

## Re: Sorting a map by value

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.cpp/2004-08/3264.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-08/3264.html)

---

**From:** Richard Herring (*junk\_at\_[127.0.0.1]*)

**Date:** 08/25/04

Date: Wed, 25 Aug 2004 12:19:54 +0100

In message <opsc9yjwt16jxvii7@localhost.localdomain>, Kevin W. <contact@in.sig> writes

>>> *How do I sort a map by the value, rather than the key?*

>>

>> *You can't sort maps. They are always automatically sorted by key.*

>

>*Yes, but I should be able to define the "less than" predicate in the constructor,*

Yes, and it defines the ordering for the lifetime of the map. Once established, you can't change it.

>*or I should be able to pass a custom predicate into the sort function*

No, because there is no standard sort function which can be applied to maps. `std::sort()` needs random-access non-const iterators.

>*(as I understand it, you can sort a map because it is only*

>*automatically sorted when pairs are added or removed).*

No. As Kai-Uwe Bux has pointed out, the key element of each pair is const. But even if you could rearrange their order, it wouldn't do you any good, because the key lookup algorithm depends on the elements being correctly ordered. The next time you tried to access an element by key, you'd get UB.

>

>*What I'm really asking is:*

>*What are the arguments to this functor (values, pointers or references),*

The same as to `std::less`, namely (const reference to) `key_type`, and it returns `bool`. And it must implement a strict weak ordering:

`a < b && b < c => a < c;`

`eq(a,b) && eq(b,c) => eq(a,c)` where `eq(a,b)` means `!(a < b) && !(b < a)`

>*and*

>*How do I access the value from these arguments?*

comp.lang.c++. Re: Sorting a map by value

>

The arguments `_are_` the values (or references to them.)

--

Richard Herring