# Thread-Safe Initialization

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

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

There are three ways in C++ to initialize variables in a thread-safe way.

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 initialisation in the main-thread**
>
> The easiest and fourth way to initialise a variable in a thread-safe way: initialise the variable in the main-thread before you create any child threads.

We will explain each thread-safe initialization method in the next 3 lessons.