Lab Exercise: Unit-testing class `Calculator`

In this exercise, you will use your existing implementation of class `Calculator` to create your first NUnit test suite of an application class. You will organize your application code and test code in separate projects and "link" them together, just as you saw in the introductory video `NUnitIntro.mp4` for this class. If you have *not* watched the video, do so now!

Also, it is a prerequisite for this exercise that you have downloaded and installed NUnit.

In Lab Exercise 1, you tested class `Calculator` as well as you could using hand-testing. In this exercise you will use NUnit to test the class again.

Exercise 1

1. Create a solution for this exercise.
2. Copy the application project from Exercise 1 to the solution folder and include the project in the solution.
3. Add another class library project to the solution. You should name this project `Calculator.Test.Unit`. In this project, add a reference to the application project and a reference to `NUnit.Framework.dll.`

Exercise 2

Add a new C# source file to the test project (that's `Calculator.Test.Unit).` In this file, define the class Calculator`UnitTests`. This class will hold all your unit tests for class `Calculator`.

Exercise 3

Implement your unit tests in the file you added to the test project above – test the class `Calculator` as thoroughly as you can using unit tests. Is it difficult?

Exercise 4

Compare your test in Lab Exercises 1 and 2 and reflect on hand-testing vs. unit testing with a framework:

Extensibility Which form of test is easiest to extend, e.g. if new functionality is required for class `Calculator`?

Maintainability Which form of test is easiest to maintain?

Readability Imagine you are new to the project. Which form of test is easiest to read?

Automation Which type of test is easiest to automate? If you wanted to collect and compare test results every 15 minutes, which kind of test is it easiest to see if passed or failed?