Ordered Associative Containers

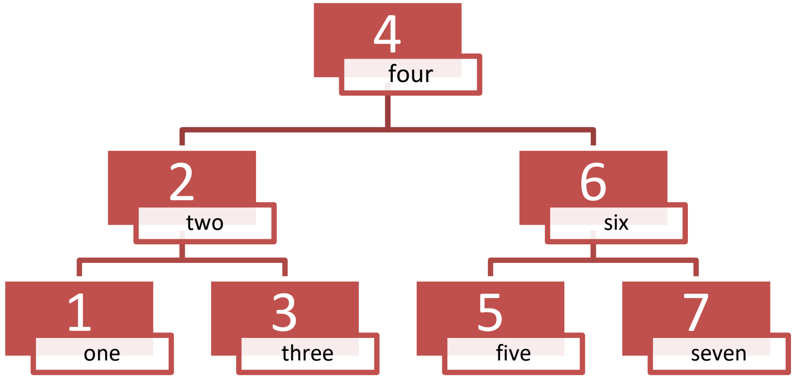
`std::set`, `std::multiset`, `std::map`, and `std::multimap`

- require a data type(s), an allocator (default), and a comparison function (default).
- support similar functionality.
- provide a key/value association.

Ordered Associative Containers	Value Associated	More Identical Keys	Header
`std::set`	no	no	<set>
`std::multiset`	no	yes	<set>
`std::map`	yes	no	<map>
`std::multimap`	yes	yes	<map>

The elements of `std::map` and `std::multimap` are pairs of type `std::pair<const key, value>`.

Ordered Associative Containers



```
std::map<int,std::string> int2String{ {3, "three"}, {2, "two"}, {1, "one"}, {5, "five"},
{6, "six"}, {4, "four"}, {7, "seven"} };
```

Ordered associative Containers are

- binary, balanced search trees.
- sorted in ascending order (default).

Ordered Associative Containers

`std::map`

- is the most popular ordered associative container.
- supports the index operator [ ].

Index operator [ ]

- enables the reading and writing access.
- creates a new key/value pair if the key is not available.

  ➡ invokes the default constructor for the value.

`at` operator

- Allows to read a key without creating the value ➡ `std::out_of_range` exception

`stlOrderedAssociativeContainerAssociativeArray.cpp`
`stlOrderedAssociativeContainersModify.cpp`