

# CECS 277 LAB COLLECTIONS ITERATOR

**OBJECTIVE:** Learn how to use several of the Java Collections classes with each other.

**INTRODUCTION:** Please remember the coding standards [here](#).

We want to write a simple program to read through a text file and come back with a list of all of the words used in that text file, and a count of the number of times each of those words were encountered. Luckily for us, the Java Collections library gives us several valuable tools to make this happen.

**PROCEDURE:**

1. Create a Scanner object to read the file "[Pride and Prejudice.txt](#)". The easiest thing to do is to put that file in the root directory of this project. Then you do not have to supply a path when you create the File object to read in through the Scanner.
2. Create a Map object that maps from a String to an Integer. The String (the key) is the word that we are counting, and the Integer value is the number of times that we have seen that String so far.
  - a. We want to see the words come out in your final report in alphabetical order, so be sure to use the proper type of Map.
3. Read through *Pride and Prejudice* one word at a time, using the hasNext() and next() methods of your Scanner.
  - a. The default behavior will divide the words up by whitespace, which is a good approach for us.
  - b. But we want to remove anything that is not a character. For that, we will use the String replaceAll member function and a little regular expression:
    - i. `String word = in.next().toLowerCase().replaceAll("[^a-z]", "");` will run everything to lower case, and get rid of anything that is not a letter. This approach is far from perfect, but it will get the job done reasonably well.
4. Each time you read a word:
  - a. If it is not in your Map yet
    - i. Create a new entry in your Map with that word as the key, and a value of 1.
  - b. Otherwise
    - i. Retrieve the number of times that we have seen this word so far.
    - ii. Increment it by one
    - iii. Use the Map replace function to update the count for that word.
5. Then, at the end, report out your work:
  - a. Use the Map keyset() member method to get a list of all of the words that you have counts for.
  - b. Create an iterator on that set (which will give you the words in alphabetical order if you used the proper sort of Map).

## CECS 277 LAB COLLECTIONS ITERATOR

- c. Using the Iterator hasNext and Next, print out each word that you have counted, and the count (extracting that from the Map that you built).

### WHAT TO TURN IN:

- Your IteratorRunner.java file that has all of your code for this lab.
- Your console.txt file with your sample output.