**How to sort a TreeMap by value in java**

[**JAVA COLLECTIONS**](http://beginnersbook.com/category/java-collections/)

[**A TreeMap**](http://beginnersbook.com/2013/12/treemap-in-java-with-example/) is always sorted based on its keys, however if you want to sort it based on its values then you can build a logic to do this using comparator. Below is a complete code of sorting a TreeMap by values.

import java.util.\*;

class TreeMapDemo {

//Method for sorting the TreeMap based on values

public static <K, V extends Comparable<V>> Map<K, V>

sortByValues(final Map<K, V> map) {

Comparator<K> valueComparator =

new Comparator<K>() {

public int compare(K k1, K k2) {

int compare =

map.get(k1).compareTo(map.get(k2));

if (compare == 0)

return 1;

else

return compare;

}

};

Map<K, V> sortedByValues =

new TreeMap<K, V>(valueComparator);

sortedByValues.putAll(map);

return sortedByValues;

}

public static void main(String args[]) {

TreeMap<String, String> treemap = new TreeMap<String, String>();

// Put elements to the map

treemap.put("Key1", "Jack");

treemap.put("Key2", "Rick");

treemap.put("Key3", "Kate");

treemap.put("Key4", "Tom");

treemap.put("Key5", "Steve");

// Calling the method sortByvalues

Map sortedMap = sortByValues(treemap);

// Get a set of the entries on the sorted map

Set set = sortedMap.entrySet();

// Get an iterator

Iterator i = set.iterator();

// Display elements

while(i.hasNext()) {

Map.Entry me = (Map.Entry)i.next();

System.out.print(me.getKey() + ": ");

System.out.println(me.getValue());

}

}

}

**Output:**

Key1: Jack

Key3: Kate

Key2: Rick

Key5: Steve

Key4: Tom