allow the user to input their own data into the text file.

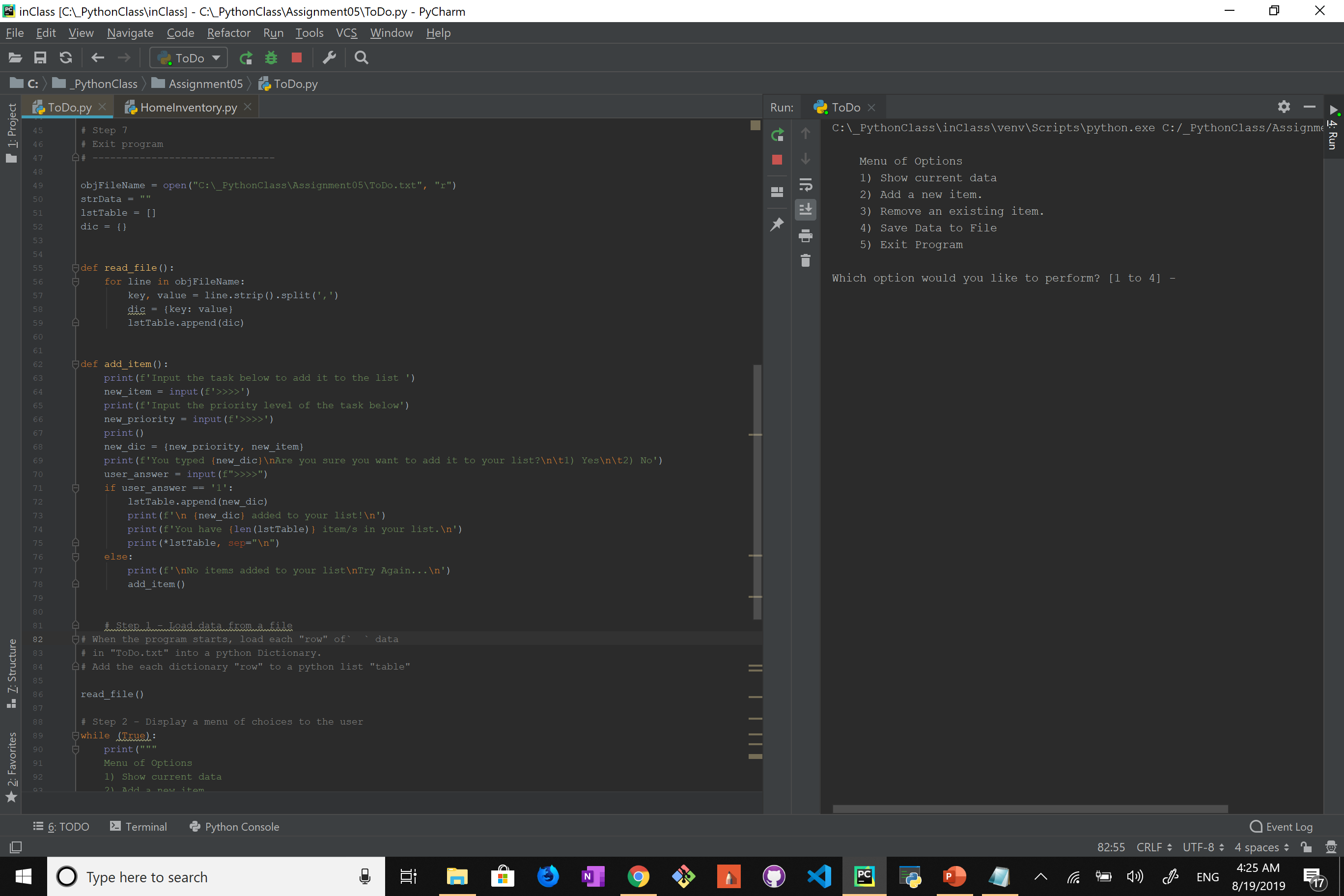


Even though this works it can run into trouble if the user doesn’t input the data correctly, so to fix this I decided to separate the user input into two, so now it ask for the key first and then the value and compiles it into a dictionary and then appends that dictionary into the list.

I found that sometimes the program would switch the key and the value around (mainly if the input has a space in it) I don’t know why this happens and I couldn’t figure out how to fix it.

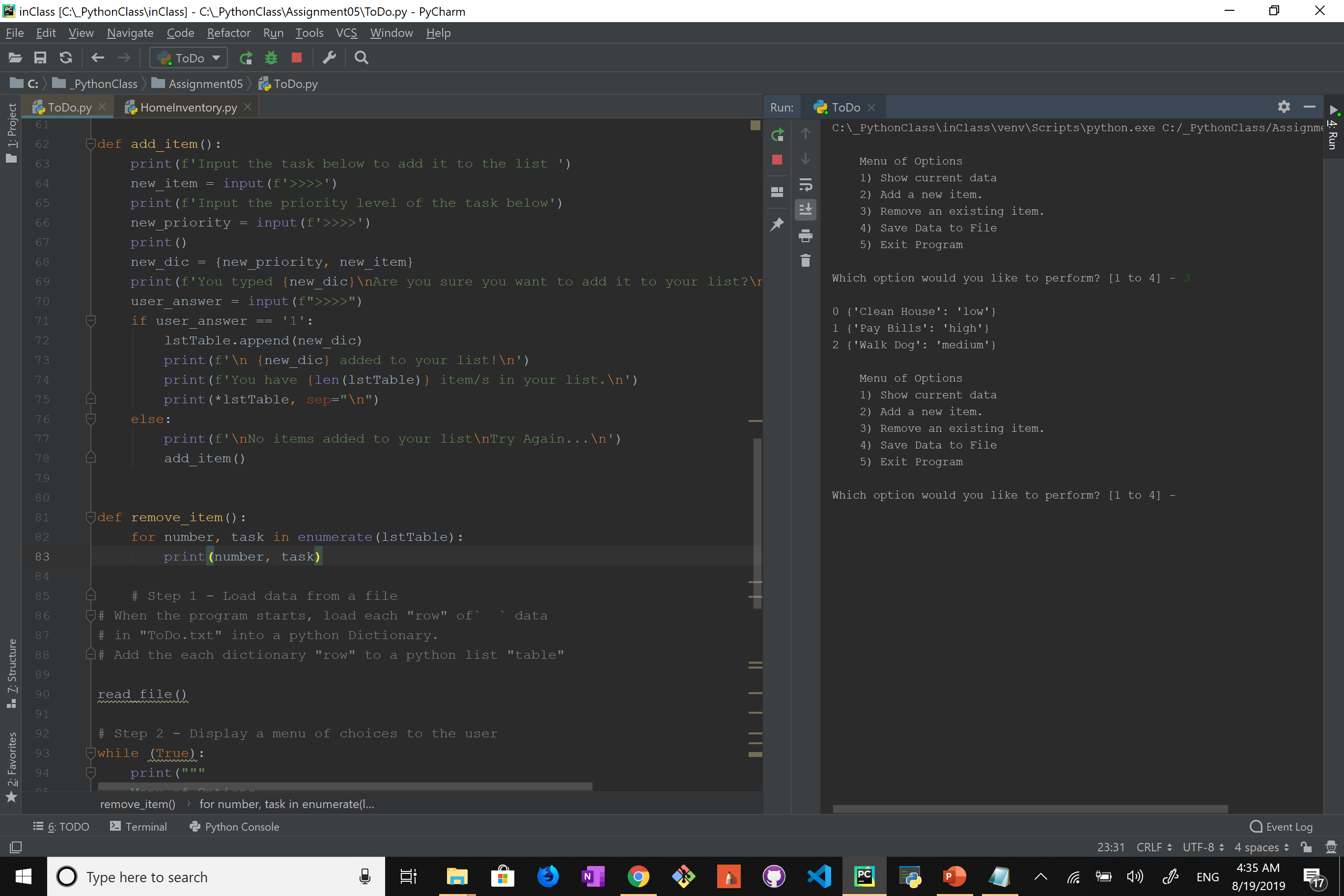
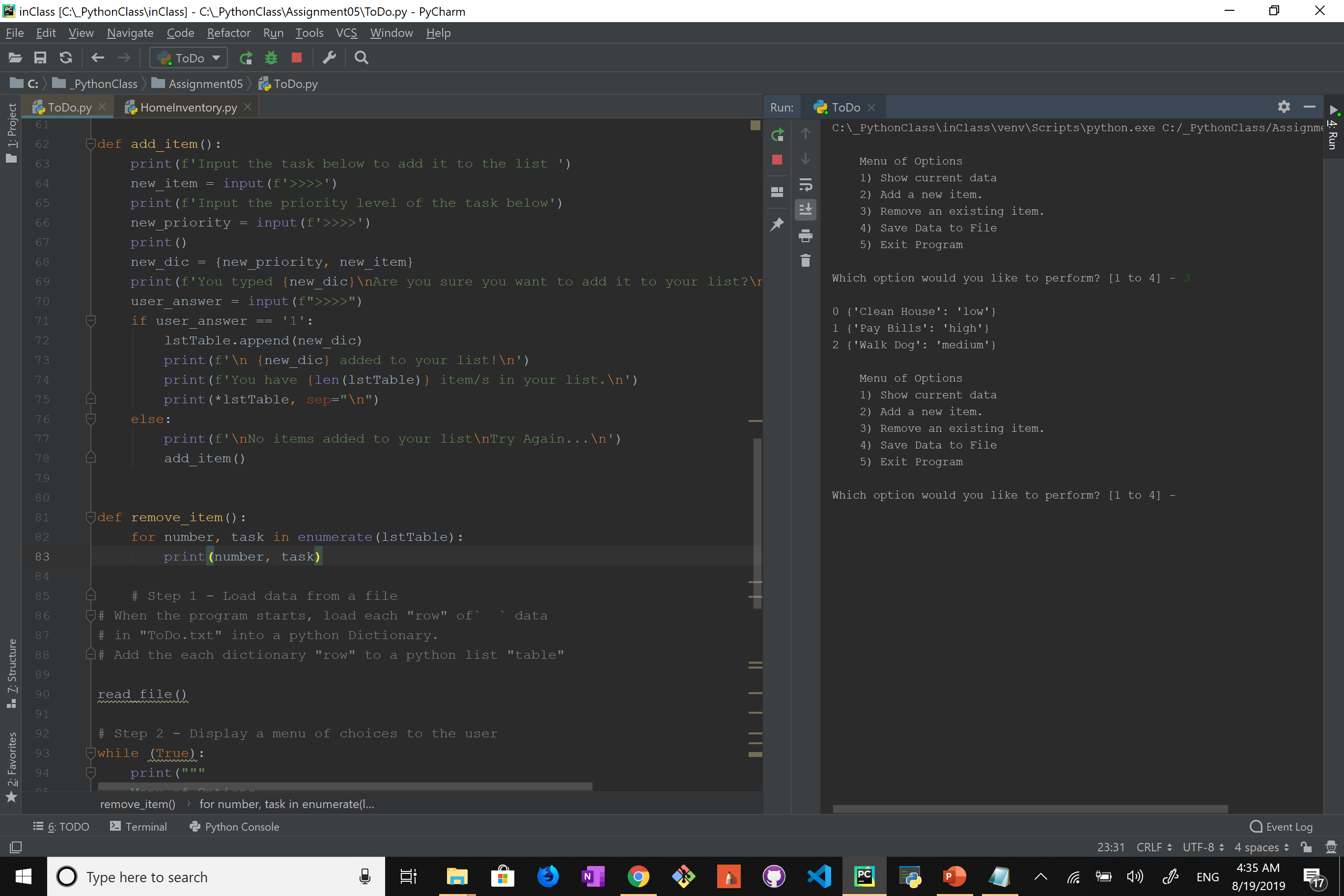


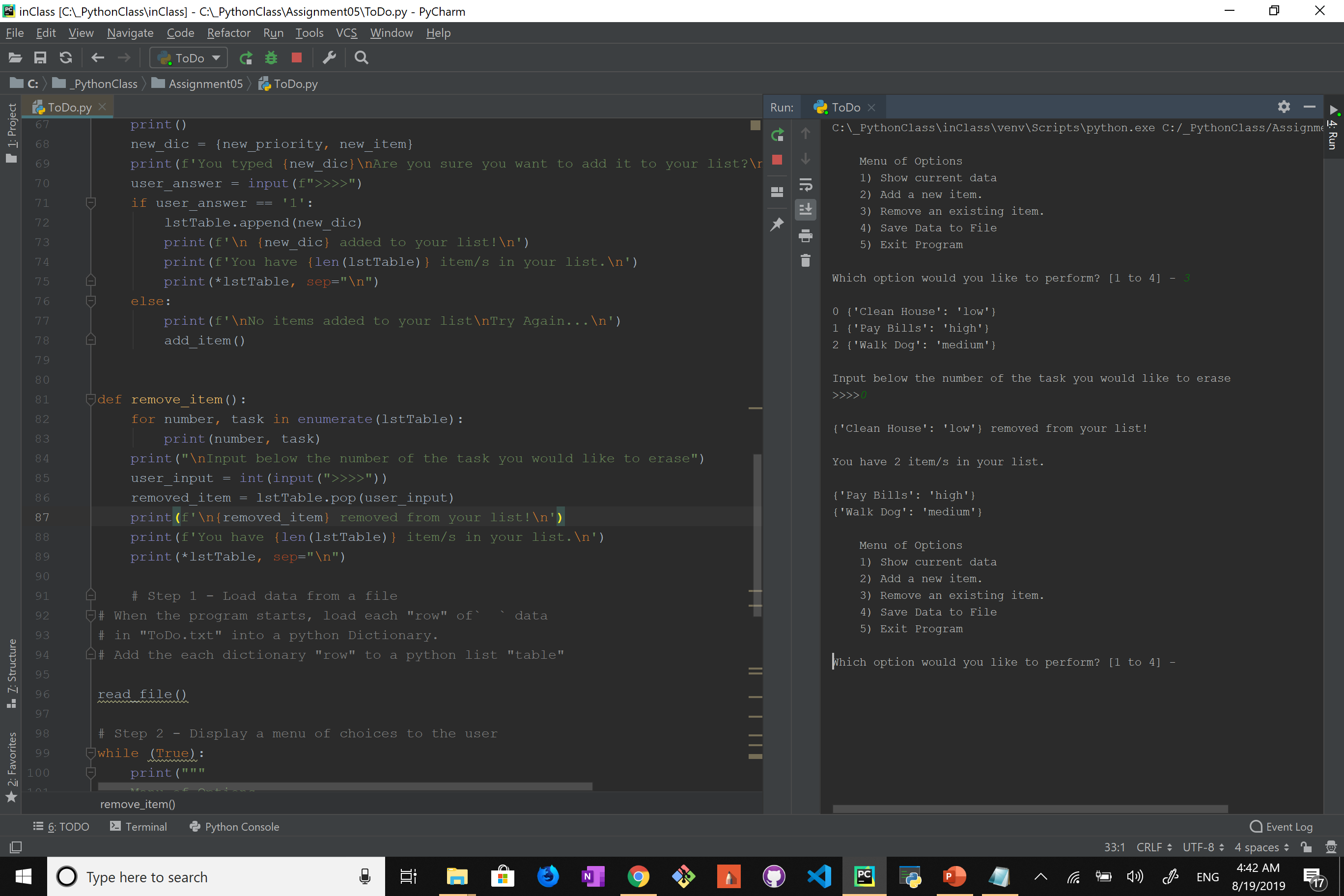
To finish off the “add\_item” function I decided to set an “if/else” statement so the user can approve and make sure their input is correct before the new dictionary is appended to the list.



Next it was time to add the option for the user to remove one of the items from the list. I found that the best way is to give each item a number (their position) and ask the user to simply input the number of the task they one to remove.

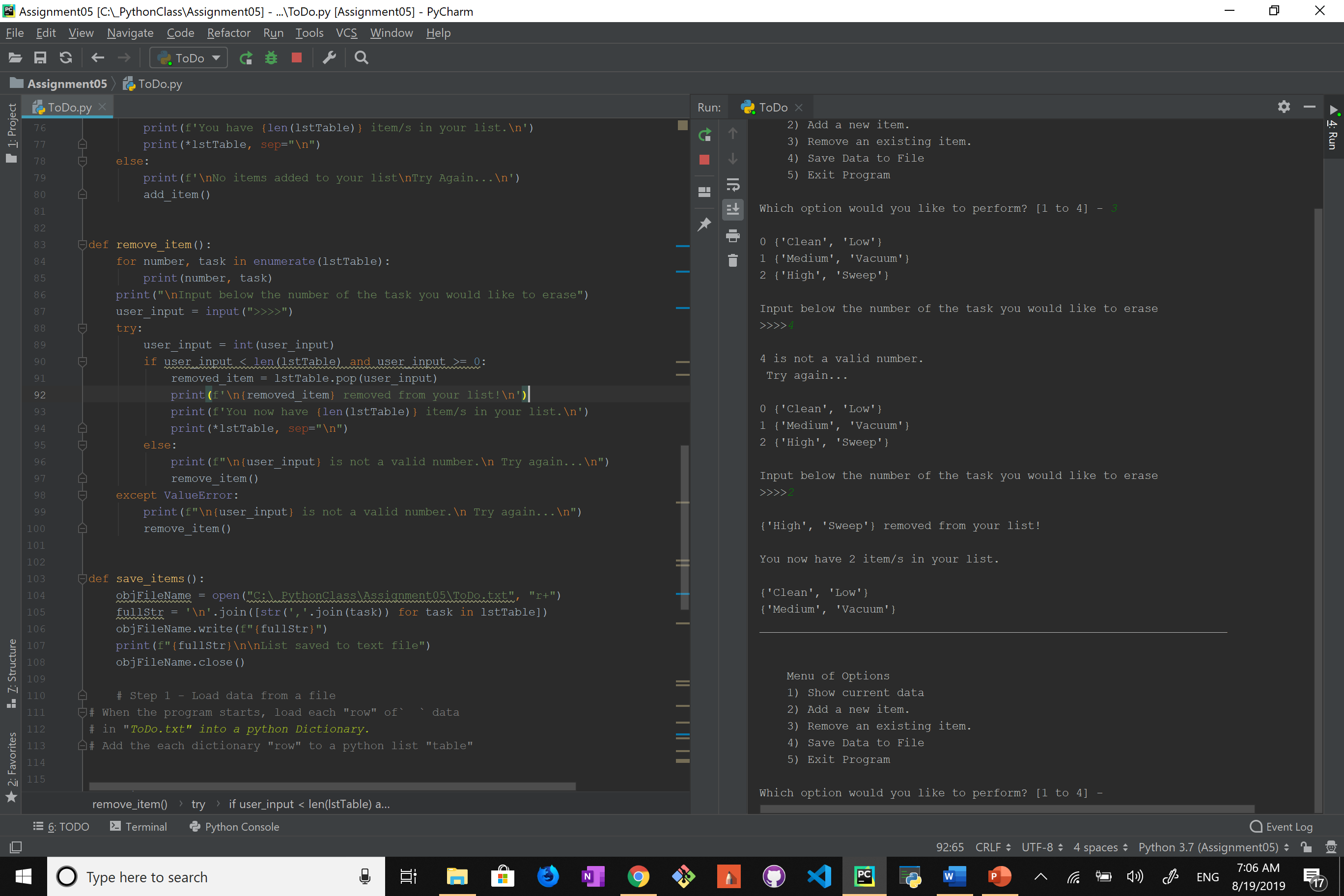
Here is how I did to print their position:



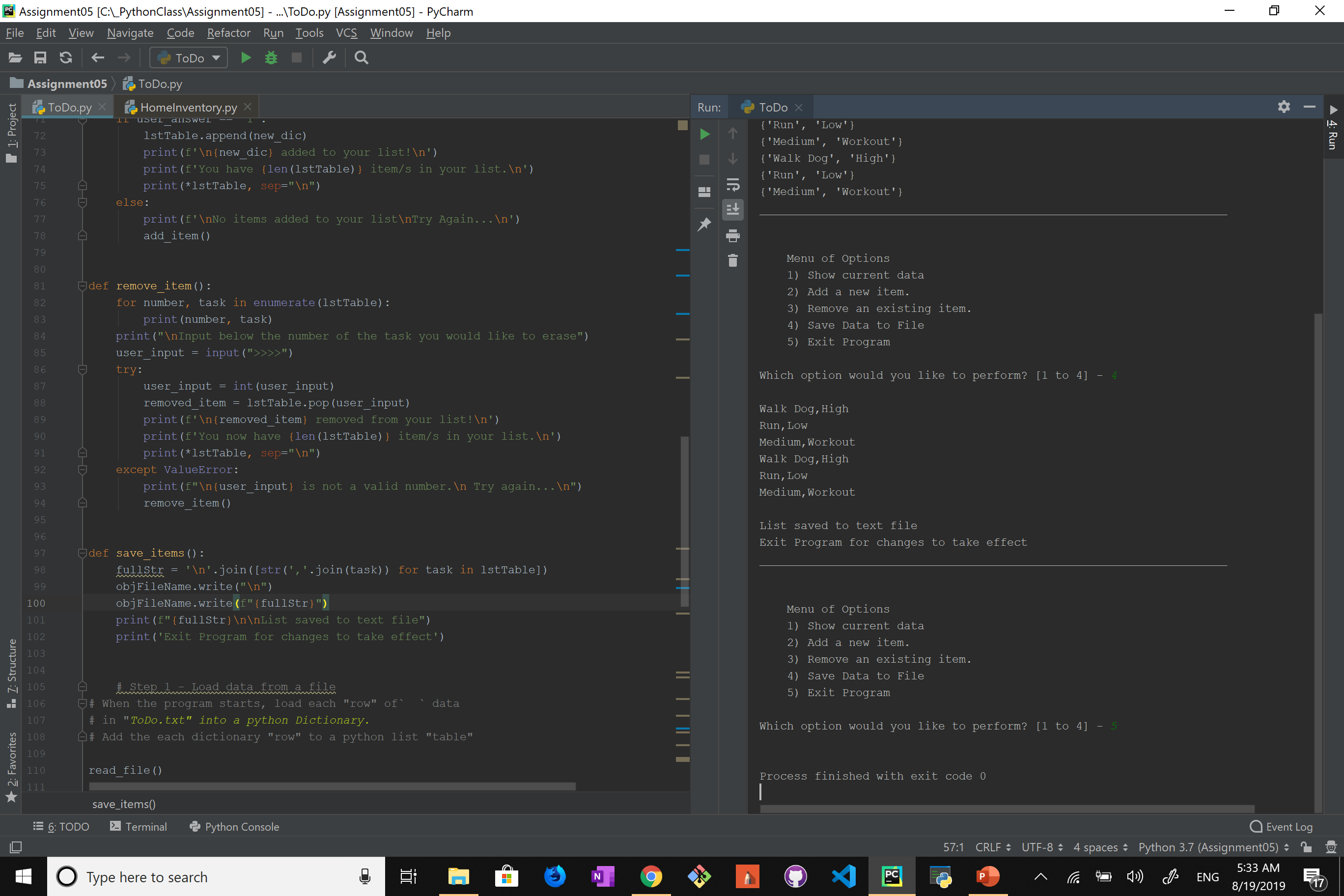


I then added the input option and used “pop” to remove the selected item from the list, also some messages to show that the item has been removed successfully.

The problem with this is the same one I had before where if the user inputs an incorrect number or a string the program fails. So, by doing some research I found a way to guarantee that the user input a number and that number is within range.



Finally, the last option is to save and write the data to the .txt file. After some more research I found this to be the best way for me to do it.



Everything was working fine but playing around with the program I found some major bugs;

* New list just writes itself below the old one.
* Erasing an item didn’t remove it from the txt file.
* Blank lines cause the program to not work.

I fix all three errors by simply wiping the txt file clean after the data is already stored as a list inside the program, all modification made to the data happen within python and are not passed to the file until the user chooses to. I also added comments within the code to make it easier to understand.

