Java Comparable Interface

Java Comparable interface is a member of collection framework which is used to compare objects and sort them according to the natural order.

The natural ordering refers to the behaviour of compareTo() method which is defined into Comparable interface. Its sorting technique depends on the type of object used by the interface.

If object type is string, then it sorts it Lexicographically.

If object type is wrapper class object like: integer or list then it sorts according to their values.

If object type is custom object like: user defined object then sorts according to the defined compareTo() method.

Classes that implements this interface can be sorted automatically by calling Collections.sort() method.

Declaration of this interface is given below.

public interface Comparable<T>

- **Comparable Method**

It contains single method compareTo() that is given below.

int compareTo(T o) : It compares object with the specified object for order.

The compareTo() method compares the current object with the provided object. This method is already implemented for default wrapper classes and primitive data types but, this method also needs to be implemented for user-defined classes.

It returns positive integer, if the current object is greater than the provided object.

If the current object is less than the provided object then it returns negative integer.

If the current object is equal to the provided object, then it returns zero.

Exceptions

This method returns NullPointerException, if the specified object is null and ClassCastException if the specified object's type prevents it from being compared to this object.

Example : Sorting list

Let's take an example to sort an ArrayList that stores integer values. We are using sort() method of Collections class that sort those objects which implements Compares interface. Since integer wrapper class implements Comparable so we are able to get sorted objects. Let's see the below example.

```
import java.util.*;

public class Demo {
   public static void main(String a[]){
      ArrayList<Integer> list = new ArrayList<>();          // Creating List
      // Adding elements
      list.add(100);
      list.add(2);
      list.add(66);
      list.add(22);
      list.add(10);
      System.out.println(list);          // Displaying list
      Collections.sort(list);            // Sorting list
      System.out.println("Sorted List : "+list);              // Displaying sorted list
   }
}
```

Output:

```
[100, 2, 66, 22, 10]
Sorted List : [2, 10, 22, 66, 100]
```

Example: Sorting String objects

While sorting string objects, the comparable sorts it based on lexicographically. It means a dictionary like sorting order. See the below example.

```
import java.util.*;

class Demo {
   public static void main(String a[]){
      ArrayList<String> list = new ArrayList<>();
      // Adding elements
      list.add("D");
      list.add("L");
      list.add("A");
      list.add("Z");
      list.add("C");
      // Displaying list
      System.out.println(list);
      // Sorting list
      Collections.sort(list);
      System.out.println("Sorted List Asc: "+list);
      Collections.sort(list, Collections.reverseOrder());
      System.out.println("Sorted List Dec: "+list);
   }
}
```

Output:

```
[D, L, A, Z, C]
Sorted List Asc: [A, C, D, L, Z]
Sorted List Dec: [Z, L, D, C, A]
```