## Strings: equality, and case-insensitive comparisons

## The goal for this exercise is to learn how to compare strings (to compare textual information) in C#.

 The really nice thing about the **string** objects is that they can be compared using the equality and inequality operators (== and !=, respectively). They cannot be compared using <, >, <=, **>=.**

 You should notice two things: that two strings are == only if they are identical (i.e., they have the same letters, AND the same case), and that unlike Java, they're compared based on the text, not based on which object each variable refers to (if this last part doesn't make sense (maybe because you haven't used Java), then ignore it ☺ ).

 If you want to compare two strings without case-sensitivity, you should use code like this (where **input**, and **input2** are string variables, and the word true tells the String.Compare method to do the comparison IGNORING any case differences)

if(0 == String.Compare(input, input2, true))

{

 Console.WriteLine( input + " and " + input2 + " are the same, ignoring case!" );

}

else

{

 Console.WriteLine( input + " and " + input2 + " are NOT the same, ignoring case!" );

}

 **What you need to do for this exercise:**

1. Within the starter project that has been provided to you, add the following code at the end of the file (but BEFORE that final, closing } – this code should go INSIDE the namespace, not outside it.
NOTE: I highly recommend that you verify that this compiles now before going on to the next step.
 **class String\_Basics
 {
 public void RunExercise()
 {
 }
 }**
2. Within the RunExercise method, ask the user to type in two separate strings, and then tell the user that they've typed the same thing in twice **if** the user's two inputs are ==. Tell the user they've typed in something different, if the strings aren't the same.

	1. While you're just using this exercise to familiarize yourself with basic C# string comparison, you can use this anywhere you can to compare two strings. For example (you are not required to do this) you might ask the user if they want to continue, and then compare their answer to the word "yes". You would probably also want to compare their answer to the string "y", just in case they abbreviate their answer, but at this point you should have the general idea on how to use this.
3. Next, tell the user if the first message is the same as the second message, if the first message comes before the second message, or if the first message would go after the second message, using the **String.Compare** method in it's "case-sensitive" mode.

	1. You will probably need to search the web for information about this method. Using your search engine of choice try searching for stuff like “C# string.compare” or “C# string compare tutorial” until you find something that helps you understand this. Adding “C#” to your search string will help the engine narrow down your results to just the C# language, and “tutorial” will ask for stuff that’s intended to explain stuff to beginners. You may need to try several different searches (and several results from each search) before you find something that helps you, personally, out.
	2. Make sure to try things like "A" vs. "a", and numbers/symbols vs. letters, etc. **Put in a comment (at the top of the StringBasics class) explaining how words will be sorted when the case-sensitive version of String.Compare is being used.**
4. Finally, do the same thing as the prior part, but this time use the String.Compare method in it's case **IN**sensitive mode.
	1. Make sure to try things like "A" vs. "a", and numbers/symbols vs. letters, etc. **Put in a comment (at the top of the StringBasics class) explaining how words will be sorted when the case-INsensitive version of String.Compare is being used.** I.e., what does it mean for one word to be "less than" another word, according to String.Compare?
5. For this exercise, this means that you will ask the user for two separate strings to compare, then compare them THREE times – first using the == operator, a second time using the String.Compare method (case-sensitive), and a third time using the String.Compare method (case-**IN**sensitive).