

Interfaces used for sorting

Now that I've covered interfaces and generic classes, I want to review in more detail, interfaces I mentioned in previous lectures.

The first is Comparable.

For an array, we can simply call `Arrays.sort`, and pass it an array, but as I have previously mentioned, the elements in the array, need to implement Comparable.

Types like `String`, or primitive wrapper classes like `Integer` or `Character` are sortable, and this is because they do implement this interface.

Comparable Interface

The interface declaration in Java.

```
public interface Comparable<T> {  
    int compareTo(T o);  
}
```

It's a generic type, meaning it's parameterized.

Any class that implements this interface, needs to implement the compareTo method.

Comparable Interface

```
public interface Comparable<T> {  
  
    int compareTo(T o);  
  
}
```

This method takes one object as an argument, shown on this slide as the letter o, and compares it to the current instance, shown as this.

The table on this slide shows what the results of the compareTo method should mean, when implemented.

| resulting Value | Meaning |
|-----------------|-----------------------------|
| zero | <code>O == t h i s</code> |
| negative value | <code>t h i s < o</code> |
| positive value | <code>t h i s > o</code> |