**How to get the Sub Map from TreeMap example – Java**

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

In this example we are gonna see how to get a sub map from TreeMap. We are using subMap() method of TreeMap class. Please refer the comments in the below program for more details.

**Example**

import java.util.\*;

class TreeMapDemo {

public static void main(String args[]) {

// Create a TreeMap

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");

treemap.put("Key6", "Ram");

// Displaying TreeMap elements

System.out.println("TreeMap Contains : " + treemap);

// Getting the sub map

/\* public SortedMap<K,V> subMap(K fromKey,K toKey): Returns

\* a view of the portion of this map whose keys range from

\* fromKey, inclusive, to toKey, exclusive.

\* (If fromKey and toKey are equal, the returned map is empty.)

\* The returned map is backed by this map, so changes in the

\* returned map are reflected in this map, and vice-versa.

\* The returned map supports all optional map operations that

\* this map supports.

\*/

SortedMap<String, String> sortedMap = treemap.subMap("Key2","Key5");

System.out.println("SortedMap Contains : " + sortedMap);

// Removing an element from Sub Map

sortedMap.remove("Key4");

/\* Displaying elements of original TreeMap after

\* removing an element from the Sub Map. Since Sub Map is

\* backed up by original Map, the element should be removed

\* from this TreeMap too.

\*/

System.out.println("TreeMap Contains : " + treemap);

}

}

**Output:**

TreeMap Contains : {Key1=Jack, Key2=Rick, Key3=Kate, Key4=Tom, Key5=Steve, Key6=Ram}

SortedMap Contains : {Key2=Rick, Key3=Kate, Key4=Tom}

TreeMap Contains : {Key1=Jack, Key2=Rick, Key3=Kate, Key5=Steve, Key6=Ram}