

Lab Exercise 4.2: More testing using NUnit – but now with coverage

In this exercise, you will write some more (state based) tests using NUnit. You will continue working with the code/exercise from last time, the StackCalculator class.

The recommended approach is to write and test one method to completeness (using coverage to verify that you have indeed tested all your code) at a time.

Exercise 4.2-A:

Install reSharper/dotCover if you have not already done so.

Exercise 4-B:

Add these methods/signatures to the IStackCalculator interface:

```
void Swap();           // Swap the two topmost values on the stack
                        // Throws InvalidOperationException if there are less than
                        // 2 values on the stack

void RotateUp();       // Rotate the values on the stack, so that all values move up by one, and
                        // the topmost value moves to the bottom

void RotateDown();     // Rotate the values on the stack, so that all values down up by one, and
                        // the bottommost value moves to the top
```

Exercise 4-C:

Implement these 3 new methods in the StackCalculator class and test them (implement and test one method at a time). Use dotCover to verify that you have tested all parts of the code.

You should write tests so that you try to use both “classic” Assert’s and constraints (i.e. using Expect(<expression> , <constraint>).