```java
import java.util.TreeMap;

public class maincomparator {
    public static void main(String[] args) {
        TreeMap<String ,Integer>numbers=new TreeMap
<>(new CustomComparator());
        numbers.put("First",1);
        numbers.put("Second",2);
        numbers.put("Third",3);
        numbers.put("Fourth",4);
        System.out.println("TreeMap: "+numbers);
    }
}

```

```java
import java.util.Comparator;

public class CustomComparator implements Comparator<
String> {

    @Override
    public int compare(String o1, String o2) {
        int value = o1.compareTo(o2);
        if(value>0) return -1;
        else if (value<0) {
            return 1;

        }else{
            return 0;
        }
    }
}

```

```
 1 import java.util.HashMap;
 2
 3 public class ExampleOneHashMap {
 4     public static void main(String[] args){
 5         HashMap<Integer,String> languages = new
   HashMap<>();
 6         languages.put(1, "Java");
 7         languages.put(2,"Python");
 8         languages.put(3, "JavaScript");
 9         languages.put(4, "C Sharp");
10         System.out.println("HashMap: " + languages);
11
12         String value = languages.remove(2);
13         System.out.println("Removed value: " + value
   );
14         System.out.println("Updated HashMap: " +
   languages);
15     }
16 }
17
```

```java
import java.util.TreeMap;

public class exampleTreemapOne {
    public static void main(String[] args) {
        TreeMap<String,Integer> numbers = new TreeMap<>();
        numbers.put("One", 1);
        numbers.put("Two",2);
        numbers.put("Three",3);
        System.out.println("TreeMap: " + numbers);

        int value = numbers.remove("two");
        System.out.println("Removed value: " + value);

        boolean result = numbers.remove("Three", 3);
        System.out.println("Is the entry {Three=3} removed " + result);
        System.out.println("Updated TreeMap: " + numbers);
    }
}

```

```
 1 import java.util.TreeMap;
 2
 3 public class exampleTreemapTwo {
 4     public static void main(String[] args) {
 5         TreeMap<String, Integer> numbers = new
   TreeMap<>();
 6         numbers.put("First", 1);
 7         numbers.put("Second", 2);
 8         numbers.put("Three", 3);
 9         System.out.println("TreeMap: " + numbers);
10
11         String firstKey = numbers.firstKey();
12         System.out.println("First Key: " + firstKey);
13
14         String lastKey = numbers.lastKey();
15         System.out.println("Last key: " + lastKey);
16
17         System.out.println("First Entry: " + numbers.
   firstEntry());
18
19         // Using the lastEntry() method
20         System.out.println("Last Entry: " + numbers.
   lastEntry());    }
21 }
22
```

```
 1 import java.util.HashMap;
 2
 3 public class ExampletwoHashMap {
 4     public static void main(String[] args){
 5         HashMap<String,String> newHashMap = new
   HashMap<>();
 6         newHashMap.put("key1", "Lenovo");
 7         newHashMap.put("key2", "Motorola");
 8         newHashMap.put("key3", "Nokia");
 9         newHashMap.put("key4", null);
10         newHashMap.put(null, "Sony");
11         System.out.println("Original map contains: "
    + newHashMap);
12
13         System.out.println("Size of original Map is
   : " + newHashMap.size());
14
15         HashMap<String,String> copyHashMap = new
   HashMap<>();
16         copyHashMap.putAll(newHashMap);
17         System.out.println("copyHashMap mappings= "
    + copyHashMap);
18         String nullKeyValue = copyHashMap.remove(null
   );
19         System.out.println("copyHashMap null key
   value = " + nullKeyValue);
20         System.out.println("copyHashMap after
   removing null key = " + copyHashMap);
21         System.out.println("Size of copyHashMap is:"
    + copyHashMap.size());
22
23     }
24 }
25
```