Thread-Safe Initialization

This lesson gives a brief introduction to the thread safe initialization of variables in concurrent programming with C++.

If a variable is never modified, there is no need for synchronization by using an expensive lock or an atomic. We only have to ensure that it is initialized in a thread-safe way.

There are three ways to do this in C++:

- 1. Constant expressions.
- 2. The function std::call_once in combination with the flag std::once_flag.
- 3. A static variable with block scope.



Thread-safe initialization in the main-thread

The easiest and fourth way to initialize a variable in a thread-safe way: initialize the variable in the main-thread before we create any child threads.

In the next lesson, we'll discuss thread-safe initialization from the perspective of concurrency in C++ with Constant Expressions.