Introduction

Let's see which concept will we deal with in this section.

Having finished the chapters on language fixes and clarifications, we're now ready to look at wide-spread features. Improvements described in this section also have the potential to make your code more compact and expressive.

For example, with Structured bindings, you can leverage much easier syntax tuples (and tuple-like expressions). Something easy in other languages like Python is now possible with C++17.

In this chapter, you'll learn:

- Structured bindings/Decomposition declarations
- How to provide Structured Binding interface for your custom classes
- Init-statement for `if`/`switch`
- *Inline variables* and their impact on header-only libraries
- Lambda expressions that might be used in a `constexpr` context
- How to properly wrap the `this` pointer in lambda expressions
- Simplified use of **nested namespaces**
- How to test for header existence with `__has_include` directive

Let's get started with structured binding declarations.