More on attributes

This lesson touches upon some more features of the attributes in C++ 17

WE'LL COVER THE FOLLOWING

Ignoring unknown attributes

The feature is mostly for clarification.

Before C++17, if you tried to use some compiler-specific attribute, you might even get an error when compiling in another compiler that doesn't support it. Now, the compiler omits the attribute specification and won't report anything (or just a warning). This wasn't mentioned in the Standard, and it needed clarification.

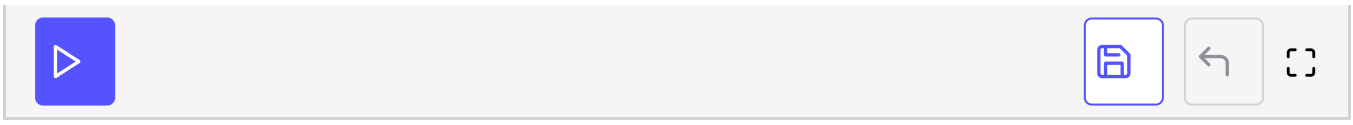
```
#include <iostream>
using namespace std;

// compilers which don't
// support MyCompilerSpecificNamespace will ignore this attribute

[[MyCompilerSpecificNamespace::do_special_thing]]

void foo();

int main() {}
```



For example in GCC 7.1 there's a warning:

warning: '`MyCompilerSpecificNamespace::do_special_thing`' scoped attribute directive ignored `[-Wattributes] void foo()`;

> *Extra Info:* The change was described in P0283R2(wording) and P0283R1(reasoning).

Attribute namespaces without repetition

The feature simplifies the case where you want to use multiple attributes, like:

```
void f() {
    [[rpr::kernel, rpr::target(cpu,gpu)]] // repetition
    doTask();
}
```

Proposed change:

```
void f() {
    [[using rpr: kernel, target(cpu,gpu)]]
    doTask();
}
```

That simplification might help when building tools that automatically translate annotated code of that type into different programming models.

> *Extra Info:* The change was described in: P0028R4.

The next lesson provides a brief summary of all the concepts introduced in this chapter!