

The min, max and minmax functions

This family of functions allows us to find the minimum and maximum in a set of data. Let's find out how.

WE'LL COVER THE FOLLOWING



- Required Headers
- `std::min`, `std::max` and `std::minmax`

Required Headers

The many variations of the `min`, `max` and `minmax` functions apply to values and initializer lists. These functions need the header `<algorithm>`. Nearly the same holds for the functions `std::move`, `std::forward` and `std::swap`. You can apply them to arbitrary values. These three functions are defined in the header `<utility>`.

`std::min`, `std::max` and `std::minmax`

The functions `std::min`, `std::max` and `std::minmax`, defined in the header `<algorithm>`, act on values and initialiser lists and give you the requested value back as result. In the case of `std::minmax`, you get an `std::pair`. The first element of the pair is the minimum, the second the maximum of the values. By default, the less operator (`<`) is used, but you can specify your comparison operator. This function needs two arguments and returns a boolean. Functions that either return true or false are called predicates.

```
// minMax.cpp
#include <iostream>
#include <algorithm>
//...
using std::cout;
//...
int main(){
    cout << "std::min(2011, 2014):\t\t\t ";
    cout << std::min(2011, 2014)<<"\n";
    // 2011

    cout << "std::min({3, 1, 2011, 2014, -5}):\t";
```



```

cout << std::min({3, 1, 2011, 2014, -5})<<"\n"; // -5

cout << "std::min(-10, -5, [](...) {...}):\\t\\t";
cout << std::min(-10, -5, [](int a, int b)
    { return std::abs(a) < std::abs(b); })<<"\\n\\n"; // -5

std::pair<int, int> pairInt= std::minmax(2011, 2014);
auto pairSeq= std::minmax({3, 1, 2011, 2014, -5});
auto pairAbs= std::minmax({3, 1, 2011, 2014, -5}, [](int a, int b)
    { return std::abs(a) < std::abs(b); });

cout << "pairInt.first, pairInt.second:\\t\\t";
cout << pairInt.first << ", " << pairInt.second << "\\n"; // 2011,2014

cout << "pairSeq.first, pairSeq.second:\\t\\t";
cout << pairSeq.first << ", " << pairSeq.second << "\\n"; // -5,2014

cout << "pairAbs.first, pairAbs.second:\\t\\t ";
cout << pairAbs.first << ", " << pairAbs.second << "\\n"; // 1,2014

return 0;
}

```



The functions `std::min`, `std::max`, and `std::minmax`

The table provides an overview of the functions `std::min`, `std::max` and `std::minmax`

Function	Description
<code>min(a, b)</code>	Returns the minimal value of <code>a</code> and <code>b</code> .
<code>min(a, b, comp)</code>	Returns the minimal value of <code>a</code> and <code>b</code> according to the predicate <code>comp</code> .
<code>min(initializer list)</code>	Returns the minimal value of the initializer list.
<code>min(initializer list, comp)</code>	Returns the minimal value of the initializer list according to the predicate <code>comp</code> .

`max(a, b)`

Returns the maximal value of `a` and `b`.

`max(a, b, comp)`

Returns the maximal value of `a` and `b` according to the predicate `comp`.

`max(initializer list)`

Returns the maximal value of the initializer list.

`max(initializer list, comp)`

Returns the maximal value of the initializer list according to the predicate `comp`.

`minmax(a, b)`

Returns the minimal and maximal value of `a` and `b`.

`minmax(a, b, comp)`

Returns the minimal and maximal value of `a` and `b` according to the predicate `comp` according to the predicate `comp`.

`minmax(initializer list)`

Returns the minimal and maximal value of the initializer list.

`minmax(initializer list, comp)`

Returns the minimal and maximal value of the initializer list according to the predicate `comp`.

The variations of ``std::min``, ``std::max`` and ``std::minmax``

Now, let's talk about another useful function the `std::move`.