

Scalable Data Infrastructures - Code Exercise 04

Overview

Loops allow us to run a single block of code multiple times. This can be useful in many situations. Whether it's validating user input or counting the number of items in an array, being able to run the same block of code gives us one more way to control the flow of our code. In this code exercise, you'll work with a for loop to output the number of times a user performs a task.

Instructions

As with any programming task, you should start by reading through the problem and creating a problem analysis. This will give you a starting point for beginning your code. Then create a new solution in Visual Studio using the following naming convention: LastName_FirstName_CE04.

The first thing you'll do is ask the user to enter an item that they need to wash, such as a glass or a towel. Then, you'll ask the user to enter how many of those items they have to wash. Thus, if the user enters cars and 10, it means there are 10 cars to wash.

Next, create a loop that will run for as many times as it needs to in order to wash each item. That is, the loop should count up from 1 to the number entered by the user. If the user entered car and 10, then each time the loop runs, the output should indicate that that number of cars has been washed, adding one to the output each time. As always, the output should give the user meaningful output and not just the output of the item and the number. It would be the difference between the following output:

1 car

and this output:

1 car(s) has been washed.

Things to Consider

We've learned about two kinds of loops. One runs while something is true and the other runs a specific number of times. Since this exercise specifically indicates it must run a certain number of times, you should use that loop. You will lose points if you use the wrong loop for a count.

Keep in mind that a ReadLine always returns a string data type. You will need to convert that user input to a proper data type in order to use it in your loop (there's no need to validate that the user entered a number; for this exercise, we'll assume the user enters the correct data).

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 4

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.

[illegible]