| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/HashSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/HashMap.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Hashtable.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/HashSet.html)    [**NO FRAMES**](http://docs.google.com/HashSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#35nkun2) |

## **java.util**

Class HashSet<E>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<E>  
 [java.util.AbstractSet](http://docs.google.com/java/util/AbstractSet.html)<E>  
 **java.util.HashSet<E>**

**Type Parameters:**E - the type of elements maintained by this set **All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Iterable](http://docs.google.com/java/lang/Iterable.html)<E>, [Collection](http://docs.google.com/java/util/Collection.html)<E>, [Set](http://docs.google.com/java/util/Set.html)<E> **Direct Known Subclasses:** [JobStateReasons](http://docs.google.com/javax/print/attribute/standard/JobStateReasons.html), [LinkedHashSet](http://docs.google.com/java/util/LinkedHashSet.html)

public class **HashSet<E>**extends [AbstractSet](http://docs.google.com/java/util/AbstractSet.html)<E>implements [Set](http://docs.google.com/java/util/Set.html)<E>, [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

This class implements the Set interface, backed by a hash table (actually a HashMap instance). It makes no guarantees as to the iteration order of the set; in particular, it does not guarantee that the order will remain constant over time. This class permits the null element.

This class offers constant time performance for the basic operations (add, remove, contains and size), assuming the hash function disperses the elements properly among the buckets. Iterating over this set requires time proportional to the sum of the HashSet instance's size (the number of elements) plus the "capacity" of the backing HashMap instance (the number of buckets). Thus, it's very important not to set the initial capacity too high (or the load factor too low) if iteration performance is important.

**Note that this implementation is not synchronized.** If multiple threads access a hash set concurrently, and at least one of the threads modifies the set, it *must* be synchronized externally. This is typically accomplished by synchronizing on some object that naturally encapsulates the set. If no such object exists, the set should be "wrapped" using the [Collections.synchronizedSet](http://docs.google.com/java/util/Collections.html#synchronizedSet(java.util.Set)) method. This is best done at creation time, to prevent accidental unsynchronized access to the set:

Set s = Collections.synchronizedSet(new HashSet(...));

The iterators returned by this class's iterator method are *fail-fast*: if the set is modified at any time after the iterator is created, in any way except through the iterator's own remove method, the Iterator throws a [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html). Thus, in the face of concurrent modification, the iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Note that the fail-fast behavior of an iterator cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast iterators throw ConcurrentModificationException on a best-effort basis. Therefore, it would be wrong to write a program that depended on this exception for its correctness: *the fail-fast behavior of iterators should be used only to detect bugs.*

This class is a member of the  [Java Collections Framework](http://docs.google.com/technotes/guides/collections/index.html).

**Since:** 1.2 **See Also:**[Collection](http://docs.google.com/java/util/Collection.html), [Set](http://docs.google.com/java/util/Set.html), [TreeSet](http://docs.google.com/java/util/TreeSet.html), [HashMap](http://docs.google.com/java/util/HashMap.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.util.HashSet)

| **Constructor Summary** | |
| --- | --- |
| [**HashSet**](http://docs.google.com/java/util/HashSet.html#HashSet())()            Constructs a new, empty set; the backing HashMap instance has default initial capacity (16) and load factor (0.75). |
| [**HashSet**](http://docs.google.com/java/util/HashSet.html#HashSet(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/HashSet.html)> c)            Constructs a new set containing the elements in the specified collection. |
| [**HashSet**](http://docs.google.com/java/util/HashSet.html#HashSet(int))(int initialCapacity)            Constructs a new, empty set; the backing HashMap instance has the specified initial capacity and default load factor (0.75). |
| [**HashSet**](http://docs.google.com/java/util/HashSet.html#HashSet(int,%20float))(int initialCapacity, float loadFactor)            Constructs a new, empty set; the backing HashMap instance has the specified initial capacity and the specified load factor. |

| **Method Summary** | |
| --- | --- |
| boolean | [**add**](http://docs.google.com/java/util/HashSet.html#add(E))([E](http://docs.google.com/java/util/HashSet.html) e)            Adds the specified element to this set if it is not already present. |
| void | [**clear**](http://docs.google.com/java/util/HashSet.html#clear())()            Removes all of the elements from this set. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/util/HashSet.html#clone())()            Returns a shallow copy of this HashSet instance: the elements themselves are not cloned. |
| boolean | [**contains**](http://docs.google.com/java/util/HashSet.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Returns true if this set contains the specified element. |
| boolean | [**isEmpty**](http://docs.google.com/java/util/HashSet.html#isEmpty())()            Returns true if this set contains no elements. |
| [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/HashSet.html)> | [**iterator**](http://docs.google.com/java/util/HashSet.html#iterator())()            Returns an iterator over the elements in this set. |
| boolean | [**remove**](http://docs.google.com/java/util/HashSet.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Removes the specified element from this set if it is present. |
| int | [**size**](http://docs.google.com/java/util/HashSet.html#size())()            Returns the number of elements in this set (its cardinality). |

| **Methods inherited from class java.util.**[**AbstractSet**](http://docs.google.com/java/util/AbstractSet.html) |
| --- |
| [equals](http://docs.google.com/java/util/AbstractSet.html#equals(java.lang.Object)), [hashCode](http://docs.google.com/java/util/AbstractSet.html#hashCode()), [removeAll](http://docs.google.com/java/util/AbstractSet.html#removeAll(java.util.Collection)) |

| **Methods inherited from class java.util.**[**AbstractCollection**](http://docs.google.com/java/util/AbstractCollection.html) |
| --- |
| [addAll](http://docs.google.com/java/util/AbstractCollection.html#addAll(java.util.Collection)), [containsAll](http://docs.google.com/java/util/AbstractCollection.html#containsAll(java.util.Collection)), [retainAll](http://docs.google.com/java/util/AbstractCollection.html#retainAll(java.util.Collection)), [toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray()), [toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray(T%5B%5D)), [toString](http://docs.google.com/java/util/AbstractCollection.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Methods inherited from interface java.util.**[**Set**](http://docs.google.com/java/util/Set.html) |
| --- |
| [addAll](http://docs.google.com/java/util/Set.html#addAll(java.util.Collection)), [containsAll](http://docs.google.com/java/util/Set.html#containsAll(java.util.Collection)), [equals](http://docs.google.com/java/util/Set.html#equals(java.lang.Object)), [hashCode](http://docs.google.com/java/util/Set.html#hashCode()), [removeAll](http://docs.google.com/java/util/Set.html#removeAll(java.util.Collection)), [retainAll](http://docs.google.com/java/util/Set.html#retainAll(java.util.Collection)), [toArray](http://docs.google.com/java/util/Set.html#toArray()), [toArray](http://docs.google.com/java/util/Set.html#toArray(T%5B%5D)) |

| **Constructor Detail** |
| --- |

### HashSet

public **HashSet**()

Constructs a new, empty set; the backing HashMap instance has default initial capacity (16) and load factor (0.75).

### HashSet

public **HashSet**([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/HashSet.html)> c)

Constructs a new set containing the elements in the specified collection. The HashMap is created with default load factor (0.75) and an initial capacity sufficient to contain the elements in the specified collection.

**Parameters:**c - the collection whose elements are to be placed into this set **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified collection is null

### HashSet

public **HashSet**(int initialCapacity,  
 float loadFactor)

Constructs a new, empty set; the backing HashMap instance has the specified initial capacity and the specified load factor.

**Parameters:**initialCapacity - the initial capacity of the hash maploadFactor - the load factor of the hash map **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is less than zero, or if the load factor is nonpositive

### HashSet

public **HashSet**(int initialCapacity)

Constructs a new, empty set; the backing HashMap instance has the specified initial capacity and default load factor (0.75).

**Parameters:**initialCapacity - the initial capacity of the hash table **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is less than zero

| **Method Detail** |
| --- |

### iterator

public [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/HashSet.html)> **iterator**()

Returns an iterator over the elements in this set. The elements are returned in no particular order.

**Specified by:**[iterator](http://docs.google.com/java/lang/Iterable.html#iterator()) in interface [Iterable](http://docs.google.com/java/lang/Iterable.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[iterator](http://docs.google.com/java/util/Collection.html#iterator()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[iterator](http://docs.google.com/java/util/Set.html#iterator()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[iterator](http://docs.google.com/java/util/AbstractCollection.html#iterator()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/HashSet.html)> **Returns:**an Iterator over the elements in this set**See Also:**[ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html)

### size

public int **size**()

Returns the number of elements in this set (its cardinality).

**Specified by:**[size](http://docs.google.com/java/util/Collection.html#size()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[size](http://docs.google.com/java/util/Set.html#size()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[size](http://docs.google.com/java/util/AbstractCollection.html#size()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/HashSet.html)> **Returns:**the number of elements in this set (its cardinality)

### isEmpty

public boolean **isEmpty**()

Returns true if this set contains no elements.

**Specified by:**[isEmpty](http://docs.google.com/java/util/Collection.html#isEmpty()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[isEmpty](http://docs.google.com/java/util/Set.html#isEmpty()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Overrides:**[isEmpty](http://docs.google.com/java/util/AbstractCollection.html#isEmpty()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/HashSet.html)> **Returns:**true if this set contains no elements

### contains

public boolean **contains**([Object](http://docs.google.com/java/lang/Object.html) o)

Returns true if this set contains the specified element. More formally, returns true if and only if this set contains an element e such that (o==null ? e==null : o.equals(e)).

**Specified by:**[contains](http://docs.google.com/java/util/Collection.html#contains(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[contains](http://docs.google.com/java/util/Set.html#contains(java.lang.Object)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Overrides:**[contains](http://docs.google.com/java/util/AbstractCollection.html#contains(java.lang.Object)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/HashSet.html)> **Parameters:**o - element whose presence in this set is to be tested **Returns:**true if this set contains the specified element

### add

public boolean **add**([E](http://docs.google.com/java/util/HashSet.html) e)

Adds the specified element to this set if it is not already present. More formally, adds the specified element e to this set if this set contains no element e2 such that (e==null ? e2==null : e.equals(e2)). If this set already contains the element, the call leaves the set unchanged and returns false.

**Specified by:**[add](http://docs.google.com/java/util/Collection.html#add(E)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[add](http://docs.google.com/java/util/Set.html#add(E)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Overrides:**[add](http://docs.google.com/java/util/AbstractCollection.html#add(E)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/HashSet.html)> **Parameters:**e - element to be added to this set **Returns:**true if this set did not already contain the specified element

### remove

public boolean **remove**([Object](http://docs.google.com/java/lang/Object.html) o)

Removes the specified element from this set if it is present. More formally, removes an element e such that (o==null ? e==null : o.equals(e)), if this set contains such an element. Returns true if this set contained the element (or equivalently, if this set changed as a result of the call). (This set will not contain the element once the call returns.)

**Specified by:**[remove](http://docs.google.com/java/util/Collection.html#remove(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[remove](http://docs.google.com/java/util/Set.html#remove(java.lang.Object)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Overrides:**[remove](http://docs.google.com/java/util/AbstractCollection.html#remove(java.lang.Object)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/HashSet.html)> **Parameters:**o - object to be removed from this set, if present **Returns:**true if the set contained the specified element

### clear

public void **clear**()

Removes all of the elements from this set. The set will be empty after this call returns.

**Specified by:**[clear](http://docs.google.com/java/util/Collection.html#clear()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Specified by:**[clear](http://docs.google.com/java/util/Set.html#clear()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/HashSet.html)>**Overrides:**[clear](http://docs.google.com/java/util/AbstractCollection.html#clear()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/HashSet.html)>

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()

Returns a shallow copy of this HashSet instance: the elements themselves are not cloned.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a shallow copy of this set**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/HashSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/HashMap.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Hashtable.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/HashSet.html)    [**NO FRAMES**](http://docs.google.com/HashSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#35nkun2) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).