**CSCI 2302**

**Abstract Classes & Interfaces Chapter**

**Comparable Lab**

Contents

[Intro 1](#_Toc99286304)

[Notes 1](#_Toc99286305)

[Learning Goals 3](#_Toc99286306)

[Task 3](#_Toc99286307)

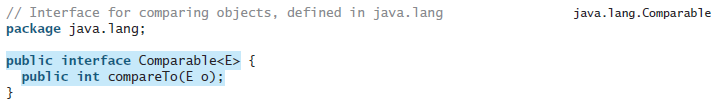
[Submit 4](#_Toc99286308)

Intro: Java Comparable interface is used to compare objects and thus can sort them according to the natural order.

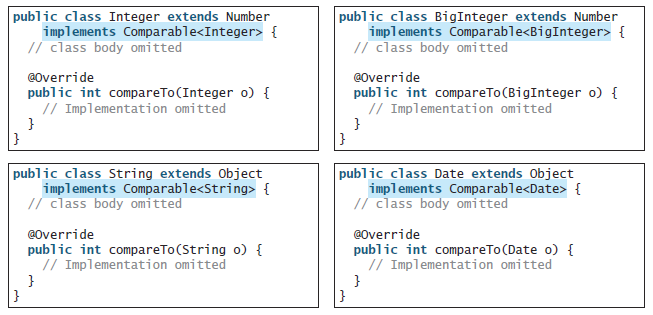
Notes: The Comparable interface defines the compareTo method for comparing objects. The compareTo method determines the order of this object (this) with the specified/passed in object (*object* o), and if this is less then o -1 is returned, if this is the same as/equal to o 0 is returned, and if this is greater than o 1 is returned.

|  |  |
| --- | --- |
| **Comparing Result** | **Return Value** |
| this < o | -1 |
| this == o | 0 |
| this > o | 1 |

In order to accomplish this, the two objects must be comparable, so the common behavior for the objects must be comparable. Java provides the Comparable interface for this purpose. The interface is defined as follows:



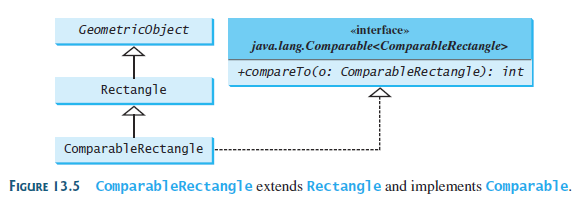
The Comparable interface is a generic interface. The generic type E is replaced by a concrete type when implementing this interface. Examples:



Thus, numbers are comparable, strings are comparable, as well as dates. You can use the compareTo method to compare two numbers, two strings, and two dates. See SortComparableObjects.java for in use example.

Since the compareTo method is ***not*** defined in the Object class, the Comparable interface is defined in Java to enable objects to be compared if they are instances of the Comparable interface. It is strongly recommended (though not required) that compareTo should be consistent with equals. That is, for two objects o1 and o2, o1.compareTo(o2) == 0 if and only if o1.equals(o2) is true.

UML Diagram:



Learning Goals:

To practice abstract class implementation.

To learn how to implement the Comparable interface and to use the compareTo method.

Task: Complete the steps outlined below:

1. Download the Comparable\_Lab and rename it with mySFAuserName\_Comparable\_Lab.java.
2. Define/Implement the abstract methods from the abstract GeometricObject classes to the inherited classes: Circle.java, Rectangle.java, and Triangle.java.
3. Implement the Comparable interface in each of the geometric object classes, Rectangle, Circle, Triangle, and override the compareTo method in each class (see below example Rectangle.java).

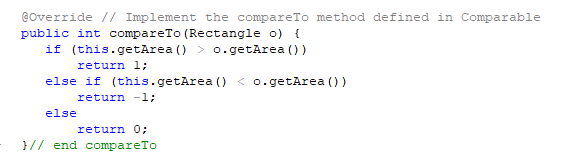
To do so, make sure each of the geometric object subclasses have the following added to the class:

implements Comparable<datatype>

Example:



Then, in each of those geometric object subclasses, override the compareTo method:



1. Modify the Comparable\_Lab.java to complete the following:
   1. Instantiate at least two of each geometric object (i. e. 2-3 circles, 2-3 rectangles, 2-3 triangles) and then invoke the compareTo method; see if you can do 2 of the same kind of geometric objects (i. e. circle to circle) and 2 different kinds of geometric objects (i. e. circle to triangle) so you can see how the method works.

Example of an output:

Creating 2 circles with the same radius

Comparing the circles: 0

Creating 2 triangles where the second triangle has larger sides

Comparing the triangles: -1

Creating 2 rectangles where the first rectangle has a larger width

Comparing the rectangles: 1

Comparing rectangle to a circle: 1

Submit: Submit mySFAUserName\_Comparable\_Lab.java in the Dropbox in Brightspace by D2L.