NAMESPACES AND FUNCTION OVERLOADING

Name collisions and how to avoid them

Florian Warg, Max Staff

April 23, 2017

NAME COLLISIONS

- a project uses libraries A and B
- A and B offer functions with similar names and arguments

```
/* Library A */
digest hash(block input) { return md5(input); }
/* Library B */
digest hash(block input) { return sha1(input); }
/* User code */
hash(my_message);
```

- the compiler cannot deduce which function you want to call
- this is called a name collision
- it will cause a compiler error

NAMESPACES

- libraries use namespaces to avoid name collisions
- namespaces can contain functions, types and objects
- symbols can be accessed with ns::symbol

```
namespace LibA {
    digest hash(block input);
}
namespace LibB {
    digest hash(block input);
}
/* call hash from Lib A */
LibA::hash(my_message);
/* call hash from Lib B */
LibB::hash(my_message);
```

NESTED NAMESPACES

- namespaces can also contain other namespaces
- this can be used to organize software into packages

```
namespace Lib {
    namespace crypto {
        digest hash(block input) { ... }
    }
}
Lib::crypto::hash(my_message);
```

SPLIT NAMESPACES

- namespaces can be opened and closed at any point
- symbols inside the same namespace are visible to each other

```
namespace A {
    int x;
} /* namespace A */

namespace A {
    int y;
} /* namespace A */
```

IMPORT SYMBOLS

- symbols can be imported into the local namespace
- achieved with a ”using directive”

```
1 #include <string>
2 using std::string;
3 int main() {
4     std::string s1;
5     string s1; // imported std::string as string
6 }
```

IMPORT NAMESPACES

- whole namespaces can be imported, too
- this will import all symbols

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int main() {
5     string name; // std::string
6     cin >> name; // std::cin
7     cout << "Hello, " << name << "\n";
8 }
```

BEST PRACTICE

- global namespace should have as few symbols as possible
- never import symbols in header files
- full symbol name is easier to understand in code reviews
- import single symbols rather than namespaces
- in production code almost always use full names

FUNCTION OVERLOADING