Assertions at Compile Time

In this lesson, we will learn about assertions at compile time in modern C++.

`static_assert` is the tool in modern C++ used to make our code safe.

static_assert

The usage of `static_assert` is quite easy. `static_assert` requires an expression and a string. The expression must be predicate that can be evaluated at compile time. Predicate means the expression returns `true` or `false`. If the expression evaluates to `false`, we will get an error message at compile-time with the string as a message. Of course, we get no executable.

There are a few points we must consider.

- The `static_assert` expression will be evaluated at compile-time, and we have no runtime overhead.
- Expressions that can be evaluated at compile time are called constant expressions.
- We can use `static_assert` expressions in all parts of our program. Therefore, it is a good idea to put general requirements on our source code in a separate header. As a result, the `static_assert` expression will be automatically verified at compile time if we include the header. This helps to port our code to a new platform since we can easily check if the new platform supports the type requirements.

The example in the next lesson will build on your understanding of this concept.