The Comparison Criterion

This lesson talks about the rules followed by ordered associative containers when comparing the values inside them.

The default comparison criterion of the ordered associative containers is std::less. If we want to use a *user-defined* type as the key, we have to overload the operator <. It's sufficient to overload the operator < for our data type because the C++ runtime compares, with the help of the relation (! (elem1<elem2 || elem2<elem1)), two elements for equality.

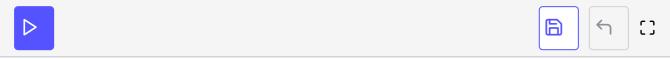
We can specify the sorting criterion as a template argument that must implement a *strict weak ordering*.

i Strict weak ordering

Strict weak ordering for a sorting criterion on a set 5 is given if the following requirements are met:

- For every s from S it has to hold that s < s is not possible.
- For all s1 and s2 from S it must hold: If s1 < s2, then s2 < s1 is not possible.
- For all s1, s2 and s3 with s1 < s2 and s2 < s3 the following must hold: s1 < s3.
- For all s1, s2 and s3 with s1 not comparable with s2 and s2 not comparable with s3 the following must hold: s1 is not comparable with s3.

In contrast to the definition of the *strict weak ordering*, the usage of a comparison criterion with *strict weak ordering* is a lot simpler for an std::map.



In the next lesson, we'll discuss functions used for searching.