Comparable interface

Comparable interface is used to order the objects of user-defined class.This interface is found in java.lang package and contains only one method named compareTo(Object).It provide only single sorting sequence i.e. you can sort the elements on based on single datamember only.For instance it may be either rollno,name,age or anything else.

Syntax:

|  |
| --- |
| **public int compareTo(Object obj):** is used to compare the current object with the specified object. |

We can sort the elements of:

1. String objects
2. Wrapper class objects
3. User-defined class objects

|  |
| --- |
| **Collections** class provides static methods for sorting the elements of collection.If collection elements are of Set type, we can use TreeSet.But We cannot sort the elements of List.Collections class provides methods for sorting the elements of List type elements. |

Method of Collections class for sorting List elements

|  |
| --- |
| **public void sort(List list):** is used to sort the elements of List.List elements must be of Comparable type. |

***Note: String class and Wrapper classes implements the Comparable interface.So if you store the objects of string or wrapper classes, it will be Comparable.***

**Example of Sorting the elements of List that contains user-defined class objects on age basis**

Student.java

1. **class** Student **implements** Comparable{
2. **int** rollno;
3. String name;
4. **int** age;
5. Student(**int** rollno,String name,**int** age){
6. **this**.rollno=rollno;
7. **this**.name=name;
8. **this**.age=age;
9. }
11. **public** **int** compareTo(Object obj){
12. Student st=(Student)obj;
13. **if**(age==st.age)
14. **return** 0;
15. **else** **if**(age>st.age)
16. **return** 1;
17. **else**
18. **return** -1;
19. }
21. }

Simple.java

1. **import** java.util.\*;
2. **import** java.io.\*;
4. **class** TestSort3{
5. **public** **static** **void** main(String args[]){
7. ArrayList al=**new** ArrayList();
8. al.add(**new** Student(101,"Vijay",23));
9. al.add(**new** Student(106,"Ajay",27));
10. al.add(**new** Student(105,"Jai",21));
12. Collections.sort(al);
13. Iterator itr=al.iterator();
14. **while**(itr.hasNext()){
15. Student st=(Student)itr.next();
16. System.out.println(st.rollno+""+st.name+""+st.age);
17. }
18. }
19. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestSort3)

Output:105 Jai 21

101 Vijay 23

106 Ajay 27