

# JAVA - THE LINKEDHASHSET CLASS

[http://www.tutorialspoint.com/java/java\\_linkedhashset\\_class.htm](http://www.tutorialspoint.com/java/java_linkedhashset_class.htm)

Copyright © tutorialspoint.com

This class extends HashSet, but adds no members of its own.

LinkedHashSet maintains a linked list of the entries in the set, in the order in which they were inserted. This allows insertion-order iteration over the set.

That is, when cycling through a LinkedHashSet using an iterator, the elements will be returned in the order in which they were inserted.

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

```
LinkedHashSet ( )
```

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

```
LinkedHashSet (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.

```
LinkedHashSet (int capacity)
```

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

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

## Example:

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

```
import java.util.*;

public class HashSetDemo {

    public static void main(String args[]) {
        // create a hash set
        LinkedHashSet hs = new LinkedHashSet ();
        // 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:

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