

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

# Multimap in C++ Standard Template Library (STL)

Difficulty Level : Easy • Last Updated : 17 Jan, 2022

Multimap is similar to a [map](#) with the addition that multiple elements can have the same keys. Also, it is NOT required that the key-value and mapped value pair have to be unique in this case. One important thing to note about multimap is that multimap keeps all the keys in sorted order always. These properties of multimap make it very much useful in competitive programming.

## Some Basic Functions associated with multimap:

- [begin\(\)](#) – Returns an iterator to the first element in the multimap
- [end\(\)](#) – Returns an iterator to the theoretical element that follows last element in the multimap
- [size\(\)](#) – Returns the number of elements in the multimap
- [max\\_size\(\)](#) – Returns the maximum number of elements that the multimap can hold
- [empty\(\)](#) – Returns whether the multimap is empty
- [pair<int,int> insert\(keyvalue,multimapvalue\)](#) – Adds a new element to the multimap

## C++ implementation to illustrate above functions:

### CPP

```
// CPP Program to demonstrate the implementation of multimap
#include <iostream>
#include <iterator>
#include <map>
using namespace std;
```



Start Your Coding Journey Now

[Login](#)[Register](#)

```

{
    multimap<int, int> gquiz1; // empty multimap container

    gquiz1.insert(pair<int, int>(1, 40));
    gquiz1.insert(pair<int, int>(2, 30));
    gquiz1.insert(pair<int, int>(3, 60));
    gquiz1.insert(pair<int, int>(6, 50));
    gquiz1.insert(pair<int, int>(6, 10));

    // printing multimap gquiz1
    multimap<int, int>::iterator itr;
    cout << "\nThe multimap gquiz1 is : \n";
    cout << "\tKEY\tELEMENT\n";
    for (itr = gquiz1.begin(); itr != gquiz1.end(); ++itr) {
        cout << '\t' << itr->first << '\t' << itr->second
            << '\n';
    }
    cout << endl;

    // adding elements randomly,
    // to check the sorted keys property
    gquiz1.insert(pair<int, int>(4, 50));
    gquiz1.insert(pair<int, int>(5, 10));

    // printing multimap gquiz1 again

    cout << "\nThe multimap gquiz1 after adding extra "
        << "elements is : \n";
    cout << "\tKEY\tELEMENT\n";
    for (itr = gquiz1.begin(); itr != gquiz1.end(); ++itr) {
        cout << '\t' << itr->first << '\t' << itr->second
            << '\n';
    }
    cout << endl;

    // assigning the elements from gquiz1 to gquiz2
    multimap<int, int> gquiz2(gquiz1.begin(), gquiz1.end());

    // print all elements of the multimap gquiz2
    cout << "\nThe multimap gquiz2 after assign from "
        << "gquiz1 is : \n";
    cout << "\tKEY\tELEMENT\n";
    for (itr = gquiz2.begin(); itr != gquiz2.end(); ++itr) {
        cout << '\t' << itr->first << '\t' << itr->second
            << '\n';
    }
}

```

## Start Your Coding Journey Now!

```
// key with value 3 in gquiz2
cout << "\ngquiz2 after removal of elements less than "
```

[Read](#) [Discuss](#) [Courses](#) [Practice](#) [Video](#)

```
gquiz2.erase(gquiz2.begin(), gquiz2.find(3)),
```

[Data Structures and Algorithms](#) [Interview Preparation](#) [Data Science](#) [Topic-wise Practice](#) [C](#) [C++](#)

```
        << '\n';
    }

    // remove all elements with key = 4
    int num;
    num = gquiz2.erase(4);
    cout << "\ngquiz2.erase(4) : ";
    cout << num << " removed \n";
    cout << "\tKEY\tELEMENT\n";
    for (itr = gquiz2.begin(); itr != gquiz2.end(); ++itr) {
        cout << '\t' << itr->first << '\t' << itr->second
            << '\n';
    }

    cout << endl;

    // lower bound and upper bound for multimap gquiz1 key =
    // 5
    cout << "gquiz1.lower_bound(5) : "
        << "\tKEY = ";
    cout << gquiz1.lower_bound(5)->first << '\t';
    cout << "\tELEMENT = " << gquiz1.lower_bound(5)->second
        << endl;
    cout << "gquiz1.upper_bound(5) : "
        << "\tKEY = ";
    cout << gquiz1.upper_bound(5)->first << '\t';
    cout << "\tELEMENT = " << gquiz1.upper_bound(5)->second
        << endl;

    return 0;
}
```

## Output

The multimap gquiz1 is :

KEY	ELEMENT
1	40
2	30

**Start Your Coding Journey Now!**

6 50

6 10

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

The multimap gquiz1 after adding extra elements is :

KEY	ELEMENT
-----	---------

1	40
---	----

2	30
---	----

3	60
---	----

4	50
---	----

5	10
---	----

6	50
---	----

6	10
---	----

The multimap gquiz2 after assign from gquiz1 is :

KEY	ELEMENT
-----	---------

1	40
---	----

2	30
---	----

3	60
---	----

4	50
---	----

5	10
---	----

6	50
---	----

6	10
---	----

gquiz2 after removal of elements less than key=3 :

KEY	ELEMENT
-----	---------

3	60
---	----

4	50
---	----

5	10
---	----

6	50
---	----

6	10
---	----

gquiz2.erase(4) : 1 removed

**Start Your Coding Journey Now!**

3 60

5 10

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)

6

10

```
gquiz1.lower_bound(5) :    KEY = 5          ELEMENT = 10
gquiz1.upper_bound(5) :    KEY = 6          ELEMENT = 50
```

## Function

## Definition

[multimap::operator=](#)

It is used to assign new contents to the container by replacing the existing contents.

[multimap::crbegin\(\) and](#)  
[multimap::crend\(\).](#)

crbegin() returns a constant reverse iterator referring to the last element in the multimap container. crend() returns a constant reverse iterator pointing to the theoretical element before the first element in the multimap.

[multimap::emplace\\_hint\(\).](#)

Insert the key and its element in the multimap container with a given hint.

[multimap::clear\(\).](#)

Removes all the elements from the multimap.

[multimap::empty\(\).](#)

Returns whether the multimap is empty.

[multimap::maxsize\(\).](#)

Returns the maximum number of elements a multimap container can hold.

[multimap::value\\_comp\(\).](#)

Returns the object that determines how the elements in the multimap are ordered ('<' by default).

[multimap::read](#)[Returns a reverse iterator pointing to the theoretical](#)

## Start Your Coding Journey Now!

## Function

## Definition

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)[`multimap::cbegin\(\)` and `multimap::cend\(\)`](#)

`cbegin()` returns a constant iterator referring to the first element in the multimap container. `cend()` returns a constant iterator pointing to the theoretical element that follows the last element in the multimap.

[`multimap::swap\(\)`](#)

Swap the contents of one multimap with another multimap of same type and size.

[`multimap::rbegin\(\)`](#)

Returns an iterator pointing to the last element of the container.

[`multimap::size\(\)`](#)

Returns the number of elements in the multimap container.

[`multimap::emplace\(\)`](#)

Inserts the key and its element in the multimap container.

[`multimap::begin\(\)` and `multimap::end\(\)`](#)

`begin()` returns an iterator referring to the first element in the multimap container. `end()` returns an iterator to the theoretical element that follows the last element in the multimap.

[`multimap::upper\_bound\(\)`](#)

Returns an iterator to the first element that is equivalent to multimapped value with key-value 'g' or definitely will go after the element with key-value 'g' in the multimap.

[`multimap::count\(\)`](#)

Returns the number of matches to element with key-value 'g' in the multimap.

## Start Your Coding Journey Now!

## Function

## Definition

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)[`multimap.find\(\)`](#)

Returns an iterator to the element with key-value 'g' in the multimap if found, else returns the iterator to end.

[`multimap.equal\_range\(\)`](#)

Returns an iterator of pairs. The pair refers to the bounds of a range that includes all the elements in the container which have a key equivalent to k.

[`multimap.insert\(\)`](#)

Used to insert elements in the multimap container.

[`multimap.lower\_bound\(\)`](#)

Returns an iterator to the first element that is equivalent to multimapped value with key-value 'g' or definitely will not go before the element with key-value 'g' in the multimap.

[`multimap.key\_comp\(\)`](#)

Returns the object that determines how the elements in the multimap are ordered ('<' by default).

### [Recent Articles on Multimap](#)

See a successful future with C++

## A COMPLETE BEGINNER TO ADVANCED C++ COURSE

25+  
Hours Of Content

150+  
Curated Problems

24x7  
Doubt Support

>  
Explore Course



Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

## Start Your Coding Journey Now!

[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)**Like** 69[Previous](#)[Next](#)

## Related Articles

1. [multimap::cbegin\(\) and multimap::cend\(\) in C++ STL](#)
2. [multimap::crbegin\(\) and multimap::crend\(\) in C++ STL](#)
3. [multimap::begin\(\) and multimap::end\(\) in C++ STL](#)
4. [List in C++ Standard Template Library \(STL\)](#)
5. [Multiset in C++ Standard Template Library \(STL\)](#)
6. [Set in C++ Standard Template Library \(STL\)](#)
7. [The C++ Standard Template Library \(STL\)](#)
8. [Binary Search in C++ Standard Template Library \(STL\)](#)
9. [Map in C++ Standard Template Library \(STL\)](#)
10. [Queue in C++ Standard Template Library \(STL\)](#)

**Start Your Coding Journey Now!**



[Read](#)[Discuss](#)[Courses](#)[Practice](#)[Video](#)**GeeksforGeeks**

## Vote for difficulty

Current difficulty : [Easy](#)

[Easy](#)[Normal](#)[Medium](#)[Hard](#)[Expert](#)

**Improved By :** [smodi2007](#), [atharvakango](#), [dlbm1302](#), [c0d3rpr0](#), [anshikajain26](#)

**Article Tags :** [cpp-containers-library](#), [cpp-multimap](#), [cpp-multimap-functions](#), [STL](#), [C++](#)

**Practice Tags :** [CPP](#), [STL](#)

[Improve Article](#)[Report Issue](#)**GeeksforGeeks**

A-143, 9th Floor, Sovereign Corporate Tower,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

## Company

[About Us](#)[Careers](#)

## Learn

[DSA](#)[Algorithms](#)

# Start Your Coding Journey Now!

Contact Us

SDE Cheat Sheet

Read

Discuss

Courses

Practice

Video

Advertise with us

Video Tutorials

Courses

News

Top News

Technology

Work & Career

Business

Finance

Lifestyle

Knowledge

Languages

Python

Java

CPP

Golang

C#

SQL

Kotlin

Web Development

Web Tutorials

Django Tutorial

HTML

JavaScript

Bootstrap

ReactJS

NodeJS

Contribute

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship

@geeksforgeeks , Some rights reserved

Start Your Coding Journey Now!