The `[[deprecated]]` Attribute

The standard attribute `[[deprecated]]` indicates that the use of the entity to which the attribute pertains is discouraged, typically in the form of a compiler warning.

Description

The standard `[[deprecated]]` attribute is used to portably indicate that a particular **entity** is no longer recommended and to actively discourage its use. Such deprecation typically follows the introduction of alternative constructs that are superior to the original one, providing time for clients to migrate to them *asynchronously* before the deprecated one is removed in some subsequent release.

An asynchronous process for ongoing improvement of legacy codebases, sometimes referred to as **continuous refactoring**, often allows time for clients to migrate — on their own respective schedules and time frames — from existing *deprecated* constructs to newer ones, rather than having every client change in lock step. Allowing clients time to move *asynchronously* to newer alternatives is often the only viable approach unless (1) the codebase is a closed system, (2) all of the relevant code is governed by a single authority, and (3) the change can be made mechanically.

Although not strictly required, the Standard explicitly encourages[1] conforming compilers to produce a diagnostic message in case a program refers to any **entity** to which the `[[deprecated]]` attribute pertains. For instance, most popular compilers emit a warning whenever a `[[deprecated]]` function or object is used:

```
                void f();
[[deprecated]]  void g();

                int a;
[[deprecated]]  int b;

void h()
{
    f();
    g();  // Warning: g is deprecated.
    a;
    b;    // Warning: b is deprecated.
}
```

[1] The C++ Standard characterizes what constitutes a well-formed program, but compiler vendors require a great deal of leeway to facilitate the needs of their users. In case any feature induces warnings, command-line options are typically available to disable those warnings (`-Wno-deprecated` in GCC), or methods are in place to suppress those warnings locally~~;~~ e.g., `#pragma GCC diagnostic ignored "-Wdeprecated"`.