# Review: Using the .Net FCL Stack class

**The goal for this exercise** is to make use of a Stack class that's already been built for you (as part of the .Net framework), and to review the Big "Oh" notation.

In the last exercise, you figured out what a Stack abstract data type is. In this exercise, you should try writing some C# code using the Stack class that is provided as part of the .Net Framework Class Library.

**What you need to do for this exercise:**  In the starter solution, in the **PCE\_Starter** project, fill in the Using\_DotNets\_Stack class.

1. In the RunExercise method create a stack, push the numbers 1, 2, and 3 onto it, and the print out the values as you pop them back off the stack. Make sure that you use the Push, Pop, and Peek methods in your code - you should be able to do this with less than 25 lines of code in total. (Note that there are many, many other methods – feel free to skim through those, but don't worry about anything other than Push, Pop, Peek, and the constructors)

Hint: You'll need to put using System.Collections; at the top of your C# file, right under using System; (if it isn't there already)

Hint: The following may (or may not) be useful:
<http://www.dotnetperls.com/stack>

1. Also, in your comments above the RunExercise method, briefly explain, in your own words what the following description means, when it talks about O(1) operation, and O(N) operations. What does N refer to? Can you offer up a plausible explanation as to why these operations take the time that they do?

|  |
| --- |
| The capacity of a **Stack** is the number of elements the **Stack** can hold. The default initial capacity for a **Stack** is 10. As elements are added to a **Stack**, the capacity is automatically increased as required through reallocation.If Count is less than the capacity of the stack, Push is an O(1) operation. If the capacity needs to be increased to accommodate the new element, **Push** becomes an O(n) operation, where n is **Count**. Pop is an O(1) operation. |

(This is taken directly from the 2005 Visual Studio online help)

Note: The objective here is to get you familiar with the syntactic details how one might implement a Stack, using the Stack class built-in to the .Net Framework Class Library as an example. So if the coding portion of this exercise seems really easy to you, that is good, and normal! ☺