Overview

Let's start our discussion on ordered associative containers by defining some simple characteristics.

The ordered associative container `std::map` and `std::multimap` associate their key with a value. Both are defined in the header `<map>`. `std::set` and `std::multiset` need the header `<set>`.

All four ordered containers are parametrized by their type, their allocator and their comparison function. The containers have default values for the allocator and the comparison function, depending on the type. The declaration of `std::map` and `std::set` shows this very nicely.

```
template < class key, class val, class Comp= less<key>,
           class Alloc= allocator<pair<const key, val> >
class map;

template < class T, class Comp = less<T>,
           class Alloc = allocator<T> >
class set;
```

The declaration of both associative containers shows that `std::map` has an associated value. The key and the value are used for the default allocator: `allocator<pair<const key, val>>`. With a little bit more imagination, you can derive more from the allocator. `std::map` has pairs of the type `std::pair<const key, val>`. The associated value `val` does not matter for the sort criteria: `less<key>`. All observations also hold for `std::multimap` and `std::multiset`.