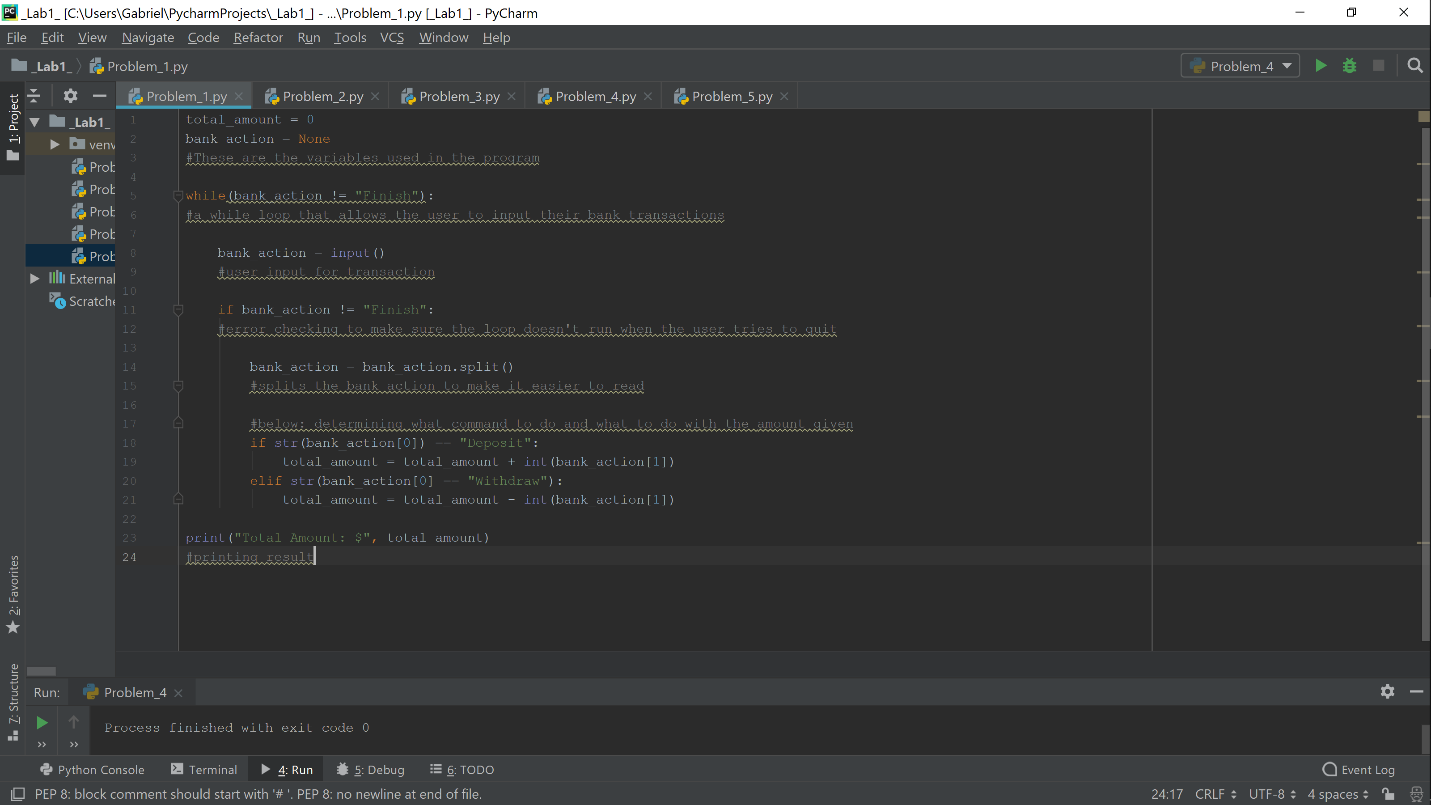
Wiki Page

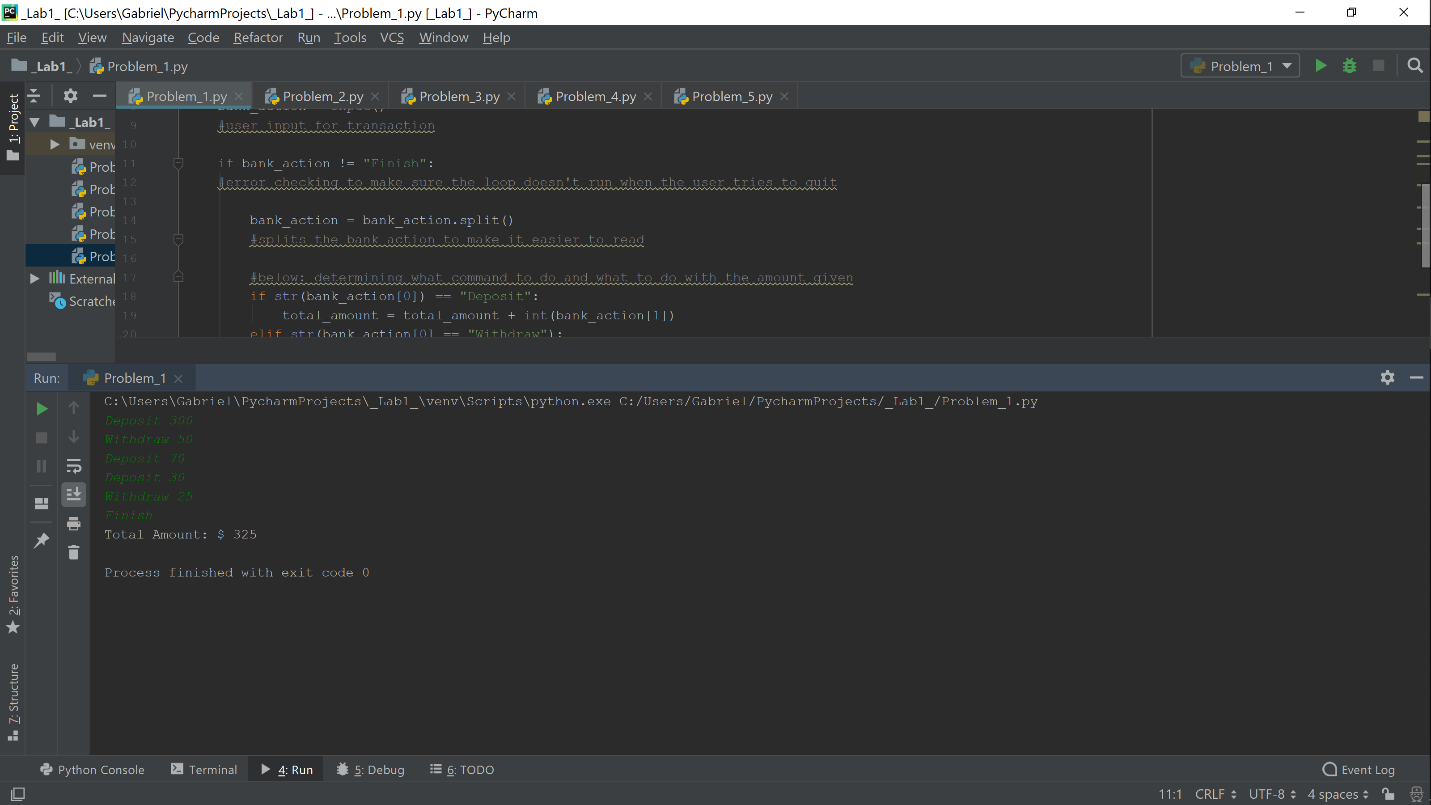
Gabriel Rhoads

Lab 1

Problem 1:

For this problem I had to make a bank account transaction system that took input from the user and made either a Withdraw or a Deposit based on their selection and the amount they requested. Here is the code and sample output for this program:



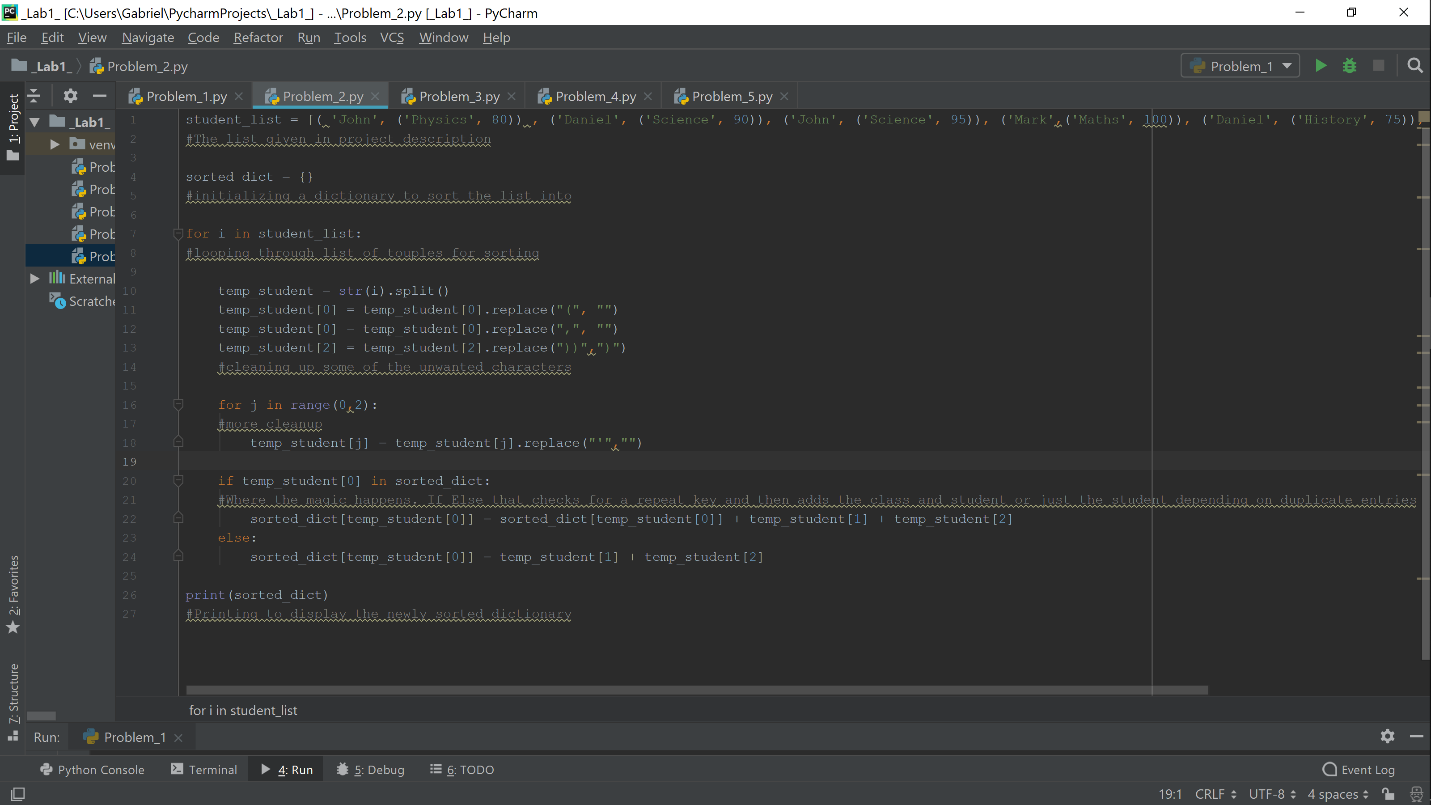


As you can see I used a simple while statement to create the transaction loop and then evaluated the users choice with an if else statement and balanced the bank account accordingly.

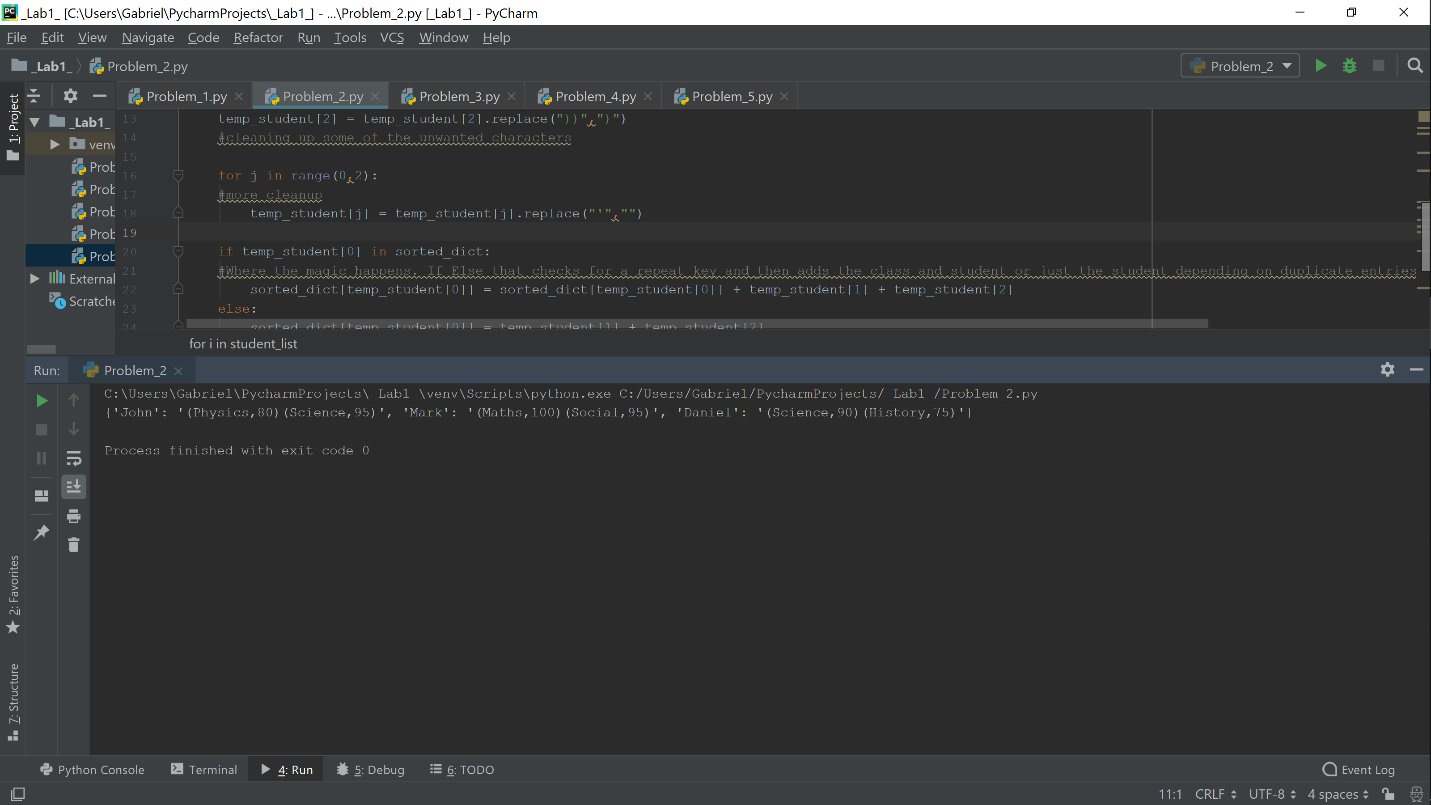
Problem 2:

For this problem I had to take a predetermined list of students and their marks in classes and sort them into a dictionary with keys based on their names and values based on their class/marks.

Here is the Code:



And the output:

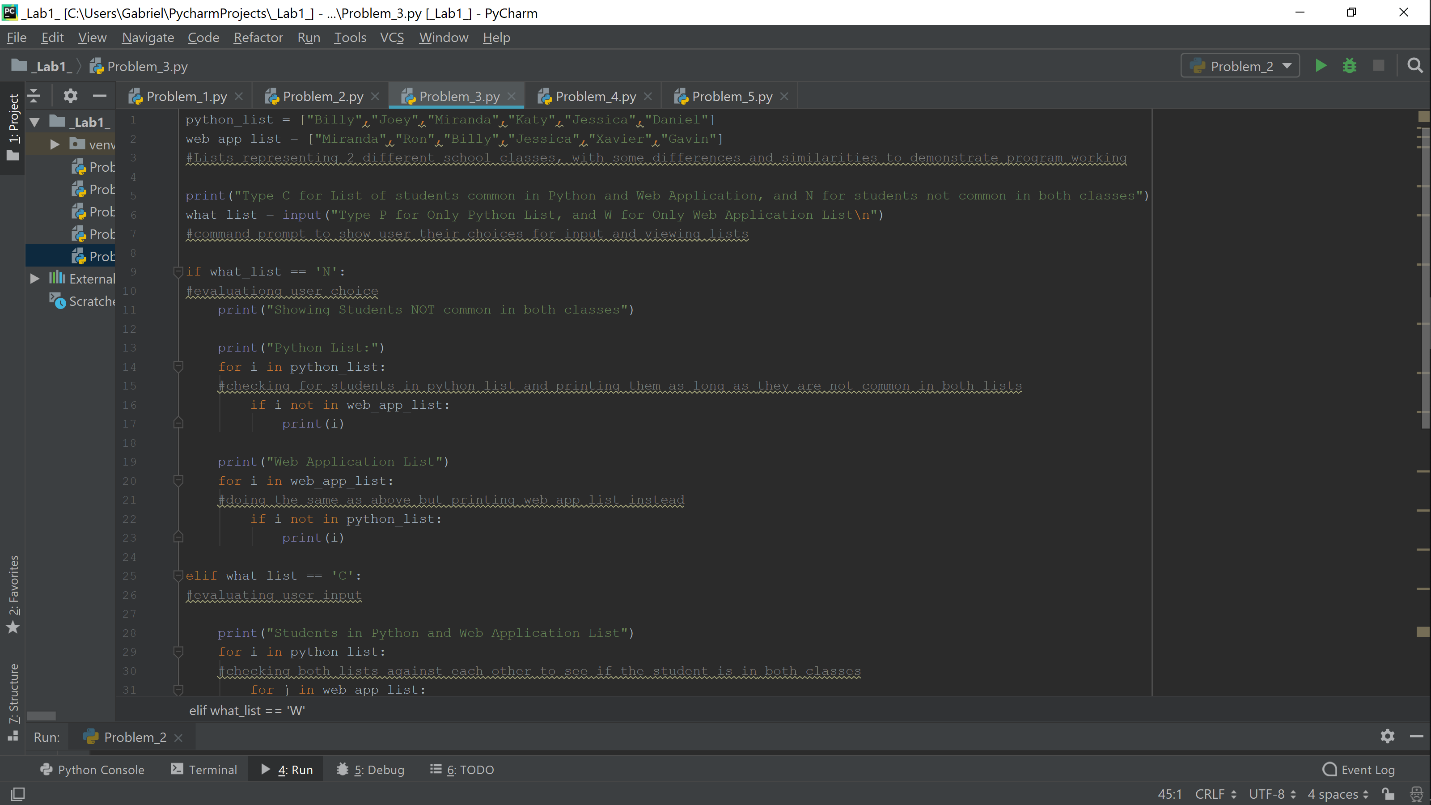


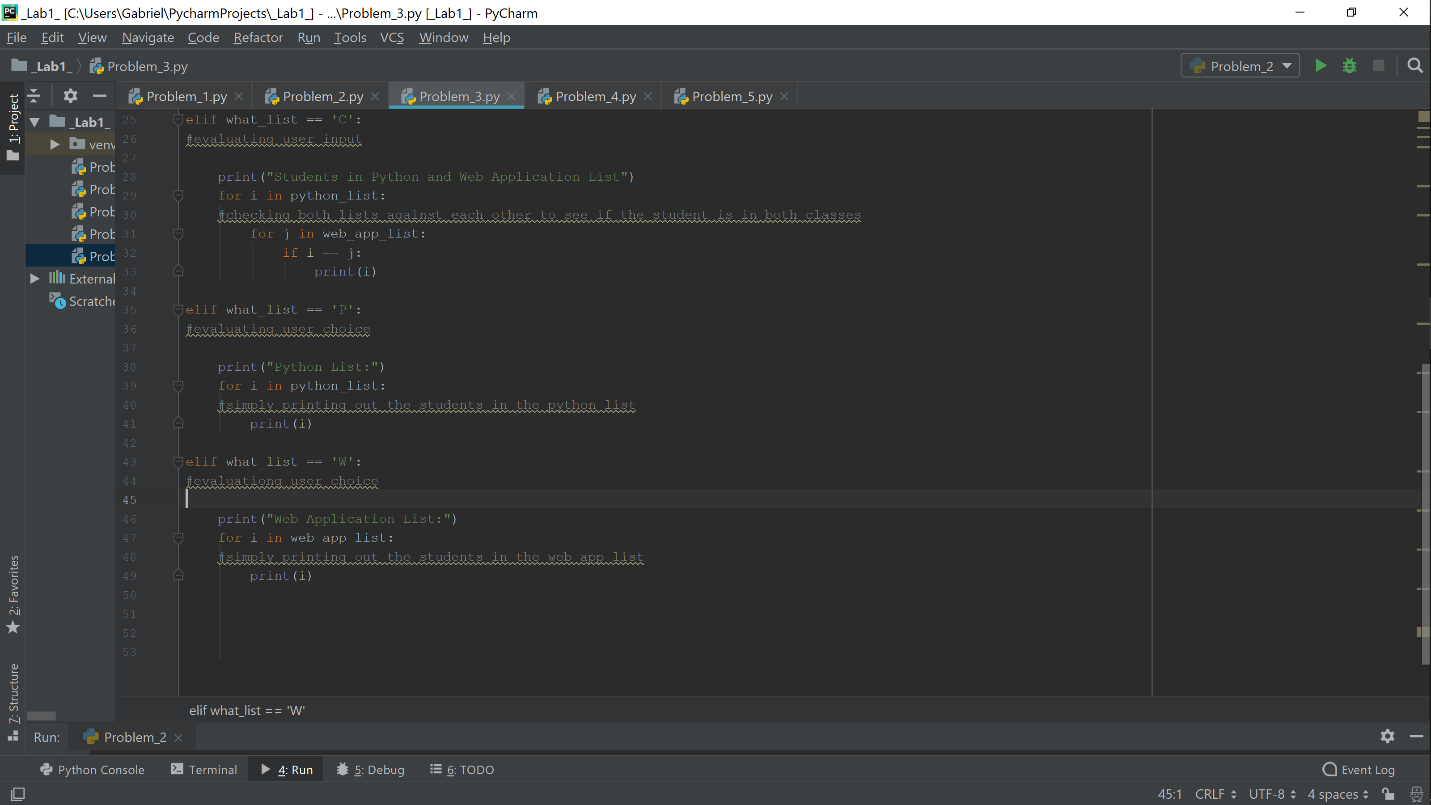
As you can see I used a nested for loop to look for repeat keys and parse the lists apart from each other and check them against each other. I then sorted their classes into their key-name to finish and print out the dictionary. Pretty straight forward and easy program overall.

Problem 3:

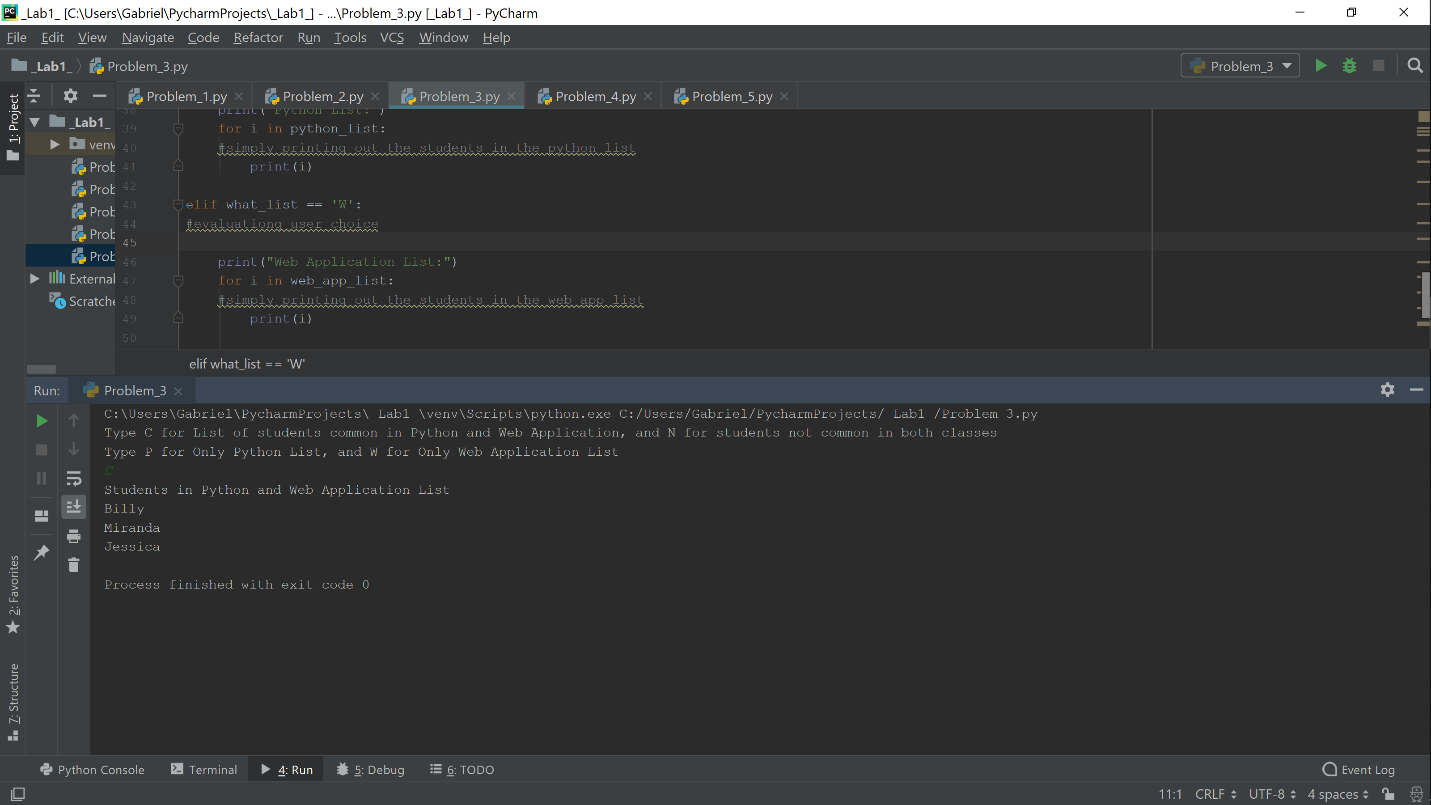
For this program I had to take 2 lists of classes that students were enrolled in and find the common students and uncommon students from those lists and print them to the console. I used predetermined, hardcoded lists to ensure that all cases were met.

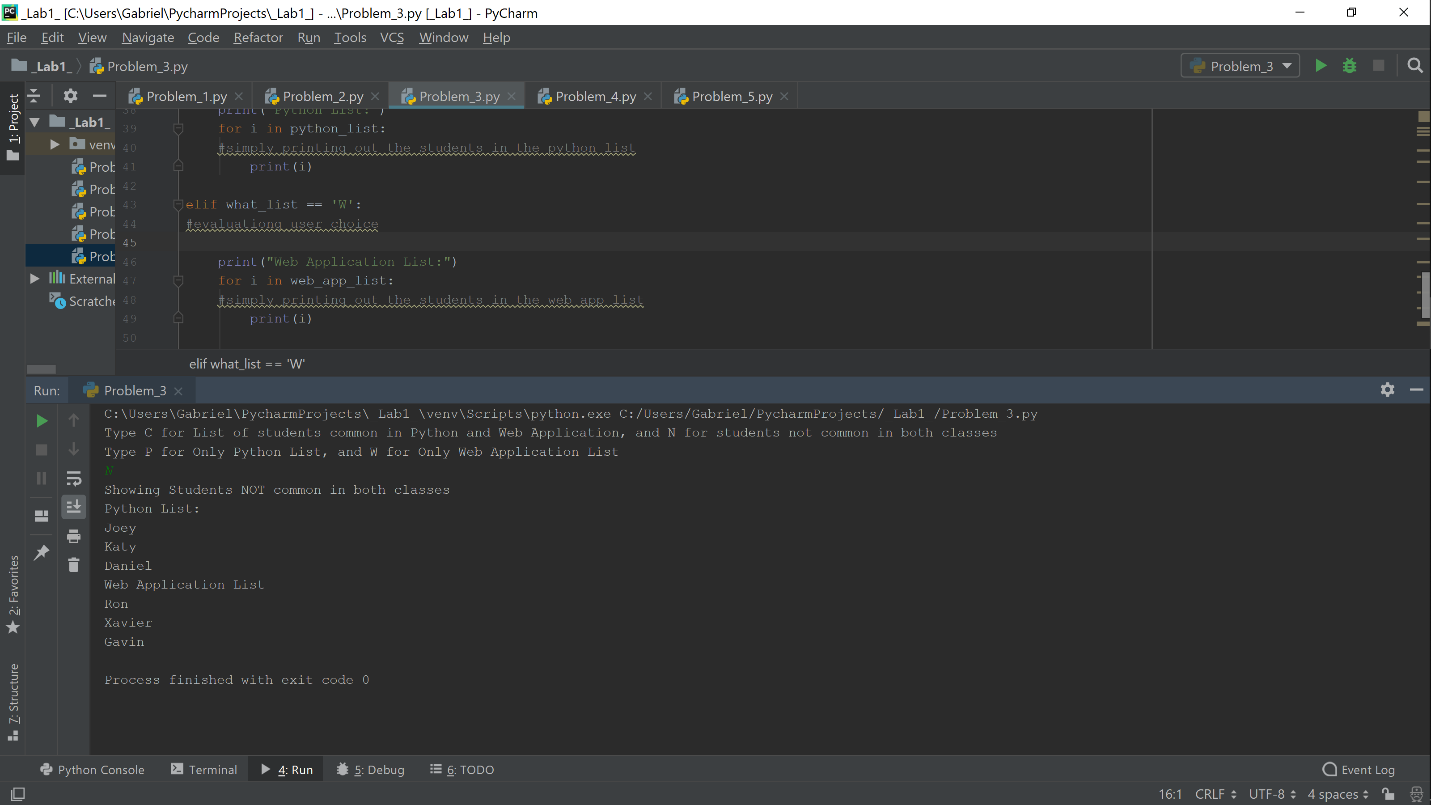
Code:





Output:



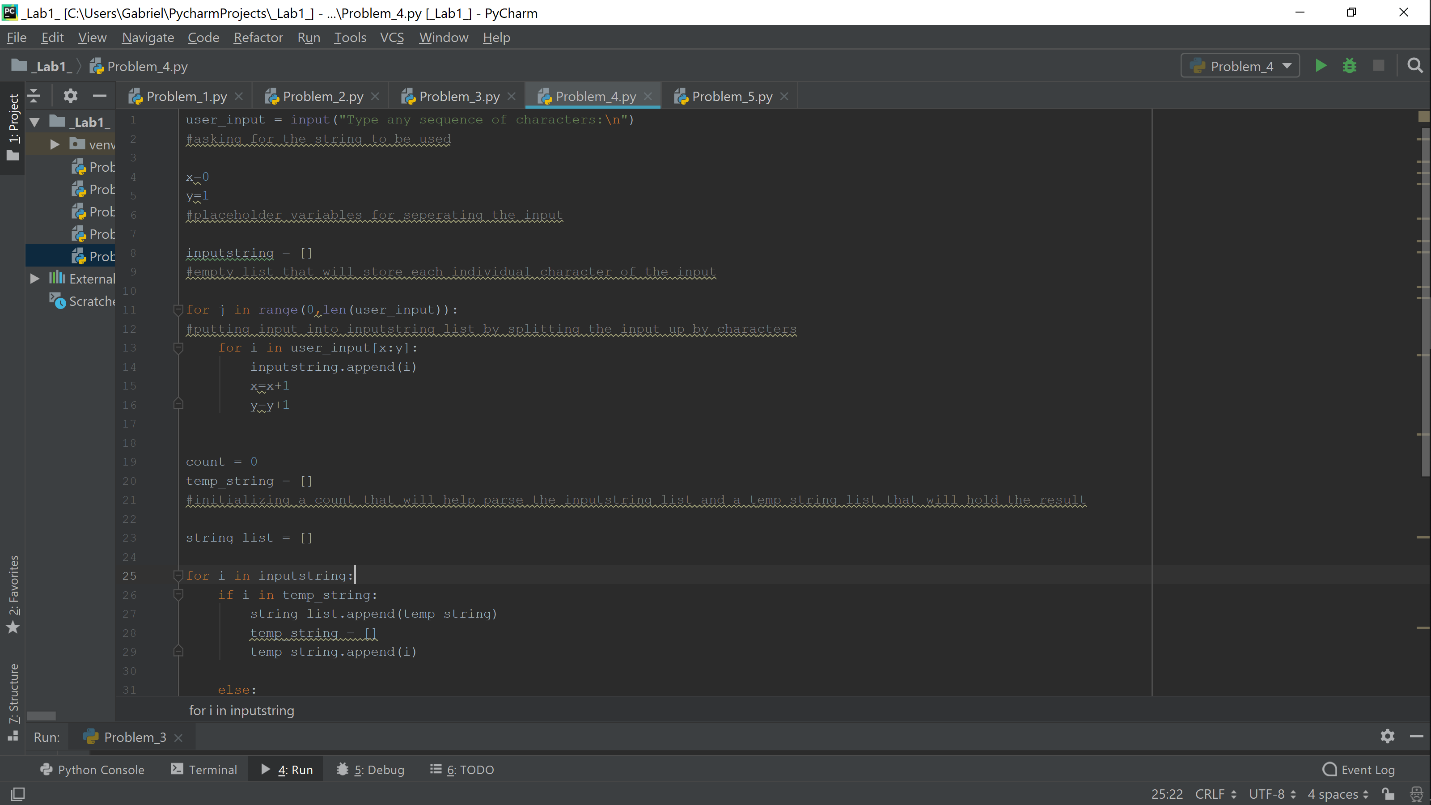


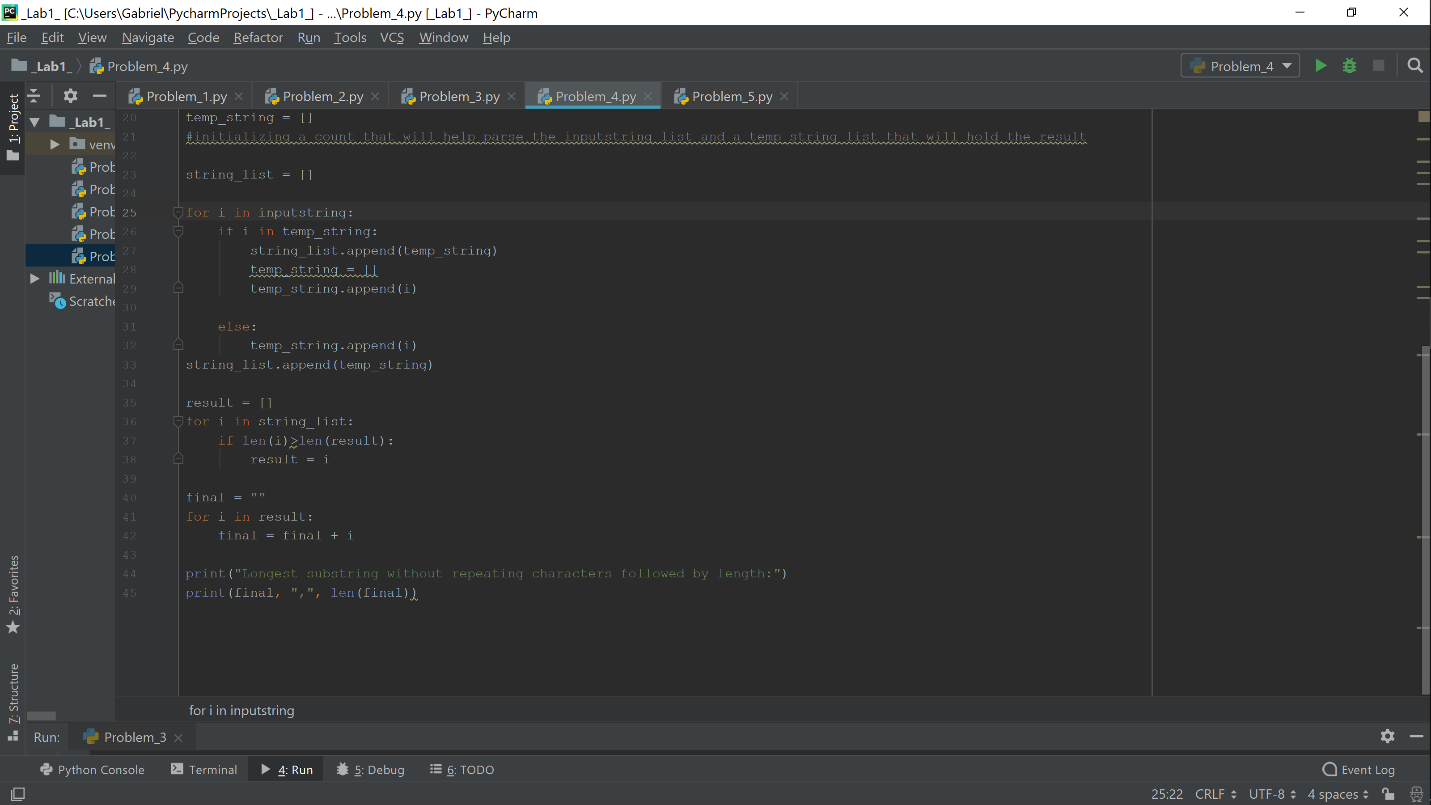
I used a couple of conditioned for loops to find students that were common in both classes and then printed the lists based on the desire of the user input. I ran into a few problems with this one but I think it turned out really well in the end.

Problem 4:

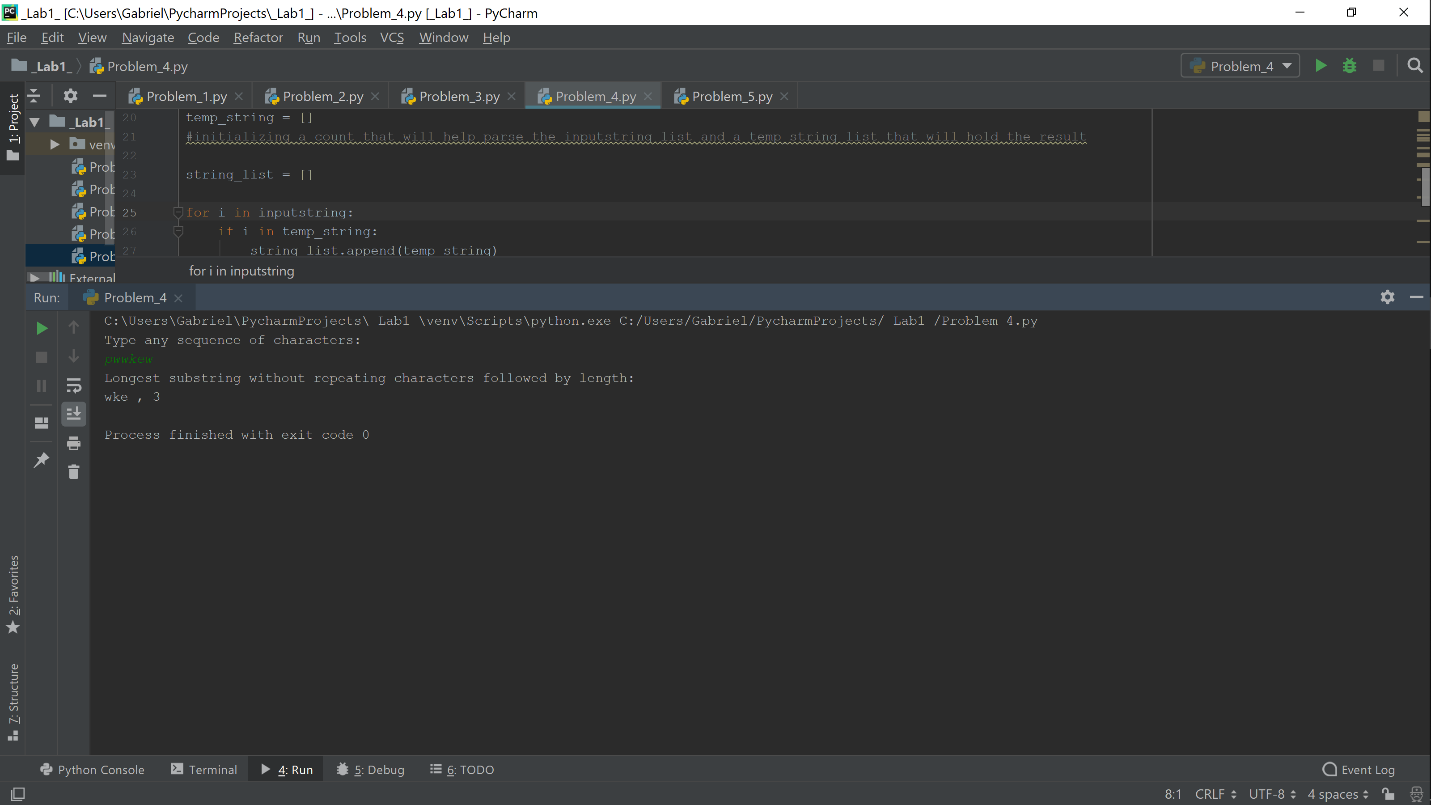
For this problem we had to give the command prompt a string and find the largest substring without repeat characters. It sounds simple but it gave me the most trouble over all.

Code:





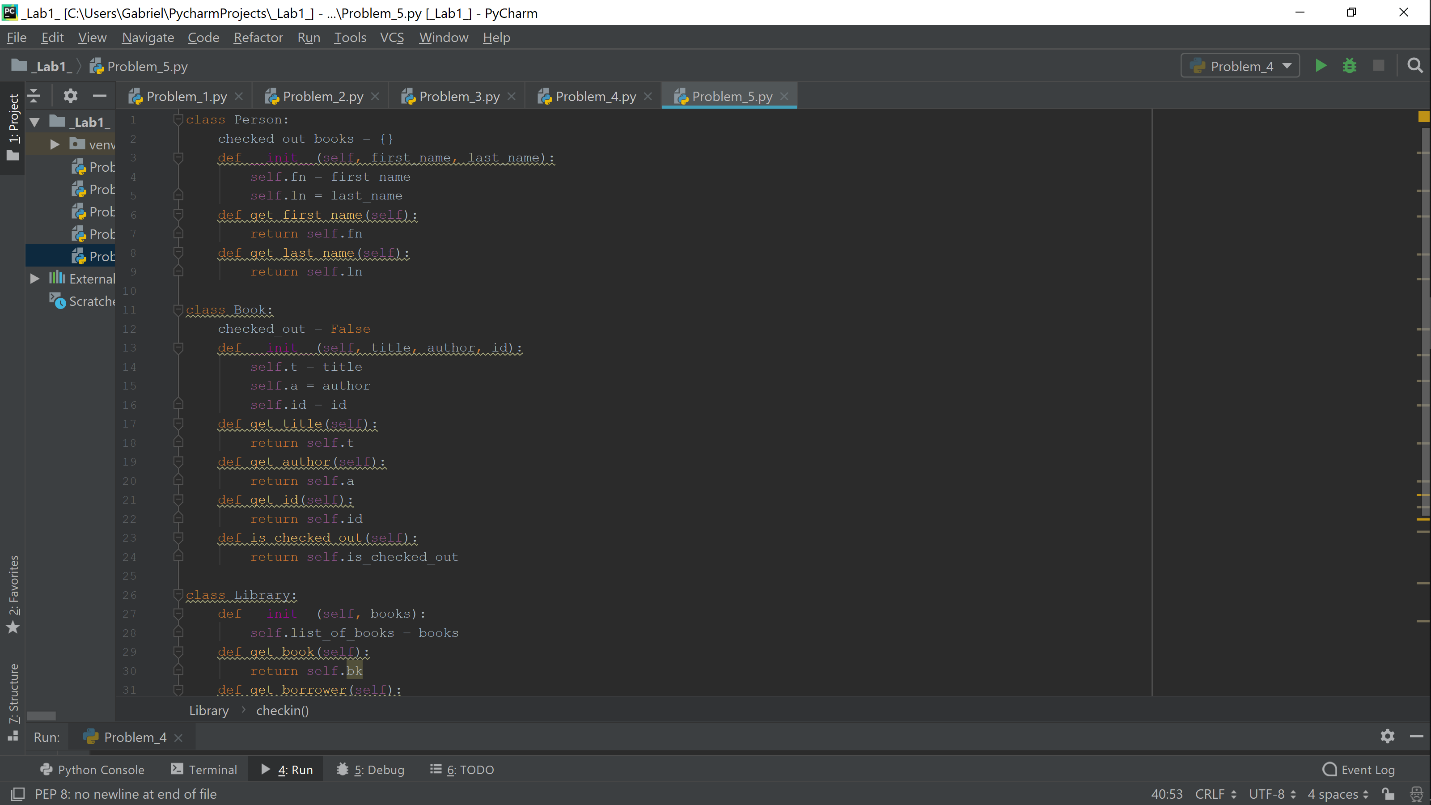
Output:

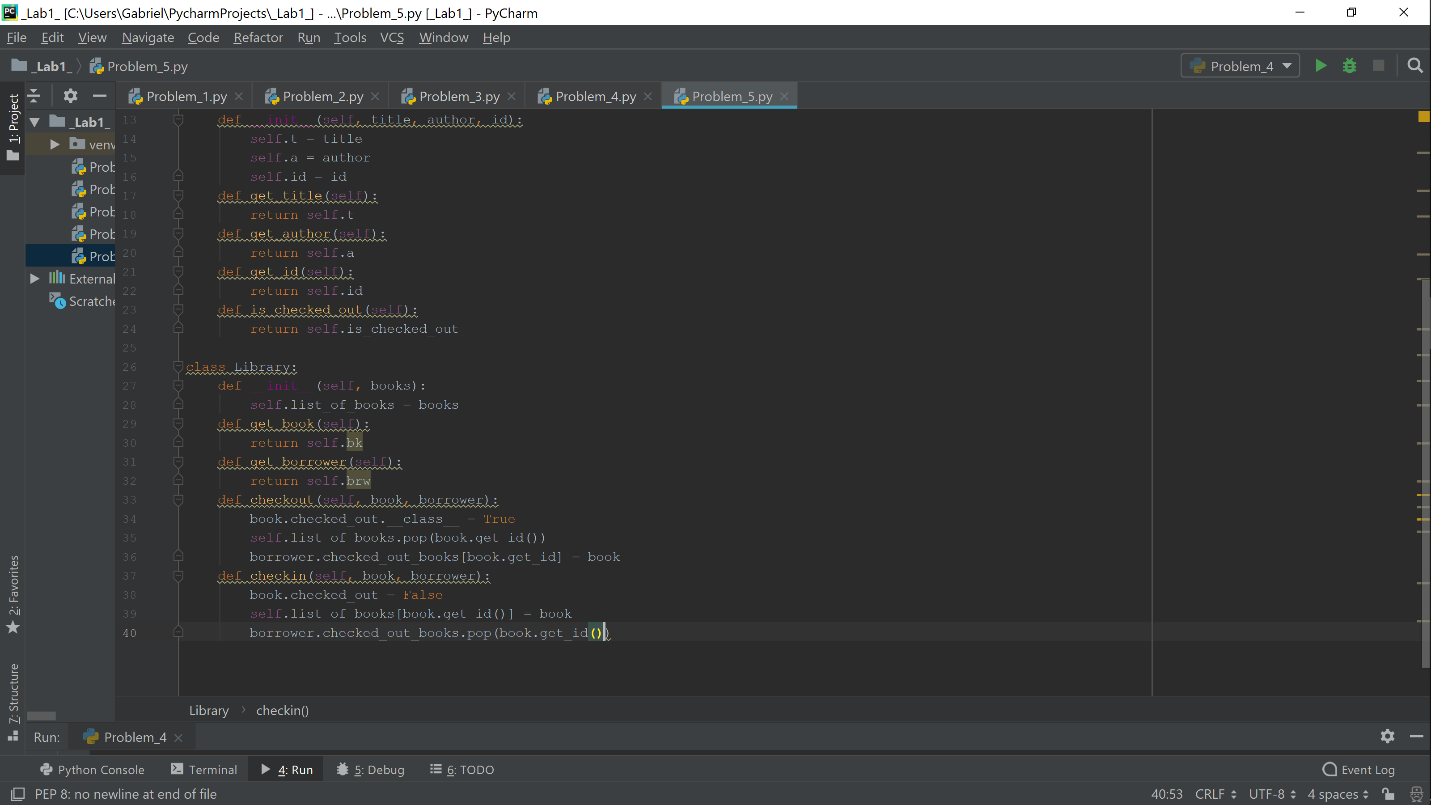


I used the sample output given in the lab sheet but it works for all cases. I was able to split the string into a list of lists that checked every character against itself and then put the substrings into a separate list. The program then compares the substrings list and finds the longest one.

Problem 5:

This one was really open ended, it was tasked with creating 5 classes for a library check out system and asked for inheritance and knowledge of object oriented programming.





Unfortunately this is where I ran out of time and I was unable to finish the entire program. However I was able to get 3 working classes and started work on 2 subclasses. I created functions that would check out a book id from a dictionary of books passed into Library class and then would check out the desired book under the given borrowers name and assign that book to the borrower.

Problem 6: Nothing to show ☹