Lab KeyValuePair<K,V> - Part 2

Description:

- Implement the interface `Comparable<KeyValuePair<K,V>>`
  Notice how we substituted a generic type *KeyValuePair<K,V>* for the type parameter *E* in `Comparable<E>`.
  Have Eclipse generate the method stub for the interface method `compareTo`.
  The method `compareTo` will be used to compare two instances of `KeyValuePair<K,V>`.
  Don't implement it quite yet though.
  First let's do a bit more planning: how should we implement compareTo ?
- What is it, that makes an instance of a `KeyValuePair<K,V>` smaller or greater than another?
  For our implementation we say that an instance of a KeyValuePait<K,V> is smaller if and only if the key value is smaller.
  Similarly an instance of a KeyValuePair<K,V> is greater than another if and only if the key value is greater.
- Now we shifted the problem: how can we tell whether a key value is smaller or greater?
  It would be helpful if K (the type of the key) implemented the Comparable<K> interface. In that case we could just use the compareTo method of K to find out whether a key value is smaller or greater.
- To allow us to do that we will modify class `KeyValuePair<K,V>` so that it restricts the possible types that can be used as a type argument for K. We will only allow types that implement Comparable<K>
- How can that be done?
  Change the unbounded type parameter *K* in `KeyValuePair<K,V>` to a bounded type parameter *K extends Comparable<K>*.
  like this: `KeyValuePair<K extends Comparable<K>, V>`
- The header of your class declarations should look now like this:

```
public class KeyValuePair<K extends Comparable<K>, V>
    implements Comparable< KeyValuePair <K, V>>
```

  Notice that you specify the restriction on type K only once - when you first introduce the type parameter.
- Now you are ready to implement the method compareTo.
  Make sure that two KeyValuePairs are greater / smaller based on the value of the key.

- In main do the following:
  - Create 2 more KeyValuePairs for
    LA .. 3819702
    SF .. 812826
  - Create a generic List, call it cities, and initialize it with the 4 KeyValuePairs that we have created
  - Print the list – one item per line (label the output)
  - Sort the list
  - Print the now sorted list – one item per line

Output:

```
Original List:
SLC: 189899
NY: 8244910
LA: 3819702
SF: 812826

Sorted List:
LA: 3819702
NY: 8244910
SF: 812826
SLC: 189899
```