Set

Set is an interface which is implemented by three classes
HashSet
LinkedHashSet
TreeSet

When to choose Set interface?
If we don't want to store duplicate objects.
If insertion order is not required.
Set (I) extends Collection(I) and belongs to java.util package.

1. HashSet :-
   - It is a class implements Set Interface.
   - Introduced in 1.2 version.
   - It will accept both homogeneous and heterogeneous values.
   - Duplicates are not allowed
   - Only one null object is allowed.
   - DataStructure is HashTable.
   - It will not follow insertion order.
   - Default capacity is 16
   - It has 4 constructors

ex:-

```
public class CollFramWork {
	public static void main(String[] args) {
		Set s=new HashSet();
		s.add(25);
		s.add("e");
		s.add(6);
		s.add(null);
		s.add('a');
		s.add(null);
		System.out.println(s); //[null, a, e, 6, 25]
		s.remove(null);
		System.out.println(s); //[a, e, 6, 25]
		Iterator i=s.iterator();
		while(i.hasNext()) {
			System.out.println(i.next());
		}
	}
}
```

2. LinkedHashet :-

LinkedHashet extends to HashSet and implements Set interface.

Characteristics :-

Introduces in 1.4 version
Accepts both homogeneous and heterogeneous objects.
Duplicates are not allowed.
Only one null object is accepted.
It follows HashTable DataStructure to store the objects and LinkedList to maintain insertion order.
Insertion order is preserved.
It has 4 constructors.
Default capacity is 16

ex:-
```
public class CollFrameWork {
	public static void main(String[] args) {
		LinkedHashSet hs=new LinkedHashSet();
		hs.add("a");
		hs.add("a");
		hs.add(1);
		hs.add(null);
		hs.add('a');
		System.out.println(hs); //[a, 1, null, a]
```

ex2:-
```
public class CollFrameWork {
	public static void main(String[] args) {
		LinkedHashSet hs=new LinkedHashSet();
		hs.add("a");
		hs.add("a");
		hs.add(1);
		hs.add(null);
		hs.add('a');
		System.out.println(hs); //[a, 1, null, a]
		ArrayList a= new ArrayList(hs);
		a.add("a");
		a.add("a");
		System.out.println(a); //[a, 1, null, a, a, a]
		Iterator i=a.iterator();
		while(i.hasNext()) {
			System.out.println(i.next());
		}
	}
}
```

3. TreeSet
   TreeSet implements Set Interface

Characteristics:-

Introduced in1.2v
Heterogeneous objects are not allowed.
Duplicates are not allowed
Nullobject is not allowed
It follows TreeMap data Structure
Output will be in sorted order.
It has 4 constructors.
Default capacity is 16.

ex:-
```
public class CollFrameWork {
	public static void main(String[] args) {
		TreeSet hs=new TreeSet();
		hs.add(8);
		hs.add(10);
		hs.add(2);
		hs.add(6);
		System.out.println(hs); //[2, 6, 8, 10]
		Iterator i=hs.iterator();
		while(i.hasNext()) {
			System.out.println(i.next());
		}
		System.out.println("Reverse order");
		Iterator i1=hs.descendingIterator();
		while(i1.hasNext()) {
			System.out.println(i1.next());
		}
	}
}
```

ex2:-
```
public class CollFrameWork {
	public static void main(String[] args) {
		TreeSet hs=new TreeSet();
		hs.add(8);
		hs.add(10);
		hs.add(2);
		hs.add(6);
		System.out.println(hs); //[2, 6, 8, 10]
		System.out.println(hs.first()); //2
		System.out.println(hs.last()); //10
		System.out.println(hs); //[2, 6, 8, 10]
		System.out.println("----------");
		System.out.println(hs.pollFirst()); //2
		System.out.println(hs.pollLast()); //10
```

```
                System.out.println(hs); //[6, 8]
        }
}
```

Note:-
In Treeset output will defaultly in sorted order
It allows only homogenous objects i.e it allows only comparable type of objects, if we add heterogeneous objects it gives ClassCastException.
If we add null objects it gives NullPointerException.
first()→ provides first object
last()→ provides last object
pollFirst()→provides first object and delete it from tree
pollLast()→ provides last object and delete it from tree

ex :- printing TreeSet in descending order using Comparator

```
public class CollFrWork  implements Comparator{

        @Override
        public int compare(Object o1, Object o2) {
                String cf = (String)o1;
                String cf1 = (String)o2;
                if(cf.compareTo(cf1)>0) {
                        return 1;
                }
                else if(cf.compareTo(cf1)<0) {
                        return -1;
                }
                return 0;
        }

        public static void main(String[] args) {
                TreeSet ht=new TreeSet(new CollFrWork());
                ht.add("sanjaya");
                ht.add("abhimanyu");
                ht.add("krishna");
                ht.add("arjuna");
                ht.add("gandhari");
                ht.add("droupadhi");
                ht.add("hidimbi");
                System.out.println(ht);
        }
}
```