ject Sort Example (Comparable And

ECTIONS

sort an ArrayList of Objects by property using **comparable** and
›king for sorting a simple ArrayList of String or Integer then you

and ArrayList<Integer>
ıding order

method to sort a simple array list. However if the ArrayList is
case you have two options for sorting- **comparable and**
g through the example of them, let's see what's the output when
/ithout implementing any of these interfaces.

ıparable and comparator?

ve a Student class which has properties like Student name, roll

```
ng studentname, int studentage) {

name;
```

```
ge;

{

ng studentname) {
ame;

o) {

tudentage) {
e;
```

Student Object. We do it like this –

```
 args[]){
list = new ArrayList<Student>();
t(223, "Chaitanya", 26));
t(245, "Rahul", 24));
t(209, "Ajeet", 32));
```

ADVERTISE HERE
DeveloperMedia
Connecting you with today's most influential { developers }.
ADVERTISE WITH US

```
st);

st){
println(str);
```

on the List of Objects and boom! I got the the error message like

g.Error: Unresolved compilation problem:
nod sort(List) of type Collections is not applicable for the
 type Student is not a valid substitute for the bounded parameter
in(Details.java:11)

od on an ArrayList of Objects which actually doesn't work until
arable and Comparator.

e importance of these interfaces. Let's see how to use them to get

bject> with Comparable

List<Student> based on the student Age property. This is how it
nparable interface and then Override the **compareTo** method.

```
parable {




ng studentname, int studentage) {

ame;
e;


e as the above example
mparestu) {
mparestu).getStudentage();

areage;

like this */
udentage;




 + ", name=" + studentname + ", age=" + studentage + "]";
```

ons.sort on ArrayList

```
ing args[]){
list = new ArrayList<Student>();
t(223, "Chaitanya", 26));
t(245, "Rahul", 24));
t(209, "Ajeet", 32));

st);

st){
println(str);
```

```
6]
```

need Comparator anymore?

by the same class whose objects are sorted so it binds you with

ɔst of the cases but in case you want to have more than way of
ıld use comparators. Read more about them here:

ɔct> multiple properties with Comparator

ɖ of Comparator for sorting.

```
ıng studentname, int studentage) {

ıame;
ɛe;


ıe as the above examples

ıst by Student Name*/
ıt> StuNameComparator = new Comparator<Student>() {

 s1, Student s2) {
..getStudentname().toUpperCase();
..getStudentname().toUpperCase();


ɔareTo(StudentName2);


ımpareTo(StudentName1);


ıst by roll no*/
ıt> StuRollno = new Comparator<Student>() {

 s1, Student s2) {
```

```
no();
no();

'

 + ", name=" + studentname + ", age=" + studentage + "]";

ing args[]){
list = new ArrayList<Student>();
t(101, "Zues", 26));
t(505, "Abey", 24));
t(809, "Vignesh", 32));

nt Name*/
ent Name Sorting:");
st, Student.StuNameComparator);

st){
println(str);


perty*/
Num Sorting:");
st, Student.StuRollno);
st){
println(str);
```

e related posts

ample

st example

example – Java

using index in java example

ımple in java

PM

M

1:21 PM

says that student is not an abstract class and doesn’t override
ator even i used same codes with yours

ys
12:20 PM

the compareto method “method doesnot override or
a supertype” when i replace(object o)with (Student

7:56 PM

ame error, even after i copy paste this same syntax. There is
e way we are overriding the method.

says

1:25 PM

issue and I realize that Comparable interface assume Object

(Object comparestu) as parameter when overriding compareTo
need to make sure comparestu is of type Student,by casting:
e able to access getStudentage method.
d by specifying the parameter of Comparable interface when
in our case. (Student comparestu)will be compareTo
cast at this line: int
getStudentage();

s

7:39 AM

nd great explanation…

3 AM

forgot to specify the type…

11:17 PM

›e is mentioned, he doesn’t need to typecase inside the
›z he’s already getting the argument as Student.

:50 AM

you print out an object with the highest value of one of their

ıt the name of the student with the highest age from an array
lo you do that?

10:00 PM

/ one field in a custom object. Can you advise on how to sort by

ıen sort by name then…

3:03 AM

3OGBA for the clarification!

6 AM

e nd comparator concepts

ys
:34 AM

nt to sort for multiple values like first by age then by name

ome objects , call these mehtods of sorting in order in which

Comparator() {

in lhs, Campain rhs) {
;
);
rmat = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

se(lhs.getDatetime());
se(rhs.getDatetime());

{

<= rhsDate.getTime() ? -1 : 1;
lhsDate.getTime())

) < lhsDate.getTime() ? -1 : 1;
(lhsDate);

in for sorting

ys
.0:04 AM

e description. I want to ask something 1. Is comparable used
an integer property ?? because in other threads it can be used
pe as well and its working.

rbles compareTO()method here is taking Student object as
king when i tried, although when passing only base Object
dent from it works well.

ct obj){

11:27 AM

ead. Thank you very much.

says

7 AM

th SAME NAME but different Roll no and want to sort
en roll no ?

AM

:Sorting will give an error :-
d does not override the abstract method compareTo(Object) in

ange the line
lent comparestu) {
gram as :-
ct comparestu) {

ecute. and output will come.
, age=24]
nya, age=26]
age=32]

ys
8 PM

problem since many day finally I did change accordingly your

M

utorial.

3:18 PM

ype using Comparable.

lent st) {
To(st.name);

s
05 PM

rface .. You can sort on any field like name or tag etc… other

ıtation of compareTo method like

ıent st) {

areTo(student.getStudentName());

u can call

ent of Comparator interface…

ngh says

9 PM

topic. Just a little correction, As you highlighted, Using
ustom object only based on integer type. This is not correct, We
String base also, Like you shared Student Object above,
age, Let me show compareTo method performing sort based on

ıent compareStudent) {
To(compareStudent.name);

ort Student object based on name (String Type).

published. Required fields are marked *

Java Tutorial

- Java Tutorial
- OOPs Concepts

Java Collections

- ArrayList
- LinkedList
- ArrayList vs LinkedList
- Vector
- ArrayList vs Vector
- HashMap
- TreeMap
- LinkedHashMap
- HashSet
- TreeSet
- LinkedHashSet
- Hashtable
- HashMap vs Hashtable
- Queue
- PriorityQueue
- Deque & ArrayDeque
- Iterator
- ListIterator
- Comparable Interface
- Comparator Interface
- Java Collections Interview Q

MORE ...

- Java String
- Exception handling
- Java Multithreading