**Custom List Planning Phase**

|  |
| --- |
| **For this project, you will be exploring Test Driven Development (TDD). Writing unit tests is required in many work situations and may included as part of the interview process. Writing unit tests for projects is considered best practice and will prove to be worth the time, especially when you are working with large applications. To begin this process, consider this: what do I expect my custom list to be able to do with a set of data?** |

Research

Build a list in C# and step through the following methods (drill down on the lists to see what is being stored at each index in the array). Take notes on what you observe.

List.Add();

* When you instantiate the list object, it creates an empty array
* When you add the first one, it (I assume) creates a new array with 4 elements
* When you add the fifth item, I assume it creates a new one with 8 elements
* ?what happens when you add 4 more – does it go to 12, or 16? – it does go to 16 (0-15), as Mike demonstrated
* When I have added 9 items, so there are theoretically 16 (0-15) spots, AND I TRY TO SET THE 15TH ITEM ( realListStrings[15] = "abcd";) I GET “**System.ArgumentOutOfRangeException”**
* Returns void
* Need to be able to do this also: List<int> numbers = new List<int>() {1, 2, 3, }

List.Count; - returns the behind-the-scenes number of used spots in the array

List.Remove; -

* searches for the first item the search is equal to and
* removes it from the array;
* moves the next one to the prior index,
* etc. until you have moved the last one in the array
* set the Count to the last one +1 in the array (because 0-based)
* sets the newly-unused spot in the array to null, IF IT’S A STRING
* returns Boolean if one was removed

\*Note: Add at least 5 items. As you add them, look at the count of the underlying array.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TDD

For each method described below, briefly list 5 tests you could run.

|  |
| --- |
| *Ex:*  **Add Method**:   * Check if count increases * Check for value entered at specific index * Check if string value is correctly added * Check if each added item appears at the last index of array * Check if previous value at a given index shifted when new value was added |

**Remove Method:**

**Count Property:**

**Iterator Method:**

**Addition Method:**

**Subtraction Method:**

**Zipper Method:**

* **As a developer, I want to implement the IEnumerable interface to accomplish the Zipper method.**

\*\* Note: You may need other methods in your project in addition to the ones outlined here. Include unit tests for any additional methods you create, including the extra credit.

|  |
| --- |
| Save file as: LastName\_CustomListPlanning then Slack this document and UML to all instructors. You may start on your unit tests once you have submitted. Please have UNIT TESTS CHECKED OFF by an instructor before beginning to write the code for the custom list. |