

Username: Pralay Patoria **Book:** The C++ Standard Library: A Tutorial and Reference, Second Edition. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

8.1. Type Definitions

container :: **value _ type**

- The type of elements.
- For (unordered) sets and multisets, it is constant.
- For (unordered) maps and multimaps, it is `pair <const key-type , mapped-type >`.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: **reference**

- The type of element references.
- Typically: `container ::value_type&`.
- For `vector<bool>`, it is an auxiliary class ([see Section 7.3.6, page 282](#)).
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: **const_reference**

- The type of read-only element references.
- Typically: `const container ::value_type&`.
- For `vector<bool>`, it is `bool`.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered map, unordered multimap, string.

container :: **iterator**

- The type of iterators.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: **const_iterator**

- The type of read-only iterators.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: **reverse_iterator**

- The type of reverse iterators.
- Provided by array, vector, deque, list, set, multiset, map, multimap, string.

container :: **const_reverse_iterator**

- The type of read-only reverse iterators.
- Provided by array, vector, deque, list, set, multiset, map, multimap, string.

container :: **pointer**

- The type of pointers to elements.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: **const_pointer**

- The type of read-only pointers to elements.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: **size_type**

- The unsigned integral type for size values.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: **difference_type**

- The signed integral type for difference values.

- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

container :: key_type

- The type of the key of the elements for associative and unordered containers.
- For (unordered) sets and multisets, it is equivalent to **value_type**.
- Provided by set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap.

container :: mapped_type

- The type of the value part of the elements of associative and unordered containers.
- Provided by map, multimap, unordered map, unordered multimap.

container :: key_compare

- The type of the comparison criterion of associative containers.
- Provided by set, multiset, map, multimap.

container :: value_compare

- The type of the comparison criterion for the whole element type.
- For sets and multisets, it is equivalent to **key_compare**.
- For maps and multimaps, it is an auxiliary class for a comparison criterion that compares only the key part of two elements.
- Provided by set, multiset, map, multimap.

container :: hasher

- The type of the hashing function of unordered containers.
- Provided by unordered set, unordered multiset, unordered map, unordered multimap.

container :: key_equal

- The type of the equality predicate of unordered containers.
- Provided by unordered set, unordered multiset, unordered map, unordered multimap.

container :: local_iterator

- The type of the bucket iterators of unordered containers.
- Available since C++11.
- Provided by unordered set, unordered multiset, unordered map, unordered multimap.

container :: const_local_iterator

- The type of read-only bucket iterators of unordered containers.
- Available since C++11.
- Provided by unordered set, unordered multiset, unordered map, unordered multimap. In addition, type **allocator_type** is provided for all containers except arrays ([see Section 8.10.1, page 430](#)).