Original article: http://www.codejava.net/java-core/collections/java-collections-framework-summary-table

The major characteristics of the top level interfaces in the Java Collections Framework, under the java.util package, are summarized in the below table.

	Collection	**List**	**Queue**	**Deque**	**Set**	**Map**
Description	A group of objects Parent of the collection hierarchy	A sequence of objects Position aware (zero-based)	A group of objects waiting to be processed	A group of objects waiting to be processed	An abstraction of the mathematical set	A key-value pairing of objects Do not inherit Collection interface
Allow duplicates	Implementation dependent	Yes	Yes	Yes	No	Keys cannot duplicate
Allow nulls	Implementation dependent	Yes	No	No	At most one	At most one null key Values can be null
Insertion order	Implementation dependent	Append at end or by specific position (index)	First-in, first out (FIFO)	First-in, first-out (FIFO) like a queue or last-in, first-out (LIFO) like a stack	Implementation dependent	Implementation dependent
Sub-interfaces	Set SortedSet NavigableSet List Queue Deque		Deque		SortedSet NavigableSet	SortedMap NavigableMap
Common implementations		ArrayList LinkedList	PriorityQueue LinkedList ArrayDeque	ArrayDeque LinkedList	HashSet LinkedHashSet TreeSet	HashMap LinkedHashMap TreeMap

All interfaces, except Collection, share one thing in common. All concrete implementations are resizable, meaning they have an initial capacity that can hold x elements before resizing.

Original article: http://www.codejava.net/java-core/collections/java-collections-framework-summary-table