Before C++11

This section explains how attributes used to work before C++ 11.

WE'LL COVER THE FOLLOWING

In the era of C++98/03, each compiler introduced its own set of annotations, usually with a different keyword.

Often, you could see code with `#pragma`, `__declspec`, `__attribute` spread throughout the code.

Here's the list of the common syntax from GCC/Clang and MSVC:

GCC Specific Attributes

GCC uses annotation in the form of `__attribute__((attr_name))`. For example:

```
int square (int) __attribute__ ((pure)); // pure function
```

Documentation:

- Attribute Syntax - Using the GNU Compiler Collection (GCC)
- Using the GNU Compiler Collection (GCC): Common Function Attributes

MSVC Specific Attributes

Microsoft mostly used `__declspec` keyword, as their syntax for various compiler extensions.

> See the documentation here: `__declspec` Microsoft Docs.

```
__declspec (deprecated) void LegacyCode() { }
```

Clang Specific Attributes

Clang, as it's straightforward to customize, can support different types of annotations. Most of GCC attributes work with Clang.

> See the documentation here: Attributes in Clang — Clang documentation.

Now that you've learned how attributes used to work before C++ 11. Let's learn about the specific attributes in C++ 11 and attribute additions in C++ 14 in the next lesson.