=> Comparable :

-Comparable is an interface which is present in java.lang package

-It contains only one method i.e. compareTo()

```
   Prototype : public int compareTo(Object obj)
               obj1.compareTo(obj2);
```

->   +ve - if obj1 is greater than obj2

->   -ve - if obj2 is greater than obj1

->   0 - if obj1 is equals to obj2

-String and all wrapper classes (Integer, Long, Float etc) implements Comparable interface

```
public class Test1
{
    public static void main(String[] args)
    {
        TreeSet ts=new TreeSet();

        ts.add("aaa");       //compareTo()
        ts.add("fff");
        ts.add("www");
        ts.add("uuu");
        ts.add("ddd");
        System.out.println(ts);    //Sorted Output

        TreeSet ts1=new TreeSet();

        ts1.add(40);         //compareTo()
        ts1.add(70);
        ts1.add(30);
        ts1.add(60);
        System.out.println(ts1);
    }
}
```

==

To Sort According to the Name:

```
public class Student1 implements Comparable{
    int rollno;
    String name;

    public Student1(int rollno, String name)
    {
        this.rollno=rollno;
        this.name=name;
    }

    @Override
    public int compareTo(Object o)
    {
        Student1 st=(Student1)o;
        int i = this.name.compareTo(st.name) ;
        return i;
    }

    @Override
    public String toString()
    {
        return rollno+"-"+name;
    }
}
```

To Sort According to the rollno:

```
public class Student1 implements Comparable{
    int rollno;
    String name;

    public Student1(int rollno, String name)
    {
        this.rollno=rollno;
        this.name=name;
    }

    @Override
    public int compareTo(Object o)
    {
        Student1 st=(Student1)o;
        Integer i1 = this.rollno ;
        Integer i2 = st.rollno ;
                        ---------------------->
        int i = i1.compareTo(i2) ;
        return i;
    }

    @Override
    public String toString()
    {
        return rollno+"-"+name;
    }
}
```

```
//        if(i1>i2)
//        {
//            return 1;
//        }
//        else if(i1<i2)
//        {
//            return -1;
//        }
//        else
//        {
//            return 0;
//        }
```

```java
public class Test1 {
    public static void main(String[] args) {

        Student1 s2 = new Student1(102, "Amit");
        Student1 s5 = new Student1(105, "Arun");
        Student1 s3 = new Student1(103, "Bhavek");
        Student1 s1 = new Student1(101, "Rahul");
        Student1 s4 = new Student1(104, "Sumit");


//          TreeSet ts=new TreeSet();
//          ts.add(s1);
//          ts.add(s2);
//          ts.add(s3);
//          ts.add(s4);
//          ts.add(s5);
//          System.out.println(ts);

        ArrayList al = new ArrayList();
        al.add(s1);
        al.add(s2);
        al.add(s3);
        al.add(s4);
        al.add(s5);
        Collections.sort(al);
        System.out.println(al);
    }
}
```

Problems with Comparable interface :-

1. By implementing Comparable interface, the properties of Original class will get changed
2. By this way we can sort only for one entity for example we can sort the student object either with name or rollno at one time

(To remove above problems java provided one interface i.e. Comparator)

=> Comparator :-

-> Comparator is an interface which is present in java.util package

-> Comparator interface contains 2 methods :-

1. public int compare(Object obj1, Object obj2)
2. public boolean equals(Object obj)

```
class Student
{
    int rollno;
    String name;

    Student(int rollno, String name)
    {
        this.rollno=rollno;
        this.name=name;
    }

    @Override
    public String toString()
    {
        return rollno+"-"+name;
    }
}
```

```
class SortByName implements Comparator  //class
SortByName extends Object implements Comparator
{
    @Override
    public int compare(Object o1, Object o2)
    {
        Student ss=(Student) o1;
        Student sss=(Student) o2;
        int i=ss.name.compareTo(sss.name);
        return -i;
    }
}


class SortByRollno implements Comparator
{
    @Override
    public int compare(Object o1, Object o2)
    {
        Student ss=(Student) o1;
        Student sss=(Student) o2;
        Integer i1=new Integer(ss.rollno);
        Integer i2=new Integer(sss.rollno);
        int i=i1.compareTo(i2);
        return -i;
    }
}
```

```
public class Test4
{
    public static void main(String[] args)
    {
        Student s1=new Student(101, "Bhavek");
        Student s2=new Student(102, "Pramod");
        Student s3=new Student(103, "Divas");
        Student s4=new Student(104, "Arun");
        Student s5=new Student(105, "Sumit");

        TreeSet ts=new TreeSet(new SortByName());
        ts.add(s1);
        ts.add(s2);
        ts.add(s3);
        ts.add(s4);
        ts.add(s5);
        System.out.println(ts);

        TreeSet ts1=new TreeSet(new SortByRollno());
        ts1.add(s1);
        ts1.add(s2);
        ts1.add(s3);
        ts1.add(s4);
        ts1.add(s5);
        System.out.println(ts1);
    }
}
```

=> TASK : WAP to compare and add different elements according
to length and alphabetical order

For example :

A, B, AA, BB, AAA, BBB, AAAA

```
class MySorting implements Comparator
{
    @Override
    public int compare(Object o1, Object o2)
    {
        String s1=o1.toString();
        String s2=o2.toString();

        int leng1=s1.length();
        int leng2=s2.length();

        if(leng1>leng2)
        {
            return 1;
        }
        else if(leng1<leng2)
        {
            return -1;
        }
        else
        {
            return s1.compareTo(s2);
        }
    }
}
```

```
public class Test
{
    public static void main(String[] args)
    {
        TreeSet ts=new TreeSet(new MySorting());
        ts.add("AAAA");
        ts.add("BB");
        ts.add("A");
        ts.add("BBB");
        ts.add("AA");
        ts.add("AAA");
        ts.add("B");
        System.out.println(ts);
    }
}
```

=> What is difference between Comparable & Comparator interface :-

1. Comparable interface contains compareTo() method
   Comparator interface contains compare() method

2. Comparable interface is present in java.lang package
   Comparator interface is present in java.util package

3. By using Comparable interface original class properties will get changed
   By using Comparator interface original class properties will not be changed

4. Comparable interface can sort only one entity
   Comparator interface can sort multiple entities

5. Comparable interface is used to implicit sorting
   Comparator interface is used for explicit sorting