**Homework #8, Due Monday December 18, Grade: 5%   
Last Modified: *12/08/2017 22:23:10***

**Problem #1 Java 8 Streams and Predicates**

Complete the code for **BeerFest.java** in this zipfile [BeerFest.zip](https://courses.dce.harvard.edu/~cscie55/BeerFest.zip) It contains unimplemented code [see **ToDo** comments. For examples of lambda expressions visit the Java files listed here: [Video Rentals](https://courses.dce.harvard.edu/~cscie55/src/rentals/lambdas.htm) There are lambda expressions sprinkled around those classes.].

**Problem 2 Anagrams (extra credit)**. Anagrams are words or sequences of words that use exactly the same letters but in a different order. For instance the words "now" and "won" are anagrams of each other.

Here are some more complicated anagrams:

* "Public relations" -> "Crap built on lies"
* "Wolfgang Amadeus Mozart" -> "A famous German waltz god"
* "Eleven plus Two" -> "Twelve plus One"

Your task in this problem is to identify anagrams in a list of words. Considerations:

* Consider each line in the file as a record, i.e. a word or sequence of words that is an anagram candidate.
* Ignore white spaces and punctuation in testing for anagrams
* Ignore cases, upper vs. lower.
* To look for anagrams you need to extract a core representation of the constituent characters of each record from the data file, call it the "character key string.". For example, for the line that reads "Wolfgang Amadeus Mozart" you should generate a pair of strings, the first being the character key string, "aaaadefgglmmnoorstuwz", and the second be the word or phrase on the line. Likewise, when your program eats the line "A famous German waltz god", it should generate the same key string, allowing it to identify the two strings as anagrams of one another. Notice that the letters are in alphabetic order and that each letter appears as many times in the key string as it occurs in the line. Thus, your stream processing should generate a pair of strings,
* Use Java 8 streams to
  + read the input lines
  + eliminate duplicate lines, and
  + generate Key/Value pairs combining the character key string and the line
  + collect the results into a map where the key is the character key string and the value is the string concatenation of the anagrams. Output should have the form:

**eeehnoorttw->one two three...three two one**

* + Run your solution on the data file in this zipfile: [anagram-data.zip](https://courses.dce.harvard.edu/~cscie55/anagram-data.zip) and submit the last 20 lines of the output together with your Java code.