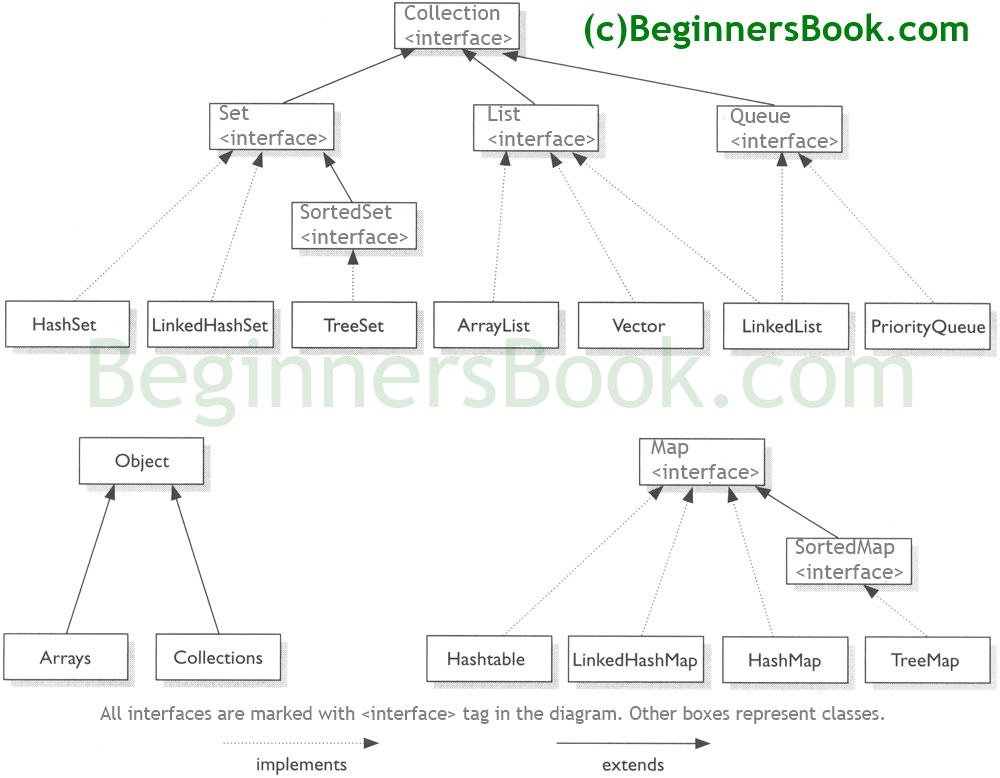
**Java Collections Framework Tutorials**

The [**Java Collections Framework**](http://docs.oracle.com/javase/tutorial/collections/intro/index.html) is a collection of interfaces and classes which helps in storing and processing the data efficiently. This framework has several useful classes which have tons of useful functions which makes a programmer task super easy. I have written several tutorials on Collections and below are the links of those. All the tutorials are shared with examples and source codes to help you understand better.

**Collections Framework hierarchy**



**List**

A List is an ordered Collection (sometimes called a sequence). **Lists may contain duplicate elements**. Elements can be inserted or accessed by their position in the list, using a zero-based index.

* [**ArrayList**](http://beginnersbook.com/2014/08/arraylist-in-java/)
* [**LinkedList**](http://beginnersbook.com/2014/08/java-linkedlist-class/)
* [**Vector**](http://beginnersbook.com/2014/08/java-vector-class/)

**Set**

A Set is a **Collection that cannot contain duplicate elements**. There are three main implementations of Set interface: HashSet, TreeSet, and LinkedHashSet. HashSet, which stores its elements in a hash table, is the best-performing implementation; however it makes no guarantees concerning the order of iteration. TreeSet, which stores its elements in a red-black tree, orders its elements based on their values; it is substantially slower than HashSet. LinkedHashSet, which is implemented as a hash table with a linked list running through it, orders its elements based on the order in which they were inserted into the set (insertion-order).

* [**HashSet**](http://beginnersbook.com/2013/12/hashset-class-in-java-with-example/)
* [**LinkedHashSet**](http://beginnersbook.com/2013/12/linkedhashset-class-in-java-with-example/)
* [**TreeSet**](http://beginnersbook.com/2013/12/treeset-class-in-java-with-example/)

**Map**

**A Map is an object that maps keys to values**. A map cannot contain duplicate keys. There are three main implementations of Map interfaces: HashMap, TreeMap, and LinkedHashMap.  
HashMap: It makes no guarantees concerning the order of iteration  
TreeMap: It stores its elements in a red-black tree, orders its elements based on their values; it is substantially slower than HashMap.  
LinkedHashMap: It orders its elements based on the order in which they were inserted into the set (insertion-order).

* [**HashMap**](http://beginnersbook.com/2014/08/java-hashmap-class/)
* [**TreeMap**](http://beginnersbook.com/2013/12/treemap-in-java-with-example/)
* [**LinkedHashMap**](http://beginnersbook.com/2013/12/linkedhashmap-in-java/)

**Iterator/ListIterator**

Both Iterator and ListIterator are used to iterate through elements of a collection class. Using Iterator we can traverse in one direction (forward) while using ListIterator we can traverse the collection class on both the directions(backward and forward). To know more differences between these two refer this article: [**Difference between Iterator and ListIterator**](http://beginnersbook.com/2014/06/difference-between-iterator-and-listiterator-in-java/).

* [**Iterator**](http://beginnersbook.com/2014/06/java-iterator-with-examples/)
* [**ListIterator**](http://beginnersbook.com/2014/06/listiterator-in-java-with-examples/)