
Chapter 5. Utilities

This chapter describes the general utilities of the C++ standard library. These utilities are small and simple classes, types, or functions that perform frequently needed tasks:

- Class `pair<>` and class `tuple<>`
- Smart pointer classes (class `shared_ptr<>` and class `unique_ptr`)
- Numeric limits[1]

[1] One could argue that numeric limits and class `ratio<>` should be part of Chapter 17, which covers numerics, but these classes are used in some other parts of the library, so I decided to describe them here.

- Type traits and type utilities
- Auxiliary functions (for example, `min()`, `max()`, and `swap()`)
- Class `ratio<>` [1]
- Clocks and timers
- Some important C functions

Most, but not all, of these utilities are described in clause 20, “General Utilities,” of the C++ standard. The rest are described along with more major components of the library either because they are used primarily with that particular component or due to historical reasons. For example, some general auxiliary functions are defined as part of the `<algorithm>` header, although they are not algorithms in the sense of the STL (which is described in Chapter 6).

Several of these utilities are also used within the C++ standard library. For example, type `pair<>` is used whenever two values need to be treated as a single unit — for example, if a function has to return two values or when elements of containers are key/value pairs — and type traits are used wherever complicated type conversions are necessary.