„C++ / GUI “
Exercise 16

namespaces: quick and dirty

There are two ways of using namespace commands: (1) keep the global namespace as clean as possible, or (2) work around the namespaces someone else decided to use for whatever reason. C++ supports both!

Assume a source code file that starts as follows:

```
#include <iostream>
using std::cout;

namespace thisIsAReallyLongName
{
        void f()
        { cout << "hello\n"; }

        namespace name
        {
                void f() { cout << "goodbye\n"; }
        }
}

// das aequivalent zu static in C
namespace
{
        void g() { cout << "moin\n"; }
        void h() { cout << "yes\n"; }
}

void h() { cout << "oh no\n"; }
```

1. What does `f();` ?
2. What does `thisIsAReallyLongName::f();` ?
3. What does `namespace l = thisIsAReallyLongName; l::f();` ?
4. What does `using l::f; f();` ?
5. What does `using namespace l::name; f();` ?
6. What does `using l::name::f; f();` ?
7. How can you call h() in the anonymus namespace?

8. How can you call the global h()?

C++ namespaces

Most functions of the C–library are also defined for C++. `printf` is defined in the file `cstdio` inside the namespace `std`.

LC namespaces

No-one will call her capacitor class C, because of easily possible name clashes. What are useful namespaces for the LC circuit?