**Comparator interface-**

1. We can use comparator to define our own sorting (customized sorting order)
2. It present in java.util package.
3. It defines two methods-

Public int compare(Object obj1, Object obj2)

Public Boolean equals();

1. Whenever we are implementing the comparator interface, compulsory we should provide the implementation for compare() method.
2. Implementing the equals method is optional, because it is already available in every java class from object class.

Example-

**package** com.test;

**public** **class** Student {

**int** id;

String name;

**int** salary;

**public** Student(**int** id, String name, **int** salary) {

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getSalary() {

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";

}

}

**package** com.test;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

ArrayList<Student> al = **new** ArrayList<Student>();

al.add(**new** Student(101, "ram", 9000));

al.add(**new** Student(102, "ashok", 3000));

al.add(**new** Student(103, "ajay", 8000));

Collections.*sort*(al, **new** NameComparator());

**for** (Student student : al) {

System.***out***.println("id>>" + student.getId() + " name>>" + student.getName() + " salary>>" + student.getSalary());

}

}

}

**Sort data by using name**

**package** com.test;

**import** java.util.Comparator;

**public** **class** NameComparator **implements** Comparator<Student> {

@Override

**public** **int** compare(Student student1, Student student2) {

**return** student1.name.compareTo(student2.name);

}

}

**Sort data by using salary**

**package** com.test;

**import** java.util.Comparator;

**public** **class** SalaryComparator **implements** Comparator<Student> {

@Override

**public** **int** compare(Student student1, Student student2) {

**if** (student1.salary == student2.salary)

**return** 0;

**else** **if** (student1.salary > student2.salary)

**return** 1;

**else**

**return** -1;

}

}

|  |  |
| --- | --- |
| Comparable | Comparator |
| It is meant for default natural sorting order | It is meant for customized sorting order |
| It is present in java.lang package | It is present in java.util package |
| This interface defines only one method compareTo() | This interface defines two methods i.e compare() and equals() |