

Employee e2 = new Employee ("balalah", 200);  
 Employee e3 = new Employee ("chiru", 50);  
 Employee e4 = new Employee ("venki", 150);  
 Employee e5 = new Employee ("nag", 100);

```
TreeSet t = new TreeSet();
t.add(e1); t.add(e2); t.add(e3); t.add(e4); t.add(e5);
s.op(t);
```

```
TreeSet t1 = new TreeSet(new MyComparator());
t1.add(e1); t1.add(e2); t1.add(e3); t1.add(e4); t1.add(e5);
s.op(t1);
```

```
} }
class Employee implements Comparable
```

```
{
    String name;
    int eid;
    Employee(String name, int eid)
    {
        this.name = name;
        this.eid = eid;
    }
}
```

```
public String toString()
{
    return name + "--" + eid;
}
```

```
public int compareTo(Object obj)
{
    int eid1 = this.eid;
    Employee e = (Employee) obj;
    int eid2 = e.eid;
    if (eid1 < eid2)
    {
        return -1;
    }
    else if (eid1 > eid2)
    {
        return 1;
    }
    else {
        return 0;
    }
}
```

```
class MyComparator
implements Comparator
{
    public int compare
    (Object obj1, Object obj2)
    {
        Employee e1 = (Employee)
            obj1;
        Employee e2 = (Employee)
            obj2;
        String s1 = e1.name;
        String s2 = e2.name;
        return
            s1.compareTo(s2);
    }
}
```

o/p →

↓ N.S.O

t → [chiru--50, nag--100, venki--150, balalah--200]

t1 → [balalah--200, chiru--50, nag--100, venki--150]

↳ customized based on emp name

## Comparison of Comparable and Comparator

Comparable	Comparator
1) It is maint for Default Natural Sorting order.	1) It is maint for customized Sorting order.
2) present in Java.lang pkg.	2) present in Java.util pkg.
3) only one method it defines. compareTo()	3) It defines two methods. compare() equals()
4) String & All Wrapper classes implement Comparable interface	4) The only implemented classes of comparator are Collator } used in RuleBasedCollator } GUI based app

→ Comparison tables of Set implemented classes:-

Property	Hash Set	Linked HashSet	Tree Set
1) Underlined DS	HashTable	LinkedList + HashTable	Balanced Tree
2) Duplicate Objects	Not allowed	Not allowed	Not allowed
3) Insertion order	Not preserved	preserved	Not preserved
4) Sorting order	X	X	✓
5) Heterogeneous Object	✓	✓	X
6) null acceptance	✓	✓	

For empty TreeSet (1.6) as first element null is allowed

From 1.7 onwards null is not allowed even as the first element.