Set

1. No duplicates element allowed
2. Can contain 1 null element
3. Methods –
4. add(E e)
5. addAll(Collection c)
6. clear()
7. contains()
8. containsAll(Collection c)
9. equals(Object o)
10. hashCode()
11. isEmpty()
12. remove(Object o)
13. removeAll(Collection c)
14. retainAll(Collection c)
15. size()
16. toArray()

* HashSet –

1. Can have 1 null element
2. Not synchronized
3. Order is not preserved
4. To make it synchronized Collections.synchronizedSet(new hashSet(….)) is used
5. InitialCapacity=16, LoadFactor=0.75
6. Constructors
7. HashSet()
8. HashSet(Collection c)
9. HashSet(int initialCapacity)
10. HashSet(int initialCapacity, float loadFactor)
11. All set methods are inherited here

* LinkedHashSet

1. Preserves insertion order
2. It maintains a doubly-linked list running through all of its entries
3. Non synchronized
4. InitialCapacity=16, LoadFactor=0.75
5. Constructors
6. LinkedHashSet()
7. LinkedHashSet(Collection c)
8. LinkedHashSet(int initialCapacity)
9. LinkedHashSet(int initialCapacity, float loadFactor)

* SortedSet

1. It is an Interface implemented by TreeSet
2. Provides a total ordering of the set
3. By default natural ordering is applied
4. To apply any other ordering we can pass comparator object in the constructor
5. Methods
6. comparator()
7. first()
8. headSet(Element toElement)
9. tailSet(Element fromElement)
10. last()
11. subSet(fromElement, toElement)

* TreeSet

1. The elements are ordered using their [natural ordering](https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html), or by a [Comparator](https://docs.oracle.com/javase/8/docs/api/java/util/Comparator.html) provided at set creation time, depending on which constructor is used.
2. Not synchronized
3. Has implemented the SortedSet interface
4. Constructors
5. TreeSet()
6. TreeSet(Collection c)
7. TreeSet(Comparator c)
8. Methods
9. ceiling()- returns the above object
10. floor()- returns the below object
11. descendingSet()
12. headSet(E toElement, boolean inclusive)
13. tailSet(E fromElement, boolean inclusive)
14. pollFirst()
15. pollLast()