Understanding Set Interface

- It is the child interface of Collection.
- A Set is a Collection that cannot contain duplicate elements
- The Set interface contains only methods inherited from Collection and adds the restriction that duplicate elements are prohibited.

HashSet:

- HashSet is available since jdk1.2V
- Underlying data structure for HashSet is HashTable.
- It doesn't allows duplicates & insertion order is not maintained.
- Default capacity when creating an HashSet is 16.
- Load Factor for HashSet is 0.75 **(No of elements/ Size of the hashTable)**
- HashSet is not synchronized by default.
- Accepts 'null' value for only once.

```
HashSet hs=new HashSet();
HashSet hs=new HashSet(int initialcapacity);
```

LinkedHashSet:

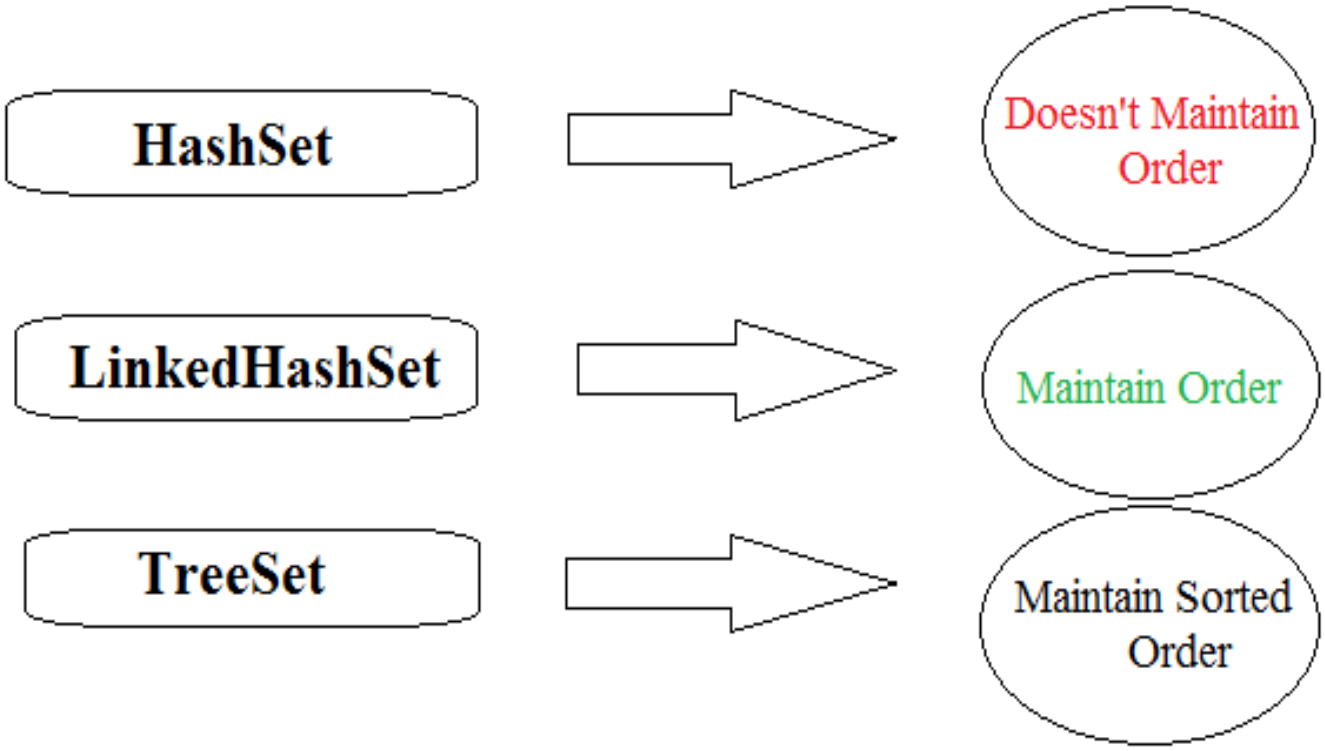
- The only difference between HashSet & LinkedHashSet is Hash set doesn't maintain the insertion order where as LinkedHashSet maintains it.
- LinkedHashSet is available since jdk1.4V.
- It inherits HashSet class and implements Set interface.

```
LinkedHashSet lhs=new LinkedHashSet();
```

TreeSet:

- TreeSet maintains sorting order of inserted elements.
- TreeSet is available since jdk1.2V.
- It will arrange the elements in ascending order using balanced binary search tree algorithm.
- TreeSet will not allow to insert Heterogeneous objects
- It doesn’t allows duplicates & insertion order is not maintained.

```
TreeSet t=new TreeSet();
```



All the three class doesn't accept duplicates elements.

TreeSet methods

descendingSet();	Returns a reverse order view of the elements contained in this set
descendingIterator();	Returns an iterator over the elements in this set in descending order.
headSet(E toElement);	Returns the elements less than to the specified element
tailSet(E toElement);	Returns the elements greater than or equal to the specified element