THE UNIVERSITY of NEW MEXICO

Lab 4: Sort Three Cities

Overview:

Listing 4.2: OrderTwoCities.java is a program that inputs the names of two cities and displays them in alphabetical order.

Your assignment is similar. Your program must input the names of THEEE cities and display them in alphabetical order. ALSO, your program must output, for each city, if that city is a one-word name (Like "Albuquerque") or a two word name (like "New York").

Grading Rubric [20 points total]:

[File Name: 1 point]: Attached one file in Blackboard Learn with the file name: **`SortThreeCities_`***yourName*`.java`.

[City Input: 2 points]: Your program must prompt the user to enter the names of three cities. Assume the user inputs city names that are correctly spelled and that start with a capital letter (assume it, means you do NOT need to check for this). Assume that each city name your program will not be tested with city names that are less than one word (i.e. blank) or more than two words.

[Alphabetical order Input: 12 points]: Your program must display each city name on a different line in alphabetical order. If a city name is repeated, then, print the name twice (and, of course, it will not matter which you print first). Java's string1.compareTo(string2) method considers the space character to come lexicographically before any letter. For example,

```
"New York".compareTo("Newark")
```

will return a value less than zero because the first three letters of both cities are the same while the 4th letter of "New York" is a space and the 4th letter of "Newark" is an 'a'. Therefore, Java considers "New York" is lexicographically before "Newark" and your program should print the two cities in that order.

[Words in Name: 5 points]: After displaying a city name, on the same line, print a space and display the number of words in the name as 1 or 2. That is, a city name is one word if it contains no spaces between non-space characters.

Penalties:

[-5 points]: Code does not adhere to those parts of the hallowed CS-152 coding standard thus far covered:

1) Correct indenting (no tabs and two spaces per block level).
2) Correct placement of brackets.
3) A comment at the top of the class giving your full name and the date.
4) In-line comments as needed. "As needed".
5) Must compile without warnings with IntelliJ's default warning settings.

Note: all 5 points are lost if any ***one*** of the standards is severely broken.