

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free.

Sign up



## What is a "set" in C++? When are they useful?



I'm having a hard time conceptualizing c++ sets, actually sets in general.

What are they? How are they useful?

c++ set computer-science

edited May 25 at 22:56



Andreas DM

2,587 1 8 31

asked Sep 7 '09 at 22:48



Monte Hurd

1,273 2 15 31

1 [google.com/search?q=c%2B%2B+sets](http://google.com/search?q=c%2B%2B+sets) – Aziz Sep 7 '09 at 22:50

3 yeah thanks for that. i've even read the wikipedia article on sets but all the definitions i've found talk about "abstract data structures" and "finite sets". i guess i was just hoping for a "for dummies" plain english explanation... so i guess i can't really blame you for implying i'm dumb rather than actually helping. – Monte Hurd Sep 7 '09 at 22:59

1 @Monte, I apologize if I offended you in any way. I was just suggesting to search google, as there are lots of articles explaining the concept in question. – Aziz Sep 7 '09 at 23:32

@Aziz no worry! I see you've helpfully answered many questions. I should be less snarky. – Monte Hurd Sep 8 '09 at 1:54

### 8 Answers

Don't feel bad if you have trouble understanding sets in general. Most of a degree in mathematics is spent coming to terms with set theory:

<http://en.wikipedia.org/wiki/Set%5Ftheory>

Think of a set as a collection of unique, unordered objects. In many ways it looks like a list:

{ 1, 2, 3, 4 }

but order is unimportant:

{ 4, 3, 2, 1 } = { 1, 2, 3, 4 }

and repetitions are ignored:

{ 1, 1, 2, 3, 4 } = { 1, 2, 3, 4 }

A C++ set is an implementation of this mathematical object, with the odd feature that it is sorted internally. But this is just a detail of implementation, and is not relevant to understanding the data structure. The sorting is just for speed.

answered Sep 7 '09 at 23:06



David Crawshaw

5,737 3 27 39

Track your code, bugs, and work.  
Anywhere. For free.



GET STARTED →



C++ STL Sets are associative mappings that guarantee both sorting and uniqueness of elements within the set (Multisets guarantee the former but not the latter).

They are typically used as part of set operations - things like unions, intersections, and other interactions involving inclusion/exclusion of elements within a set.

answered Sep 7 '09 at 22:51



Amber

206k 28 348 372

Sets "in general" are a (very fundamental) concept in mathematics.

STL's `set` is based on the mathematical concept of a set: it's a collection of unique members, or a "Unique Associative Container" in STL terminology. The one slightly odd thing is that it sorts elements (in a mathematical set, there is no "order" to the elements).

Some STL implementations also support a `hash_set`, which is very similar to `set`, in that it is also an analog to the mathematical concept of a set. The big differences between `set` and `hash_set` are that `hash_sets` do *not* sort their elements, they have different performance characteristics ( $O(1)$  rather than  $O(\log n)$  look-ups, assuming a good hash function), and of course they aren't standard.

answered Sep 7 '09 at 23:12



Laurence Gonsalves

65k 12 123 185

"Set" is a kind of collection that store multiple but unique objects. It is useful when you want to collect objects but you don't care their order or how many times same object are in it.

See this for more detail: [Set in C++](#)

answered Sep 7 '09 at 22:55



NawaMan

701 3 11

"It is useful when you want to collect objects but you don't care their order or how many times same object are in it." Isn't this kind of exactly when you can't/don't want to use a set? A set is useful when you DO care about order and you DON'T want duplicates. – [DeusAduro](#) Sep 7 '09 at 23:10

1 As [DeusAduro](#) said, a c++ Set stores the elements sorted, and only allows one unique instance in the same. – [Ricardo Amores](#) Sep 7 '09 at 23:15

You are right about ordering! I mix up with Set in Java. Sorry. – [NawaMan](#) Sep 8 '09 at 0:02

What are they?

A set is a collection.

A set is like a dictionary or 'map' of key/value pairs, except that it only stores (is a collection of) keys without associated values.

A set either does or doesn't contain an instance of each possible key value. For example, a set of integers might contain the values {0, 1, 5}. A value (e.g. 5) can't be contained more than once in the set (if you call the set's insert method more than once for a given key value, the set will still contain only one instance of that key value).

How are they useful?

I don't use them nearly as often as maps.

One time I use a set is if I'm a library which gives away pointers which a client uses as a handle. I'll keep a private set which contains all the valid handle values which I've created. When the client gives me a handle, I'll test whether the handle is a valid handle by testing whether that value is contained in my set.

edited Sep 8 '09 at 2:58

answered Sep 7 '09 at 23:34



ChrisW

40.1k 5 62 140

---

The set code that was confusing me is using them exactly as you do for keeping track of handles. –

[Monte Hurd](#) Sep 8 '09 at 2:04

---

#### Citing Wikipedia:

A set is a collection of distinct objects, considered as an object in its own right. Sets are one of the most fundamental concepts in mathematics. Although it was invented at the end of the 19th century, set theory is now a ubiquitous part of mathematics, and can be used as a foundation from which nearly all of mathematics can be derived.

answered Sep 7 '09 at 22:57



Michael Borgwardt

216k 41 312 554

---

STL `set` is a [red-black tree](#) (at least that's how I think it is implemented)

Another way to look at it.

Hence the properties, fast element search, element ordering, element uniqueness, ordered traversal and so on.

It is useful when you want to keep track of unique elements like for example list of unique strings or integers but you can store more complicated structures as well.

answered Sep 7 '09 at 23:28



stefanB

33.7k 19 78 116

---

For an unordered implementation of sets in C++, check out [Boost.Unordered](#). In many cases this is a better choice than STL `set`, which I personally more or less use only to build a sorted list incrementally.

answered Sep 8 '09 at 8:43



larsmoa

6,397 2 27 56