Sort Map by Values Generic Version

In previous posts we saw how to sort map by [Key](http://data-structure-learning.blogspot.com/2015/06/sort-hashmap-by-keys.html) & [Value](http://data-structure-learning.blogspot.com/2015/06/sort-hashmap-by-values.html).

We also discussed the [Generic Version of Sorting Map by Key](http://data-structure-learning.blogspot.com/2015/06/sort-map-by-keys-generic-version.html). I would recommend that you go through all the above post as they are prerequisite to this post.

The details of this post are written in [previous](http://data-structure-learning.blogspot.com/2015/06/sort-map-by-keys-generic-version.html) post so I will just write the code and highlight the portion of the code that we have changed.

Input for the Map

Map<String, Integer> map = **new** HashMap<String, Integer>();

map.put("Bran", 45);

map.put("Robert", 34);

map.put("Roose", 43);

map.put("Lynaa", 54);

map.put("Aegon", 39);

map.put("John", 12);

**public** **static** <K, **V extends Comparable<? super V>**> Map<K, V> sortByValue(Map<K, V> map) {

List<Map.Entry<K, V>> list = **new** LinkedList<Map.Entry<K, V>>(

map.entrySet());

Collections.*sort*(list, **new** Comparator<Map.Entry<K, V>>() {

@Override

**public** **int** compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) {

**return** **o1.getValue().compareTo((o2).getValue());**

}

});

Map<K, V> result = **new** LinkedHashMap<>();

**for** (Map.Entry<K, V> entry : list) {

result.put(entry.getKey(), entry.getValue());

}

System.***out***.println(result);

**return** result;

}

Lines in bold are changed.

Output is {John=12, Robert=34, Aegon=39, Roose=43, Bran=45, Lynaa=54}

If you read this post with referring the [previous](http://data-structure-learning.blogspot.com/2015/06/sort-map-by-keys-generic-version.html) post then you won’t understand this concept. Please refer [previous](http://data-structure-learning.blogspot.com/2015/06/sort-map-by-keys-generic-version.html) post.