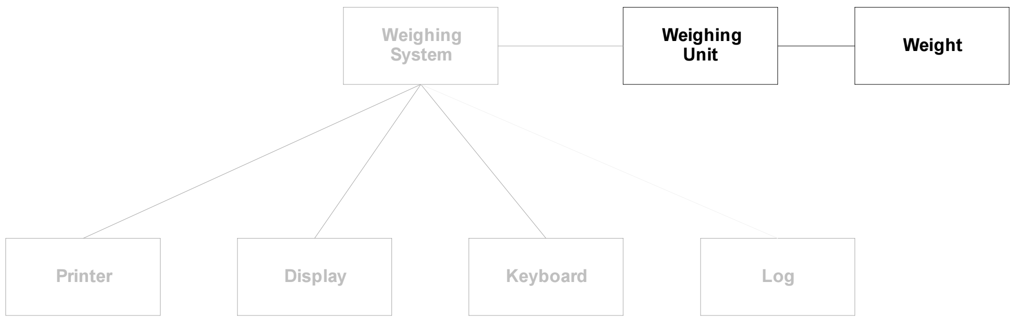
# Lab Exercise 1: Hand-testing class WeighingUnit

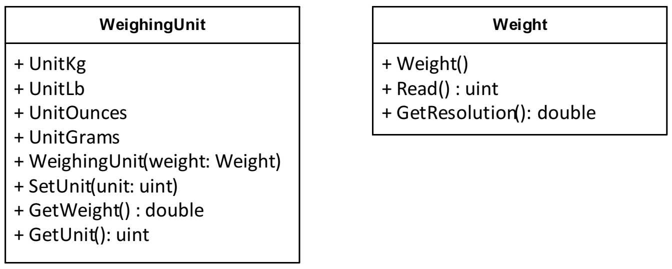
In this exercise, you will implement a simple class `WeighingUnit` which interfaces a physical weight, represented by a class `Weight`. You will then test the class `WeighingUnit` as thoroughly as possible.

---

As introduced on class, *Weighing System* is intended to be the software in a generic weighing system. In this exercise we will focus on part of this system, the `WeighingUnit` and the pertaining class `Weight`, see the below diagram.



The two classes `WeighingUnit` and `Weight` are defined by the UML class diagrams given below.



Partial description of class `Weight`:

| | |
|---|---|
| `Read()` | Returns a 10-bit value (range [0..1023] ) representing the current reading of the weight |
| `GetResolution()` | Returns the resolution of the read value in milligrams. |

I.e., if `GetResolution()` returns *500*, the weight is read in half-gram increments. If, subsequently, `Read()` returns *753*, it means that the weight currently reads 376.5 grams (+/- 0.25 grams)

Partial description of class `WeighingUnit`:

| | |
|---|---|
| `UnitXXX` | Constant values representing the unit in which a weight reading should be returned (kilograms, lbs, solid ounces or grams) |
| `GetWeight()` | Returns the current weight in the unit specified by the user |

***IMPORTANT***
It is very important that you understand that the focus of this exercise is on the test of `WeighingUnit` – *not* any other part of the system. You may have to implement some other parts of the system, but if so, it should be with the purpose of testing `WeighingUnit`.

***IMPORTANT***
Just once more: Your focus in this exercise is on the test of class `WeighingUnit`.

**Exercise 1:**
What is the focus of this exercise?

**Exercise 2:**
Download and open the Visual Studio 2010 solution provided for this exercise from CampusNet. Make sure you familiarize yourself with the structure: The solution contains 2 projects:

- The application project (`WeighingSystem`) containing a skeleton source file for `WeighingUnit`, `WeighingUnit.cs`
- The test project (`WeighingSystem.HandTest`) which contains a file `WeighingSystemHandTest.cs` containing the main program and all necessary references to the application project.

The idea behind this structure is that you implement the *application* code in the *application* project and the *test* code in the *test* project.

**Exercise 3:**
Implement class `WeighingUnit` in the supplied application project. *Do not* change the class' defined interface (i.e. the class methods' signature). You may *add* methods etc., but do not change the signature of the ones already defined. Also, you may find that you need to add one or more other classes to test class `WeighingUnit`. If so, do it.

**Exercise 4:**
Test class `WeighingUnit` as well as you can. If you find any errors, correct them and re-write your test program. Note any problems you have in the testing of the system.