

Collection	Insertion Order	Synchronization	Duplicate Elements	Allowing of Null
ArrayList<T>	Maintained. Inserts at the end of list or specified index	unsynchronized	Yes	Yes
LinkedList<T>	Maintained. Insert at end, beginning, or specified index	unsynchronized	Yes	Yes
HashSet<E>	Unordered	unsynchronized	No	Yes only 1
LinkedHashSet <E>	Maintained	unsynchronized	No	Yes only 1
Tree Set<E>	Ordered by natural or comparator ordering	unsynchronized	No	No
HashMap<K ,V>	Unordered	unsynchronized	Yes but not duplicate keys	Yes
LinkedHashMap <K, V>	ordered	unsynchronized	Yes but not duplicate keys	Yes
Tree Map<K, V>	Ordered by natural or comparator ordering	unsynchronized	Yes but not duplicate keys	Null elements but not keys

1. We can't sort a Hashmap directly using comparable or comparator because hashmap is a key, value pair <k, v> organized as a hash table and comparable and Comparator use a type parameter <T>. To sort a map, use tree map which uses comparator or comparable.
2. The remove method in iterator is to safely remove elements in a collection while iterating through it. If you do that without iterator, it will throw a concurrent modification exception.
3. A HashSet is backed by a hash table (hashmap instance) which allows add, remove, contains, and size operations to be constant time.