

Santa Monica College
CS 20B

Programming Project 6 - Due: **Thursday, December 17 at 3:00 PM**

You enjoyed exploring the common words last time and you've decided to explore some more. You plan to write a program that processes a large text file and given a word in the file will return a list of words that follow that word, sorted from most likely to least likely to follow. You will then use this capability to make up random “sentences” that are likely correct grammatically but almost completely meaningless, such as the literary gem below:

their shoulders immodesty unchained a child in our souls were in france who do whatever

The main skeleton of the code has been given to you in Project6.java. Please fill out the given methods and add any other data members or private methods you may need.

Hints:

- We plan to use the Java library HashMap class for this project. Have a look at the documentation here:

<http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html>

- You will very likely NOT need to override hashCode or equals in this project
- Map key-value pairs are stored in Entry objects. You may likely find this documentation helpful as you work with your maps:

<http://docs.oracle.com/javase/7/docs/api/java/util/Map.Entry.html>

- Think about what the type should be for the key and for the value to be able to store the information you get from the file (hint: the value is itself a mapping).

- After you have populated the map with information from the file, you will likely need another data structure, that organizes the information in the first data structure in such a way that it is ready for the getTopNFollowingWords method to use.

Deliverables:

You should submit a zip file named project6_first_last.zip (where first and last are your first and last name) containing ONLY the 2 files below.

Project6.java

report.txt - a text (not Word, etc.) file containing:

1) a 1 to 10 lines paragraph from you saying “I have tested this program and there are no known issues.” if you believe that to be the case, or a brief description of known issues in case your program has known problems or you could not fully implement it.

How you get points:

- | | |
|---|-----------|
| - Constructing a correct and efficient data structure | 60 points |
| - getTopNFollowingWords | 15 points |
| - makeRandomSentence | 25 points |

How you lose points:

- You do not follow the given directions and decide to make changes “for fun”. Specifically, **do not change the method signatures given to you.**
- Your implementation is inefficient. Your solutions should be **efficient to a level seen in class for similar problems.**
- You submit your whole workspace. **Submit only the files the project asks for.**
- If any of your code prints anything at all on the console except for the main function. **Remove all your print outs, debug statements, etc. Clean up your code and do not leave clutter behind.**
- Your code has no comments where needed. **Comment your code appropriately.**