

thispointer.com

[C++11 Tutorials](#)[Python](#)[C++ Tutorials](#)[STL](#)[Multithreading](#)[Data Science](#)[Boost Library](#) ▾[Design Patterns](#)[java](#)[GDB](#)[Datastructure](#) ▾[Subscribe](#)[Home](#) » [C++ 11](#) » [unordered_map](#) » You are reading »

How to Erase / Remove an element from an unordered_map

👤 Varun ⌚ August 4, 2016 📄 C++ 11, unordered_map 💬 No Comment

In this article we will discuss the different ways to remove an element from an unordered_map.

Unordered_map provides different overloaded version of erase() member function to delete key value pairs from map.

Erasing an element from an unordered_map by key

Unordered_map provides following overloaded version of erase() that accepts a key by value and check if any element with given key exists and if yes then deletes that element.

```
1 | size_type erase ( const key_type& k );
```

It returns the number of elements deleted. But as unordered_map can contain only unique elements, so it will return either 0 or 1.

Lets see an example,

```
1 | #include <iostream>
2 | #include <unordered_map>
3 | #include <string>
```

C++11 – Unordered_Map

[Basic Usage Detail and Example](#)[Initializing an unordered_map](#)[Searching in unordered_map](#)[Insert elements in unordered_map](#)[Erasing an element](#)[Erase elements while iterating](#)[std::map vs](#)[std::unordered_map](#)[Std::Tuple](#)[Lambda Functions](#)[C++11 Utilities](#)[Initializer_list](#)[Std::Array](#)[Rvalue References](#)[Smart Pointers](#)[Multithreading](#)[Unordered_Set](#)[Unordered_Map](#)[c++11 : std::tuple Tutorial](#)

```
4
5 int main() {
6 // Initialize an unordered_map through initializer_list
7     std::unordered_map<std::string, int> wordMap( { { "First", 1
8         2 }, { "Third", 3 }, { "Fourth", 4 }, { "Fifth", 5 }
9
10 //Erase element by key
11     if (wordMap.erase("Second") == 1)
12         std::cout << "Element Deleted" << std::endl;
13
14     return 0;
15 }
```

[c++11 : std::make_tuple
Tutorial](#)

Advertisements

Output

```
1 Element Deleted
```

Erasing an element from an unordered_map through an iterator

Unordered_map also provides an overloaded version of erase() that accepts an iterator and deletes it i.e.

Advertisements

```
1 iterator erase ( const_iterator pos );
```

It accepts an iterator and deletes the corresponding element. Also, it returns the iterator pointing to an element i.e. next to last deleted element.

Let's see an example, where it searches for an element with key "Fourth" and then deletes that using iterator,

```
1 #include <iostream>
2 #include <unordered_map>
3 #include <string>
4
5 int main() {
6 // Initialize an unordered_map through initializer_list
7     std::unordered_map<std::string, int> wordMap( { { "First", 1
8         2 }, { "Third", 3 }, { "Fourth", 4 }, { "Fifth", 5 }
9
10 // Iterator pointing to first element of unordered_map
11     std::unordered_map<std::string, int>::iterator it = wordMap.begin();
12
13 // Erase the element pointed by iterator it
14     if (it != wordMap.end())
15         wordMap.erase(it);
16
17 // Display the map contents
18     for (std::pair<std::string, int> element : wordMap)
19         std::cout << element.first << " :: " << element.second << "\n";
20
21     return 0;
22 }
```

Ouput:

```
1 Fifth :: 5
2 First :: 1
3 Second :: 2
4 Third :: 3
```

In the above output entry with key "Fourth" is not present because that's deleted.

Erasing an element by value from an unordered_map

Unordered_map store values in a key value pair and erase() function of unordered_map can take either iterator or key.

So, to delete an element by value we need to, iterate over all the elements in map and search for element whose value matches the given value. Then we will use its iterator to delete that element.

Let's see an example, it searches for an element with value 2 and then deletes that element i.e.

```
1 #include <iostream>
2 #include <unordered_map>
3 #include <string>
4
5 int main() {
6     // Initialize an unordered_map through initializer_list
7     std::unordered_map<std::string, int> wordMap( { { "First", 1
8         2 }, { "Third", 3 }, { "Fourth", 4 }, { "Fifth", 5 }
9
10    // Iterator pointing to first element of unordered_map
11    std::unordered_map<std::string, int>::iterator it = wordMap.begin()
12
13    // Search for an element with value 2
14    while(it != wordMap.end())
15    {
16        if(it->second == 2)
17            break;
18        it++;
19    }
20
21    // Erase the element pointed by iterator it
22    if (it != wordMap.end())
23        wordMap.erase(it);
24
25    // Display the map contents
26    for (std::pair<std::string, int> element : wordMap)
27        std::cout << element.first << " :: " << element.second << "\n";
28
29    return 0;
30 }
```

Subscribe For latest Tutorials

★ indicates required

Email Address ★

Subscribe

Advertisements

Output:

```
1 Fifth :: 5
2 First :: 1
3 Fourth :: 4
4 Third :: 3
```

[Click Here to Subscribe for more Articles / Tutorials like this.](#)

Like 466

Share

Related Posts:

- [c++11 unordered_map : erase elements while iterating...](#)
- [How to find an element in unordered_map](#)
- [Different Ways to initialize an unordered_map](#)
- [How to iterate over an unordered_map in C++11](#)
- [Different ways to insert elements in an unordered_map](#)
- [Unordered_map Usage Tutorial and Example](#)
- [C++ map : Erase element by key or Iterator or Range](#)
- [Different ways to Erase / Delete an element from a...](#)
- [map vs unordered_map | When to choose one over another ?](#)
- [C++ Map: Erase by Value or Callback while iterating...](#)
- [How to erase elements from a list in c++ using iterators](#)
- [How to remove elements from a List while Iterating](#)
- [How check if a given key exists in a Map | C++](#)
- [Erase elements from a Set while Iterating in C++...](#)
- [C++ : How to get element by index in List](#)
- [C++ Map : Operator \[\] - Usage Details](#)
- [C++ std::list Tutorial, Example and Usage Details](#)
- [How to copy all Values from a Map to a Vector in C++](#)
- [How to Search an element in unordered_set](#)
- [C++ : Different ways to insert elements in Set](#)

Advertisements

Search

Search ...

Search

- [How to Insert elements in an unordered set in C++11](#)
- [How to Access Element by index in a Set | C++](#)
- [How to search by value in a Map | C++](#)
- [How to Iterate over a map in C++](#)
- [C++ List – Find | Contains : How to search an...](#)

No Comments Yet

Leave a Reply

Your email address will not be published. Required fields are marked *

This
site

Name *

Email *

Website

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment

uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

« [c++11 unordered_map : erase elements while iterating in a loop](#)

[How to Create a Thread in Java using Runnable Interface](#) »

Python : List

Check if an item exists in List

Check if a list contains all the elements of other list

Create a List and initialize with values

How to Iterate over a List

Insert an element at specific index in List

Sort a list of tuples by 2nd Item

Sort a list of strings

Add an element in list | append() vs extend()

Check if all elements in a List are same

Merge / Join two or more lists

Remove Duplicates from a List

Convert a list to string

Remove element from a list by value or Index

Remove multiple elements from list

Python : Dictionary

Creating Dictionaries in Python

Iterating over Dictionaries in Python

Check if a key exists in Dictionary

Get list of all the keys in Dictionary

Get list of all the Values in a Dictionary

Remove multiple keys in Dictionary while Iterating

Remove a key from Dictionary

Add key/value pairs in Dictionary

Find keys by value in Dictionary

C++11 - Utilities

std::bind

auto specifier

Variadic Templates

C++11 - Unordered Set

1.) unordered_set Basic Example

2.) Initializing an unordered_set

3.) Inserting elements in an unordered_set

4.) Searching an element in unordered_set

5.) unordered_set - Custom Hasher & Comparator

6.) Unordered_set & User defined classes

C++11 - UnorderedMap

Basic Usage Detail and Example

Initializing an unordered_map

Searching in unordered_map

Insert elements in unordered_map

Erasing an element

Erase elements while iterating

std::map vs std::unordered_map

C++11 Smart Pointers

shared_ptr<> Tutorial and Examples

shared_ptr and Custom Deletor

shared_ptr vs raw pointer

Create shared_ptr objects carefully

Advertisement**Design Patterns**

Behavioral Design Patterns

Observer Design Pattern
State Design Pattern
Strategy Design Pattern

Structural Design Patterns

Composite Design Pattern
Flyweight Design Pattern

Creational Design Patterns

Factory Method Design Pattern

Pointers

Pointer vs Reference

Allocating 2D Array Dynamically

Callbacks in C++

Function Pointers

Function Objects & Functors

C++ Strings

Find and Replace all occurrences of a string

Find all occurrences of a sub string

Case Insensitive string::find

Convert First Letter of each word to Upper Case

STL - Vector

1.) What's std::vector and why to use it?

2.) Different ways to initialize a vector

3.) How does std::vector works internally

4.) User Defined Objects & std::vector

5.) How to use vector efficiently in C++?

6.) std::vector and Iterator Invalidation

7.) Remove repeated elements from a vector

8.) Fill a vector with random numbers

9.) Hidden cost of std::vector

10.) Adding elements in Vector

STL - Deque

1.) What is std::deque and how deque works internally.

2.) deque vs vector : What to choose ?

STL - List

1.) std::list Internals & Usage Details

2.) List vs Vector

3.) Different ways to Initialize a list

4.) Erase elements using iterators

5.) Remove elements while Iterating

6.) Remove elements based on External Criterion

7.) Get element by index in List

8.) Searching an element in std::list

9.) Different Ways to iterate over a List

Sort a Dictionary by key or Value

Copy a dictionary | Shallow vs Deep Copy

Convert a list to dictionary

Python Strings

Access characters in string by index in Python

Iterate over the characters in string

How to Replace characters in a string ?

Java - Hashmap

What is Hashing and Hash Table?

Associating Multiple values with same Key

Remove elements while Iterating

Update the value of an existing key

Get all keys by a value in HashMap

Java - HashSet

What is Hashing and Hash Table?

Create and add elements in a HashSet

Iterate over a HashSet

Search for an element in HashSet

Merge two HashSets

Initializing HashSet from an Array

Convert a HashSet into an Array

Merge an Array in a HashSet

Java Interview Questions

weak_ptr Tutorial | shared_ptr and Cyclic References

unique_ptr<> Tutorial and Examples

C++11 Multithreading

Part 1: Three Ways to Create Threads

Part 2: Joining and Detaching Threads

Part 3: Passing Arguments to Threads

Part 4 : Sharing Data & Race Conditions

Part 5 : Fixing Race Conditions using mutex

Part 6 : Need of Event Handling

Part 7: Condition Variables

Part 8: std::future and std::promise

Part 9: std::async Tutorial & Example

Part 10: std::packaged_task<> Tutorial

C++11 Rvalue References

lvalue vs rvalue

Is rvalue immutable in C++?

What is rvalue reference in C++11

Move Constructor

Converting a String to Upper & Lower Case

Trim strings in C++

C++ : How to split a string using String and character as Delimiter?

startsWith() Implementation

endsWith() Implementation

Remove Sub Strings from String

C++ Memory Management

Memory Leaks

new and delete operator

delete vs []delete

Out Of Memory Errors

Overload new & delete

Restrict Dynamic Deletion

Placement new operator

Delete 'this' pointer

Polymorphism

Virtual Functions

vTable and vPointer

10.) Sorting a List & custom Comparator

STL - Set

1.) C++ Set basic example and Tutorial

2.) Using std::set with user defined classes

3.) std::set and external Sorting criteria | Comparator

4.) Access Element by index in Set

5.) How to insert elements in Set

6.) How to iterate over a Set

7.) Removing an element from Set

8.) Erase elements while Iterating & Generic erase_if()

STL - Map

1.) std::map Usage Detail with examples

2.) std::map and Comparator

3.) std::map & User defined class objects as keys

4.) Set vs Map

5.) How to Iterate over a map in C++

6.) Map Insert Example

7.) Iterate a map in reverse order

8.) Check if a key exists in a Map

9.) Search by value in a Map

10.) Erase by Key | Iterators

11.) C++ Map : Operator []

12.) Erase by Value or callback

13.) copy all Values from a Map to vector

Method Overriding
Tutorial

Overriding with
Different Return Type

Calling Base class's
overridden method

Preventing Method
Overriding

Need of preventing
Method Overriding

STL Multimap

MultiMap Example and
Tutorial

multimap::equals_range
– Tutorial

STL Algorithms

std::sort Tutorial &
Example

std::unique Tutorial &
Example

1.) Using std::find &
std::find_if with User
Defined Classes

2.) Iterating over a
range of User Defined
objects and calling
member function using
std::for_each

Terms and Conditions

Terms and Conditions
Policy