Scalable Data Infrastructures - Project 02

Overview

This project will give you an opportunity to bring together many of the concepts you've learned so far. It will incorporate user input, input validation, loops, and arrays to create an application that asks the user for input that you will then store in an array.

Instructions

This project consists of many different parts. It would be to your advantage to carefully read through the instructions first, then create a problem analysis based on the information presented. Not doing so will be detrimental to your success and require you to spend more time than necessary on this project.

Create a new solution in Visual Studio using the following naming convention: *LastName_FirstName_Project02*.

The first thing you will do is ask the user to enter the number of items they wish to store in your application. It can be anything you like such as books, movies, toys, or games. It's up to you to decide as you will describe the item in your WriteLine when asking the user to enter the number of items. You will use a loop to validate that the user entered something; if they leave it blank, you should ask them to try again until they enter a proper whole number.

Once the number has been validated and converted, you will create an array of that size to hold that number of strings. Your array name should be something appropriate to the data that it is going to hold (books, movies, games, or the like).

Using the length of the new array, create a loop that will contain a code block that asks the user to enter the title or name of each item. Within this loop, you will need a validation loop to ensure the user enters a proper string and does not leave it blank. Once validated, that item will be assigned to the appropriate array index. Thus, each time the loop runs, it will ask the user for string data, validate that data, and then assign it to the current array index. When complete, each array index should hold one title or name of an item.

Finally, create a second loop that will output each item in the array in a meaningful way. That is, output each array element in such a way that the user will know what the list of strings means. It is left up to you as to how you want to do that, but it must make sense to the user.

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: Project 2

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	**10**	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	**10**	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.
User Input	**20**	ReadLine is set up correctly with a corresponding WriteLine that contains descriptive text to indicate what the user must do , and all user input is properly validated.	ReadLine is set up correctly, but the text of the WriteLine is not descriptive.	ReadLine is set up correctly, but there is no WriteLine to indicate what the user must input.		ReadLine is not used or is not set up properly.
Validation	**20**	All user inputs are properly validated with loops and, if required, are converted/parsed to the proper data types.	One user input is not validated or user input is converted/parsed to the wrong data type.	User input is validated with a conditional, not a loop.	More than one user input is not validated.	No user inputs are validated.
Data Gathering Loop	**20**	An array is created and its property is used to create a loop for getting additional input from the user.	An array is created, and a loop is used to gather input, but the array property is not used to determine how many times the loop will run.	An array is created and data is stored, but the wrong loop is used to gather user input.		No loop is used to store data or an array is not created.
Data Output Loop	**20**	A second loop is created to output the elements of the array and the output contains enough descriptive text to indicate to the user what the list of data means.	A second loop is created and outputs the array data, but the array property is not used to determine how many times the loop will run.	An array is created and data is output, but the wrong loop is used.		No loop is used to output the data to the console.