Exercise 15 (optional): Temperature Sensor Library

Equipment

For this exercise you will need:

- 1 x Arduino Uno
- 1 x TMP36GZ *and/or* LM35DZ

Remember to include the header file (not the c file) in your .ino file!

Reading

Chapter 13

Setup

- Connect the temperature sensor.
- Create a new tab in the IDE using either **ctrl + shift + N** or by clicking the three dots in the upper-right corner. Name this file *TemperatureSensor.h*
- Make another tab and name it *TemperatureSensor.cpp*

Questions & Exercises

The goal of this optional exercise is to show the basic construction of a C-library and how to include them in your sketches.
15a: First, let's start by constructing the *header* file. The purpose of the header file is to define the functionality of your C file, without explicitly expressing HOW it is done. Examine the following header and C++ file, take note of how they are constructed and fit together:

```
// ExampleHeader.h
// This is known as a header guard, in short, it ensure that no file is included
    more than once
// as this would cause issues with "multiple definitions" of the same class/
   function
#ifndef EXAMPLEHEADER_H_
#define EXAMPLEHEADER_H_

#include <WString.h>

// This is an example class for assigning a (student) ID to a name
class Student{
  private:
    // Classes are by default private, but we usually write it out for clarity
    // Members are normally kept private as we don't want external actors to
       change them directly
    // without good reason
    String name;
    int id;

  public:
    // We also define a simple function to change ID, notice that we only
       include the return type, input and function name
    void changeID(int newID);

    // Rather than manually assigning values to each member, we can create a
       constructor.
    // The constructor has the same name as the class and is automatically
       recognised.
    // In the following code you will see that it has as input all the members
       we wish to assign followed by :
    // afterward it sets member = inputs using the syntax member{inputVariable},
        member{otherInput}
    // This means that to create an instance of the class we now only need to
       call Student <name_of_variable>(<name>, <id>)
```

```
    // Example: Student theo("Theo", 0)
    Student(String studentName, int studentID) : id{studentID}, name{studentName
        }
    {
      // You could have the constructor do something extra, like calculations,
         before/after assigning
      // variables. However, it is not necessary if all you need it to do is
         assign members
    };
}


#endif // Since ifndef is an if statement, we also have to manually tell the
    compiler (preprocessor) that it ends here
```

With the corresponding C++ file

```
// ExampleHeader.cpp
// Include the corresponding header file
// This also includes every include in the header file, i.e. WString is now
    included as well
#include "ExampleHeader.h"

// We now want to specify the functionality of the changeID function
// This function is linked to the definition in the header file and is
// what will be called when you call Student.changeID(<new_id>)
void Student::changeID(int newID){
  this->id = newID;
}
```

15b: Based on the previous examples, write a header/C++ file-pair and construct a class for the temperature sensors. It should include:

- A type member, for example, a string, to signify which type of temperature sensor it is.
- To make the library portable, we want to support other input voltages as well. For example, if you are using the ESP8266, you apply 3.3V rather than 5.0, so it should be possible to set this in the class. Handle this by introducing an input voltage member.
- A constructor
- A function which converts the analog input from your sensor into voltage (like the one you made in exercise 9

15c: Remake your sensor script from exercise 9 using the library, verify that it works
15d: Expand the conversion to also work for the other sensor, i.e. it should be able to correctly convert for both the TMP36 and the LM35. You will need the data sheet for this (if you need a sensor to test with, let us know).