

The java.lang.Comparable interface:

`int compareTo(Object ob)` Returns a negative value if **this** is less than `ob`, 0 if they are equal, and a positive value if **this** is greater than `ob`

The java.lang.Comparable<T> interface:

`int compareTo(T t)` Returns a negative value if **this** is less than `t`, 0 if they are equal, and a positive value if **this** is greater than `t`

```
public class Item implements Comparable<Item> {
    private int rating;
    private String itemName;
    public Item(String name) { itemName = name; }
    public int compareTo(Item other) {
        int d = rating - other.rating;
        if ( d != 0 ) return d;
        return itemName.compareTo(other.itemName);
    }
    public String toString() { return itemName + ":" + rating; }
    ... // other methods for updating rating of this item
}
```

Sort an array Objects (of type Item)

1. Write code that sorts an array of items named `itemArray`.
2. Is the array sorted by **rating,item name** or by **item name,rating**?
3. How would an array of items be sorted if the `compareTo` method is defined as shown here:

```
public int compareTo(Item other) {
    return toString().compareTo( other.toString() );
}
```

Sort an array Objects (of type Item)

1. Write code that sorts an `ArrayList<Item>` of items named `itemList`.
2. How would an `ArrayList` of items be sorted if the `compareTo` method is defined as shown here:

```
public int compareTo(Item other) {
    int d = other.rating - this.rating;
    if ( d != 0 ) return d;
    return other.itemName.compareTo( this.itemName );
}
```