Summary:

In this weeks assignment you will write a small program both in Java and C++. The key objectives this week are giving you a introduction to C++ and the STL as well as a chance to set up your C++ tool chain. You should use best style in both cases specifically using what you have learned about Java style and design in the Java case and approaching the C++ case as a different problem since good C++ is not the same as good Java. In all cases you should use the available libraries of each language to support your design.

Specification

You are to design and implement two programs to find the top 10 most frequent words in documents. One will be written in Java and use the networking that we have seen before to retrieve the file. The other will be written in C++ and designed to work as a command line tool.

In both of these assignments you should use the features of the language libraries to simplify your task. While these features are not identical they do share some common features that may be of interest. I would encourage you to specifically look at the map implementations of both languages and consider how they may be used.

Fork the starting git repository using the URL below. This time we?re not providing you any starting code, but you are provided two directories one for the Java development and the other for C++ development. You should start from the provided repository so that we can have access to your repo for grading purposes.

```
https://classroom.github.com/a/hJhD62I6
```

0.1 Java Specification

Your program should take a URL through command line arguments, such as

```
http://www.gutenberg.org/files/17921/17921-0.txt
```

You will read in the contents of the file. You should then count the frequencies of all the words in the file. You may have to process the raw file data in order to do this by stripping out any punctuation and whitespace. Keep in mind that the ’ and - characters may be found in words such as "Heisenberg’s" and "eel-esque". The file you are given will only have ASCII characters.

After finding the frequencies of the words, print out up to the 10 most common words in the file you read. Remember to use good style, variable naming, and decomposition. Planning your solution out before coding it and using the appropriate Java data types will also be helpful in a good solution of this assignment.

0.2 C++ Specification

You will implement the same word frequency program in C++, however instead of reading the file from a url you will read it in from stdin and input the file using pipes which we talked about briefly at the start of lecture on Tuesday. You will use |, the pipe operator to pass in files from the command line. The pipe operator allows you to redirect the output of one process to the input of another. For example we can chain grep which can be used to search for a string and cat which prints the contents of a file as:

```
cat README.md | grep 'CS126'
```

Which searches for CS126 in the README.md file. Pipes are a powerful tool that you can read more about here:

http://www.linfo.org/pipes.html

You should also read about the unique libraries provided to you by C++ here in order to figure out the ideal solution.

http://en.cppreference.com/w/

Simply porting over your Java program to C++ will NOT yield an appropriate solution.

Testing

You are expected to properly test the interface of your programs in both languages. In the case of the Java implementation you should use Junit4 as you have been all semester. In the case of the C++ implementation we are using Catch as a testing framework.

Catch is a widely used testing framework for C++ and can be integrated into CLion which is what we are recommending as an IDE for C++. Documentation on Catch and how to use it can be found at the following:

https://github.com/catchorg/Catch2/blob/master/docs/tutorial.md#top

While we will be happy to help you resolve issues with getting started we strongly encourage you to read documentation and try solutions you can find before posting to Piazza. One of the key goals in this weeks assignment is for you to find how to work with your development environment and a core part of that is understanding your testing framework.

Design and Style

Much like we have been using the Google Java Style Guide for the Java code we have been writing we will be using the Google C++ Style Guide for C++. The style guide can be found at the following.

https://google.github.io/styleguide/cppguide.html

In this assignment we will be asking you to at least comply with the naming rules which can be found here.

`https://google.github.io/styleguide/cppguide.html#Naming`