

JAVA - THE HASHSET CLASS

HashSet extends AbstractSet and implements the Set interface. It creates a collection that uses a hash table for storage.

A hash table stores information by using a mechanism called hashing. In hashing, the informational content of a key is used to determine a unique value, called its hash code.

The hash code is then used as the index at which the data associated with the key is stored. The transformation of the key into its hash code is performed automatically.

The HashSet class supports four constructors. The first form constructs a default hash set:

```
HashSet ( )
```

The following constructor form initializes the hash set by using the elements of c.

```
HashSet (Collection c)
```

The following constructor form initializes the capacity of the hash set to capacity.

The capacity grows automatically as elements are added to the Hash.

```
HashSet (int capacity)
```

The fourth form initializes both the capacity and the fill ratio (also called load capacity) of the hash set from its arguments:

```
HashSet (int capacity, float fillRatio)
```

Here the fill ratio must be between 0.0 and 1.0, and it determines how full the hash set can be before it is resized upward. Specifically, when the number of elements is greater than the capacity of the hash set multiplied by its fill ratio, the hash set is expanded.

Apart from the methods inherited from its parent classes, HashSet defines following methods:

SN	Methods with Description
1	boolean add(Object o) Adds the specified element to this set if it is not already present.
2	void clear() Removes all of the elements from this set.
3	Object clone() Returns a shallow copy of this HashSet instance: the elements themselves are not cloned.
4	boolean contains(Object o) Returns true if this set contains the specified element
5	boolean isEmpty() Returns true if this set contains no elements.

6 **Iterator iterator()**

Returns an iterator over the elements in this set.

7 **boolean remove(Object o)**

Removes the specified element from this set if it is present.

8 **int size()**

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

Example:

The following program illustrates several of the methods supported by HashSet:

```
import java.util.*;
public class HashSetDemo {

    public static void main(String args[]) {
        // create a hash set
        HashSet hs = new HashSet();
        // add elements to the hash set
        hs.add("B");
        hs.add("A");
        hs.add("D");
        hs.add("E");
        hs.add("C");
        hs.add("F");
        System.out.println(hs);
    }
}
```

This would produce the following result:

```
[D, E, F, A, B, C]
```