Assignment 7: Reading and Programming Project due 10/24

Reading Assignment

Read the latest course announcements ([http://cis.bentley.edu/tbabaian/cs230/announcements/postmid.html](http://cis.bentley.edu/tbabaian/cs230/announcements/postmid.html)).

Review and make sure that you understand how the the Histogram.java program from class works ([http://cis.bentley.edu/tbabaian/cs230/examples/Histogram.java](http://cis.bentley.edu/tbabaian/cs230/examples/Histogram.java)).

Examine the solutions to the midterm carefully. Review the course material as necessary.

Programming Assignment

This week we revisit the **RepeatTextStats** project that you worked on for Assignment 5. The description of the assignment is repeated in this document. The project is worth 8 points with 2 points for the extra credit part.

In class we went over two strategies for identifying the length of words on a string.

1. Repeat for each line until the end of line is reached: find two consecutive separators (a separator is a comma, a period or a space character) and compute the length of a word between the two consecutive separators.

2. Repeat for each line: traverse the line starting from position 0; check the character at each position and increment the worldLength counter if that character is a letter (i.e. not a separator); when a separator is encountered - count the word that was just parsed and reset worldLength to 0.

There will be one grade for this project on your record and it will be the maximum of the grades for assignment 5 and assignment 7. For the purpose of the final grade, the TextStats project is worth 8 points.

If you have already received a high enough grade for your past submission, I suggest you work on improving the clarity of your code and its structure, with the goal of producing the decomposition into methods that evenly partitions the functionality of the program.

Programming Project

**RepeatTextStats: Compute Text Statistics**

due 11:00 p.m. on Thursday, 10/24

worth 8 points, 2 points extra credit

In this assignment you will be processing strings of text. The user will be inputting text line by line, until they enter the word STOP. Your program must compute how many words of length 1 were entered, how many words of length 2 were entered, and so on until (and including) 10-character words, and finally, the number of words of length 11 or higher. The input will consist of letters, white space characters ' ', and punctuation characters '.', ',', and ':'. Notice that a punctuation character is not considered to be part of a word even if it immediately precedes or follows a word.
For extra credit, also compute how many of each of 26 characters of English alphabet were used in the text.

When the user indicates the end of the text by inputing the word STOP on a blank line, your program must print out the text statistics in the form shown in the following sample interaction, and terminate.

This program computes input text statistics.
Enter text with no other punctuation symbols but . and , Enter the word STOP on a blank line to stop.
This is just a simple test.
zzzzzzzzzzzzzzzzzzzzzz
STOP
Printing Word-Length Statistics
1:1
2:1
4:3
6:1
11:1
Printing Character Statistics
A:1
E:2
H:1
I:3
J:1
L:1
M:1
P:1
S:5
T:4
U:1
Z:25

Notice that the program should not print statistic values that are 0. Note again that the character statistics part is for extra credit only and you should work on it after you have completed and tested the program without it. (You must do it without using a giant conditional over 26 cases to get extra credit.)

You must use a static method in your program. I suggest that you create a method that takes in a one-line string and the array(s) that contain the word-length (and maybe character) counts and updates these arrays according to the content of the string.

Note also that partial credit (up to 2 points) will be given to programs that correctly compute the text that contains only one word on each line, e.g.

This program computes input text statistics.
Enter text with no other punctuation symbols but . and , line by line. Enter the word STOP on a blank line to stop.
one
two
three
four
STOP

Printing Word-Length Statistics
3: 2
4: 1
5: 1

Printing Character Statistics
E: 3
F: 1
H: 1
N: 1
O: 3
R: 2
T: 2
U: 1
W: 1