Scalable Data Infrastructures - Code Exercise 07

Overview

This code exercise will give you a chance to work with strings as objects and manipulate a string using string methods.

Instructions

It's always best to start with a problem analysis. While one is not required to be submitted for this code exercise, it would benefit you greatly to begin by examining the requirements and creating a bulleted list of the things you'll need to complete the code. You can then begin by creating a new solution with the following naming format: *LastName_FirstName_CE07*.

Create a string that is a comma-separated list of items. An example might be the following:

> "Ford, Chevy, Subaru, Hyundai, Kia, Honda, Toyota"

You should come up with your own list of items and not use the one above. You will lose points if the above comma-separated list is part of your project.

Use a string method to create an array from the comma-separated string. Then use another string method to trim any spaces from the elements within the array. This will require the use of the proper loop to cycle through each array element and apply the appropriate method. Use another string method to make all the array elements uppercase strings.

Finally, output the array elements one at a time to the console. As always, make sure you're including additional text to give context to the user in regards to the list of items.

Remember that many string methods do not change the string. That is, the method *returns* a manipulated string rather than changing the string itself.

Things to Consider

Look to your problem analysis to verify that you've included all the requirements of the problem. If you ask your instructor or a lab specialist for assistance on the code, he/she is going to ask to see your problem analysis first. If you didn't do that, we are going to tell you to complete that task before we will look at your code.

Make sure you're including comments at the top of the code to include your name, class and term, and the assignment name and that you have meaningful comments for each line of code in the project.

Finally, remember that you must compress the entire project folder for submission. Submitting only the Program.cs file or the .sln file will result in a 0 for the activity.

Rubric: Code Exercise 7

Minimum Project Requirements

These requirements must be satisfied before any points are awarded. Failing to meet these requirements will result in a zero (0) grade.

1. Project must run when instructor compiles it.
2. The submission must be submitted in the proper format as defined in the FSO activity.
3. You will lose 5 points if the project does not follow the naming convention described in the activity's documentation.

Topic	%	Excellent (100%)	Acceptable (80%)	Good (50%)	Fair (25%)	Poor (0%)
Coding						
Comments	**5**	Comments exist at the top of the code to include name, class and term, assignment, and date, and each line of the code is properly commented.	Missing the initial comments with name, class and term, assignment, and date, but the rest of the code is commented properly.	Missing up to four line comments, but some comments present.	Missing more than four comments	No comments in the code.
Syntax	**15**	There are no syntax errors, including correct line and **formatting according to the style taught**.	There are no syntax errors, but the code does not follow the style taught.	Project code contains minor syntax errors but is easily fixed.	Project code contains more major syntax errors but are easily fixed.	Project code does not run.
String	**20**	A string of comma-separated items exists within the project.	A string of items exists, but a separator different form a comma is used.	A string of comma-separated items exists, but it is the same list of items presented in the assignment rather than an original list of items.	The comma-separated list is not hard-coded as part of the project, but is gathered by user input	No string of items exists in the project.
String Methods	**25**	The correct string methods are used to create a string array, to remove extraneous spaces, and to uppercase all the letters of each array element.	Missing one of the required string methods.	Missing two of the required string methods	The strings are manipulated using incorrect or inappropriate string methods.	No use of string methods.
Loops	**20**	A correct loop is used to remove extraneous spaces from array elements and to output the array elements to the console.	Up to one use of the incorrect loop for cycling through an array.	Both loops use the wrong type of loop to cycle through the array.		No loops used within the project.
Output	**15**	All required outputs exist and contain additional text to give context to the user.	All required outputs exist, but some additional text is missing, which leaves out context for some outputs.	All required outputs are present, but there is not additional text to give the output context.	Missing one or more required outputs.	Nothing is output to the console.