Alyssa Matthews

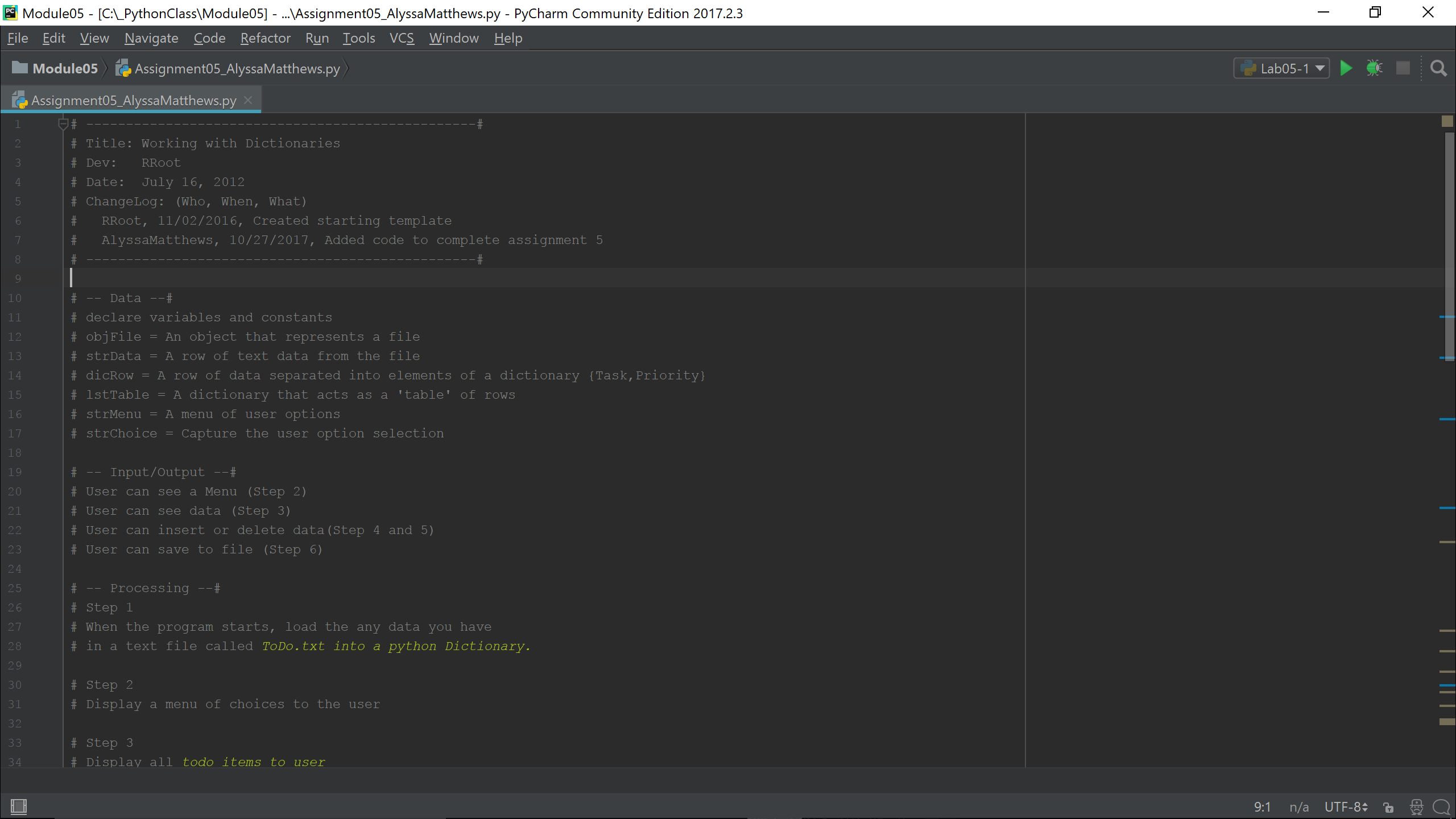
Inst. Randall Root

IT FDN 100: Foundations of Programming – Python

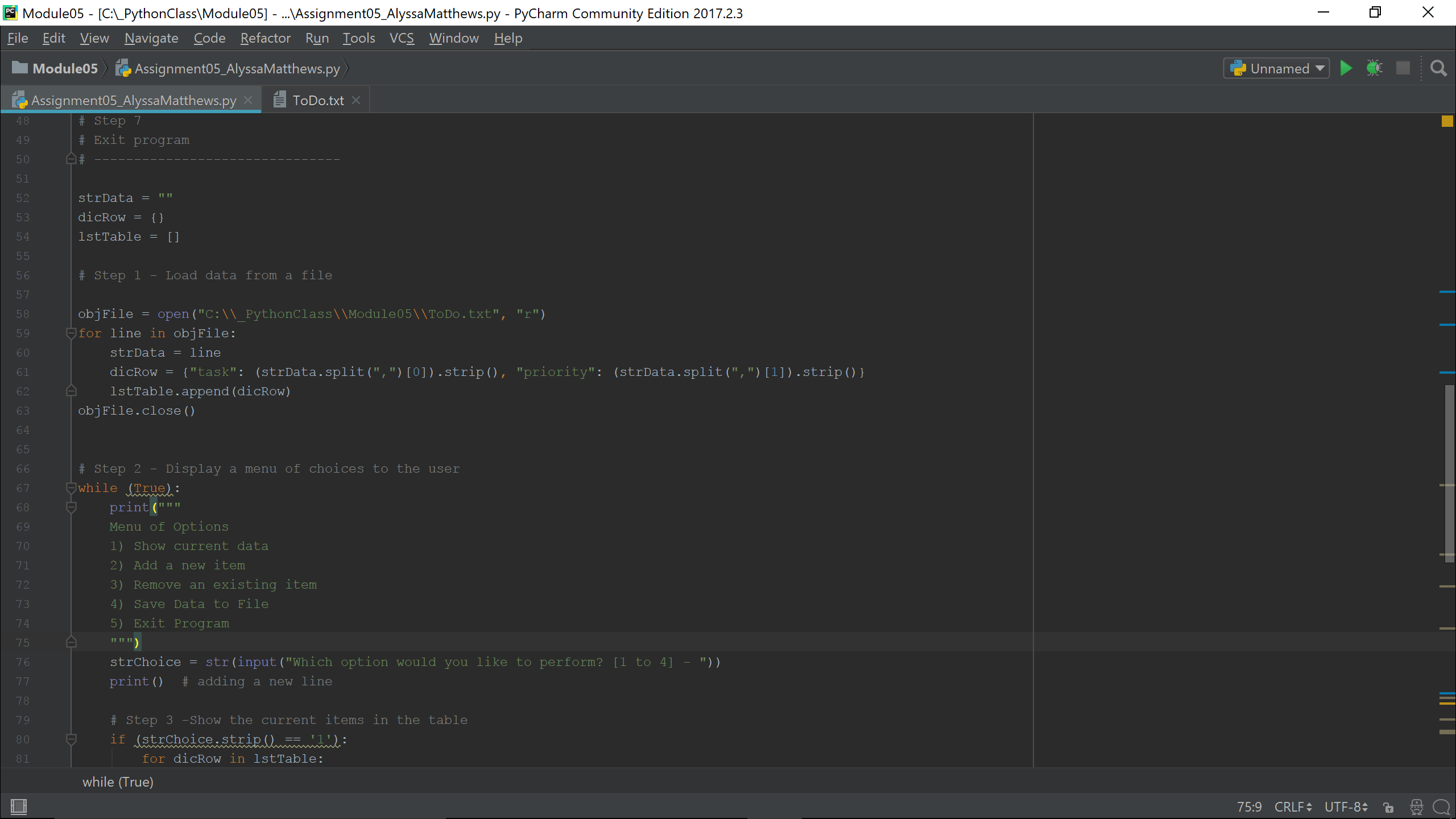
October 27, 2017

Assignment 5 – Creating a To-Do List Using Dictionaries

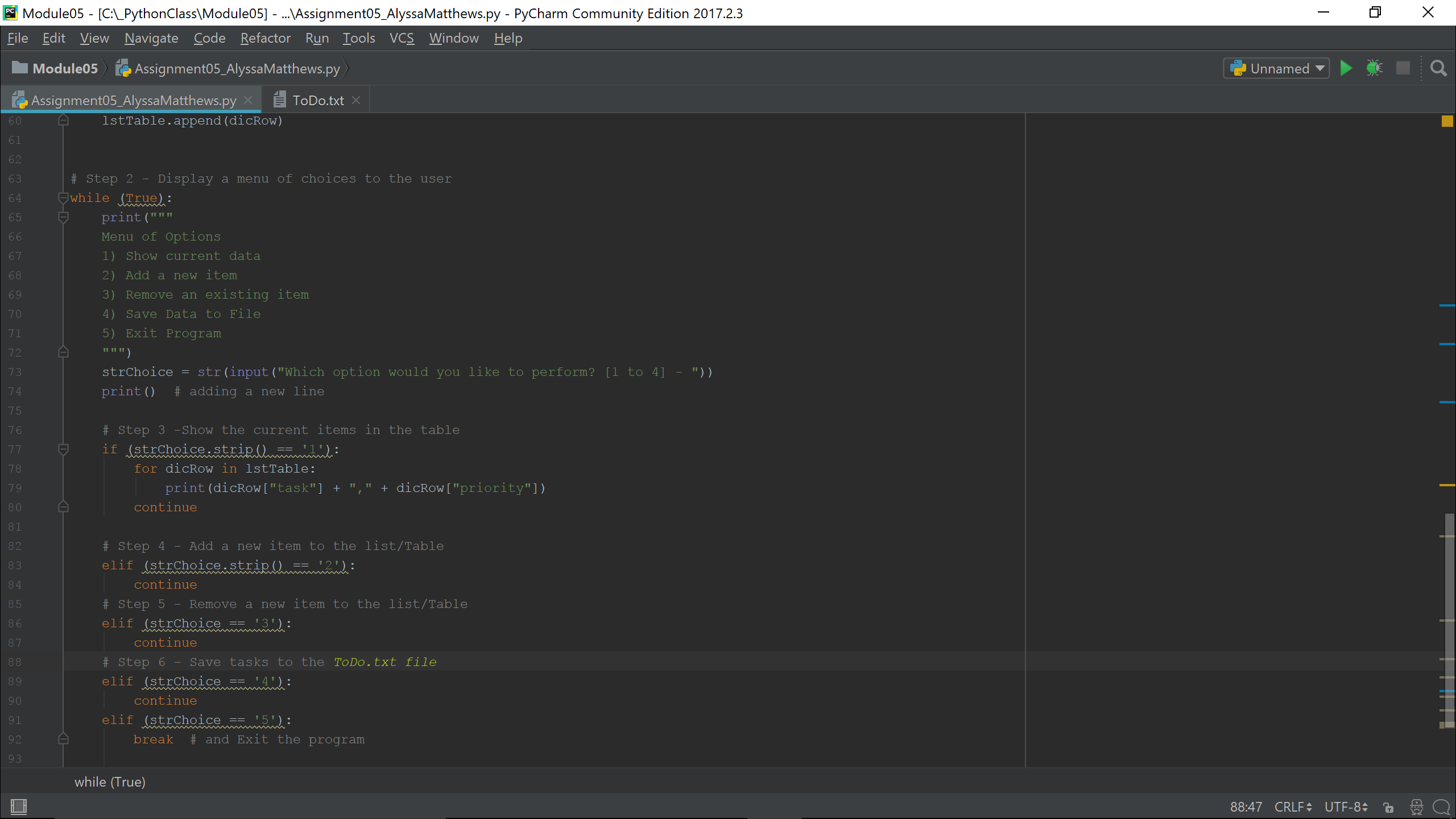
To begin, I used the starter code provided to set up the assignment and updated the header information. I also created an initial text file ToDo.txt and seeded it with the initial to-do list items.



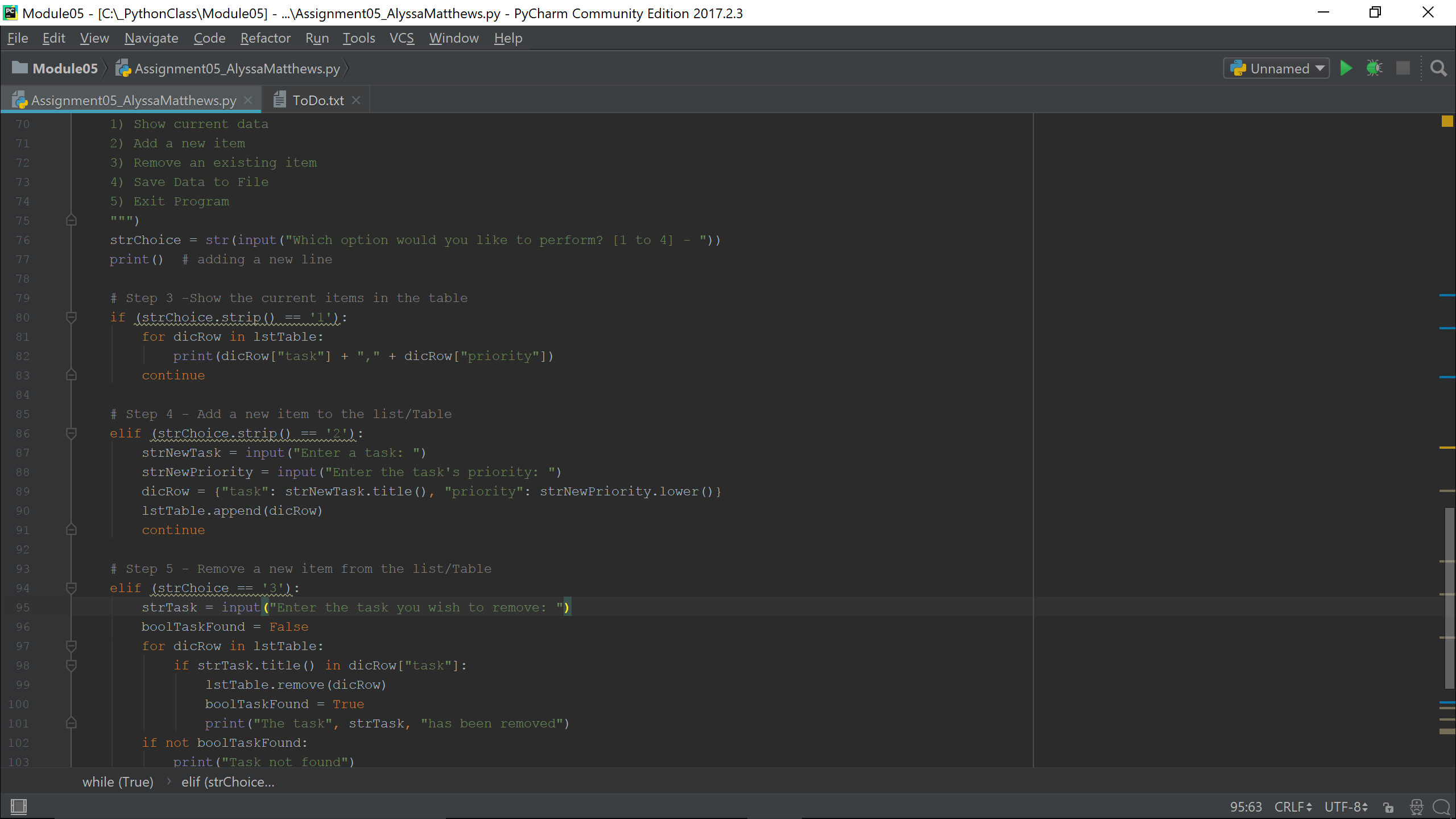
Next, I added code to opens the existing text file and read the contents to the program. For each line item in the file, it reads the line to a string, splits each string at the comma into values to be associated with the keys “task” and “priority”, and strips the string of any extraneous spaces or carriage returns. Finally, the program appends the two-dimensional list (or “table”) with the new dictionary entry. At the end of the ‘for’ loop, all entries from the text file have been read as dictionaries, and each dictionary (or entry) has been added as an item in the table list. I then close the data file:



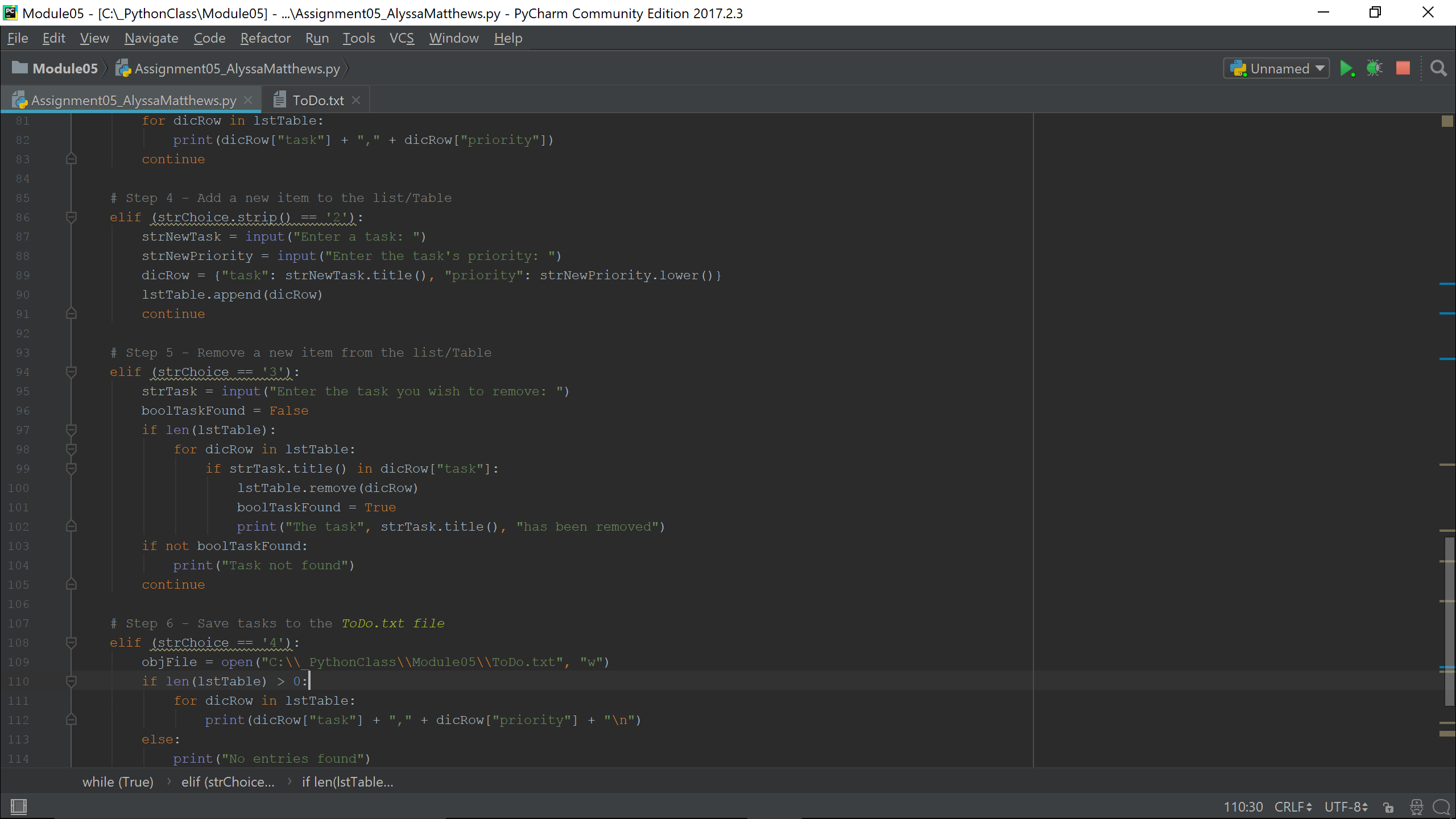
I then added code for user option 1, allowing the user to view all existing table data by printing the value associated with the key “task” and the value associated with key “priority” for each dictionary in the list:



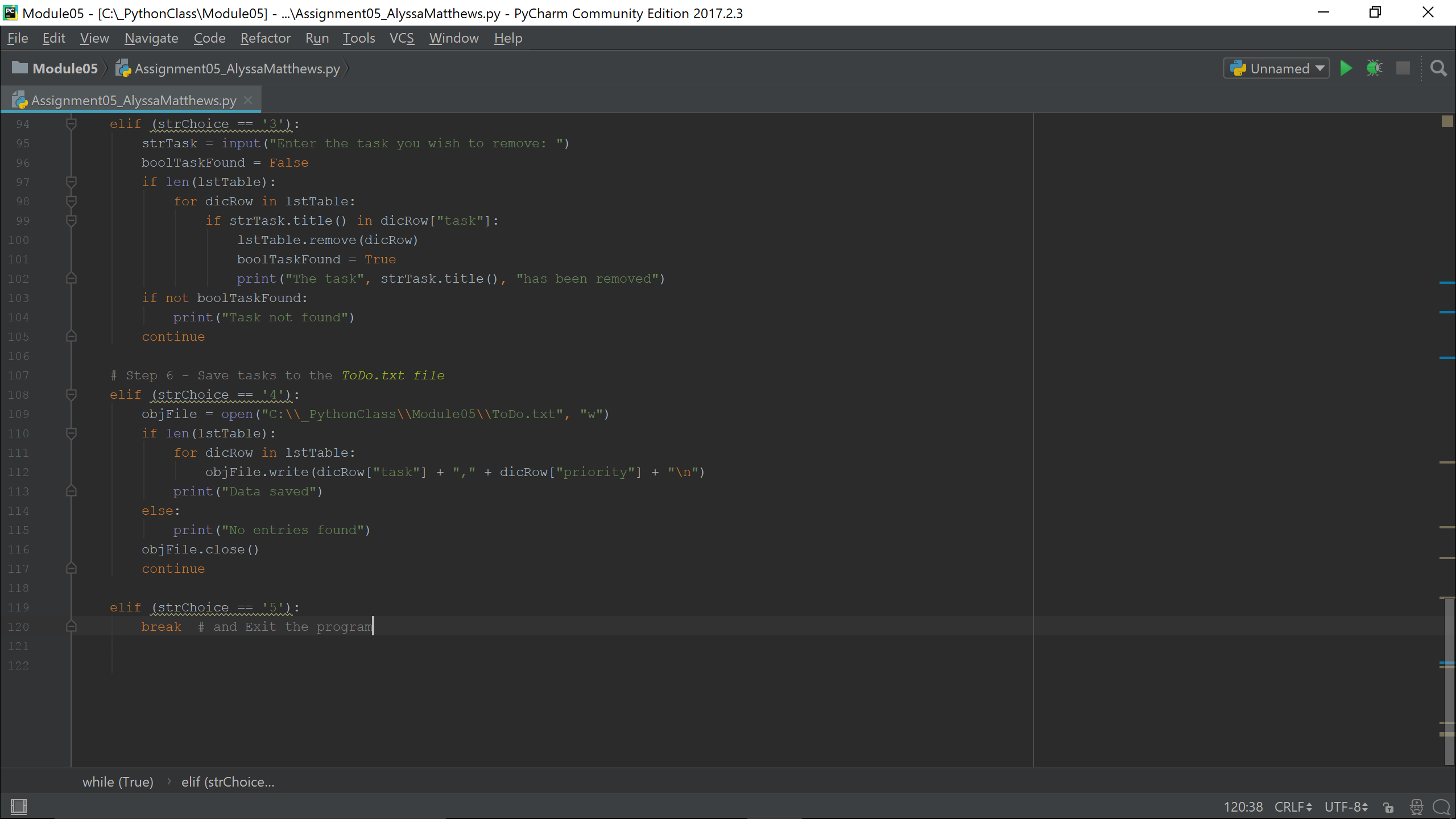
Next, I added code for user option 2, allowing the user to add new task items to the list. In this case, I added two new variables to capture the user’s entries for task and priority and store them in a new dictionary entry matching the existing capitalization formatting, then used the function to append that new dictionary to the existing list. (At this point I also added the two new variables to the ‘Data’ reference at the top of the code file.)



I then added code for the user to remove an entry from the list by providing the task name. I used a Boolean flag to track whether the task provided by the user had been found in a sub-dictionary of the list table, then looked for the user’s entry under each “task” key of the dictionaries in the list. If the list isn’t empty, and if the task is found in a given dictionary, that dictionary row is removed from the list and a readout informs the user that the item has been removed. If the task doesn’t exist (or if the list is empty), the user is notified that the task wasn’t found.



For option 4, the user can save the data back to the original text file. To do this, I used code to open the text file again with the “write” command this time, and if the list of dictionaries has any entries I used a ‘for’ loop to print the values of each dictionary on its own line. If the list doesn’t have any entries to save to the file, the user is notified. At the end of the process, the file is closed:



To finish, I spent some time adding, removing, and saving test data with the program, including testing how the program functioned when the to-do list was empty. Once everything seemed to be working properly, I saved the code file, this document, and the ToDo.txt file to a folder for upload. In addition to uploading the file to Canvas, I also uploaded the assignment to GitHub at the following link:

<https://github.com/AlyssaKM/IntroToProg-Python/blob/master/Assignment04_AlyssaMatthews.zip>