Introduction

Functions, function objects, and lambda functions are all part of the callable units class. They are 'called' to retrieve data or perform an action.

> **ℹ This chapter is intentionally not exhaustive**
> This course is about the C++ Standard library, therefore, it will not go into detail about callable units. We will provide only as much information as is necessary to use them correctly in the algorithms of the Standard Template Library. An exhaustive discussion of callable units should be part of a course about the C++ core language.

Many of STL algorithms and containers can be parametrized with callable units or short callables. **A callable is something that behaves like a function**. These are not only functions but also function objects and lambda functions. Predicates are special functions that return a boolean as a result. If a predicate has one argument, it's called a unary predicate. If a predicate has two arguments, it's called a binary predicate. The same holds true for functions. A function taking one argument is a unary function; a function taking two arguments is a binary function.

> **🔍 To change the elements of a container, our algorithm must get them by reference**
> Callables can receive their arguments by value or by reference from their container. To modify the elements of the container, they must address them directly, so it is necessary that the callable gets them by reference.

In the next lesson, we'll learn about two types of callables: functions and functions objects.