Lab Exercise 2: Testing class WeighingUnit with a Unit Test Framework

In this exercise, you will use your implementation of class `WeighingUnit` from Lab Exercise 1 to do your first unit tests using a unit test framework. You will download and install NUnit and subsequently set up your project to use it. Then, you will do the first couple of unit tests. Finally, you will reflect on how the hand-testing and the framework unit testing relate.

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

Exercise 1
Download NUnit and install it:

1. Fetch the latest stable version from http://nunit.org/ (use the .msi – installer version, that's the easiest way).
2. Run the installer and install NUnit on your PC.

Exercise 2
Add a new project to the solution from Lab Exercise 1. You should name this project `WeighingUnit.Tests`. Add a reference to `NUnit.Framework.dll` to this new project.

Exercise 3
Add a new C# source file to the project called `WeighingUnitTests`. In this file, define the class `WeighingUnitTests`. This class will hold all your unit tests for class `WeighingUnit`.

Exercise 4
Implement your unit tests – test the class `WeighingUnit` as thoroughly as you can using unit tests. Is it difficult?

Exercise 5
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 `WeighingUnit`?

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?