| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Comparable.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/lang/CloneNotSupportedException.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/Compiler.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Comparable.html)    [**NO FRAMES**](http://docs.google.com/Comparable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **java.lang**

Interface Comparable<T>

**Type Parameters:**T - the type of objects that this object may be compared to **All Known Subinterfaces:** [Delayed](http://docs.google.com/java/util/concurrent/Delayed.html), [Name](http://docs.google.com/javax/naming/Name.html), [RunnableScheduledFuture](http://docs.google.com/java/util/concurrent/RunnableScheduledFuture.html)<V>, [ScheduledFuture](http://docs.google.com/java/util/concurrent/ScheduledFuture.html)<V> **All Known Implementing Classes:** [Authenticator.RequestorType](http://docs.google.com/java/net/Authenticator.RequestorType.html), [BigDecimal](http://docs.google.com/java/math/BigDecimal.html), [BigInteger](http://docs.google.com/java/math/BigInteger.html), [Boolean](http://docs.google.com/java/lang/Boolean.html), [Byte](http://docs.google.com/java/lang/Byte.html), [ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html), [Calendar](http://docs.google.com/java/util/Calendar.html), [Character](http://docs.google.com/java/lang/Character.html), [CharBuffer](http://docs.google.com/java/nio/CharBuffer.html), [Charset](http://docs.google.com/java/nio/charset/Charset.html), [ClientInfoStatus](http://docs.google.com/java/sql/ClientInfoStatus.html), [CollationKey](http://docs.google.com/java/text/CollationKey.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [CompositeName](http://docs.google.com/javax/naming/CompositeName.html), [CompoundName](http://docs.google.com/javax/naming/CompoundName.html), [Date](http://docs.google.com/java/util/Date.html), [Date](http://docs.google.com/java/sql/Date.html), [Desktop.Action](http://docs.google.com/java/awt/Desktop.Action.html), [Diagnostic.Kind](http://docs.google.com/javax/tools/Diagnostic.Kind.html), [Dialog.ModalExclusionType](http://docs.google.com/java/awt/Dialog.ModalExclusionType.html), [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Double](http://docs.google.com/java/lang/Double.html), [DoubleBuffer](http://docs.google.com/java/nio/DoubleBuffer.html), [DropMode](http://docs.google.com/javax/swing/DropMode.html), [ElementKind](http://docs.google.com/javax/lang/model/element/ElementKind.html), [ElementType](http://docs.google.com/java/lang/annotation/ElementType.html), [Enum](http://docs.google.com/java/lang/Enum.html), [File](http://docs.google.com/java/io/File.html), [Float](http://docs.google.com/java/lang/Float.html), [FloatBuffer](http://docs.google.com/java/nio/FloatBuffer.html), [Formatter.BigDecimalLayoutForm](http://docs.google.com/java/util/Formatter.BigDecimalLayoutForm.html), [FormSubmitEvent.MethodType](http://docs.google.com/javax/swing/text/html/FormSubmitEvent.MethodType.html), [GregorianCalendar](http://docs.google.com/java/util/GregorianCalendar.html), [GroupLayout.Alignment](http://docs.google.com/javax/swing/GroupLayout.Alignment.html), [IntBuffer](http://docs.google.com/java/nio/IntBuffer.html), [Integer](http://docs.google.com/java/lang/Integer.html), [JavaFileObject.Kind](http://docs.google.com/javax/tools/JavaFileObject.Kind.html), [JTable.PrintMode](http://docs.google.com/javax/swing/JTable.PrintMode.html), [KeyRep.Type](http://docs.google.com/java/security/KeyRep.Type.html), [LayoutStyle.ComponentPlacement](http://docs.google.com/javax/swing/LayoutStyle.ComponentPlacement.html), [LdapName](http://docs.google.com/javax/naming/ldap/LdapName.html), [Long](http://docs.google.com/java/lang/Long.html), [LongBuffer](http://docs.google.com/java/nio/LongBuffer.html), [MappedByteBuffer](http://docs.google.com/java/nio/MappedByteBuffer.html), [MemoryType](http://docs.google.com/java/lang/management/MemoryType.html), [MessageContext.Scope](http://docs.google.com/javax/xml/ws/handler/MessageContext.Scope.html), [Modifier](http://docs.google.com/javax/lang/model/element/Modifier.html), [MultipleGradientPaint.ColorSpaceType](http://docs.google.com/java/awt/MultipleGradientPaint.ColorSpaceType.html), [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html), [NestingKind](http://docs.google.com/javax/lang/model/element/NestingKind.html), [Normalizer.Form](http://docs.google.com/java/text/Normalizer.Form.html), [ObjectName](http://docs.google.com/javax/management/ObjectName.html), [ObjectStreamField](http://docs.google.com/java/io/ObjectStreamField.html), [Proxy.Type](http://docs.google.com/java/net/Proxy.Type.html), [Rdn](http://docs.google.com/javax/naming/ldap/Rdn.html), [Resource.AuthenticationType](http://docs.google.com/javax/annotation/Resource.AuthenticationType.html), [RetentionPolicy](http://docs.google.com/java/lang/annotation/RetentionPolicy.html), [RoundingMode](http://docs.google.com/java/math/RoundingMode.html), [RowFilter.ComparisonType](http://docs.google.com/javax/swing/RowFilter.ComparisonType.html), [RowIdLifetime](http://docs.google.com/java/sql/RowIdLifetime.html), [RowSorterEvent.Type](http://docs.google.com/javax/swing/event/RowSorterEvent.Type.html), [Service.Mode](http://docs.google.com/javax/xml/ws/Service.Mode.html), [Short](http://docs.google.com/java/lang/Short.html), [ShortBuffer](http://docs.google.com/java/nio/ShortBuffer.html), [SOAPBinding.ParameterStyle](http://docs.google.com/javax/jws/soap/SOAPBinding.ParameterStyle.html), [SOAPBinding.Style](http://docs.google.com/javax/jws/soap/SOAPBinding.Style.html), [SOAPBinding.Use](http://docs.google.com/javax/jws/soap/SOAPBinding.Use.html), [SortOrder](http://docs.google.com/javax/swing/SortOrder.html), [SourceVersion](http://docs.google.com/javax/lang/model/SourceVersion.html), [SSLEngineResult.HandshakeStatus](http://docs.google.com/javax/net/ssl/SSLEngineResult.HandshakeStatus.html), [SSLEngineResult.Status](http://docs.google.com/javax/net/ssl/SSLEngineResult.Status.html), [StandardLocation](http://docs.google.com/javax/tools/StandardLocation.html), [String](http://docs.google.com/java/lang/String.html), [SwingWorker.StateValue](http://docs.google.com/javax/swing/SwingWorker.StateValue.html), [Thread.State](http://docs.google.com/java/lang/Thread.State.html), [Time](http://docs.google.com/java/sql/Time.html), [Timestamp](http://docs.google.com/java/sql/Timestamp.html), [TimeUnit](http://docs.google.com/java/util/concurrent/TimeUnit.html), [TrayIcon.MessageType](http://docs.google.com/java/awt/TrayIcon.MessageType.html), [TypeKind](http://docs.google.com/javax/lang/model/type/TypeKind.html), [URI](http://docs.google.com/java/net/URI.html), [UUID](http://docs.google.com/java/util/UUID.html), [WebParam.Mode](http://docs.google.com/javax/jws/WebParam.Mode.html), [XmlAccessOrder](http://docs.google.com/javax/xml/bind/annotation/XmlAccessOrder.html), [XmlAccessType](http://docs.google.com/javax/xml/bind/annotation/XmlAccessType.html), [XmlNsForm](http://docs.google.com/javax/xml/bind/annotation/XmlNsForm.html)

public interface **Comparable<T>**

This interface imposes a total ordering on the objects of each class that implements it. This ordering is referred to as the class's *natural ordering*, and the class's compareTo method is referred to as its *natural comparison method*.

Lists (and arrays) of objects that implement this interface can be sorted automatically by [Collections.sort](http://docs.google.com/java/util/Collections.html#sort(java.util.List)) (and [Arrays.sort](http://docs.google.com/java/util/Arrays.html#sort(java.lang.Object%5B%5D))). Objects that implement this interface can be used as keys in a [sorted map](http://docs.google.com/java/util/SortedMap.html) or as elements in a [sorted set](http://docs.google.com/java/util/SortedSet.html), without the need to specify a [comparator](http://docs.google.com/java/util/Comparator.html).

The natural ordering for a class C is said to be *consistent with equals* if and only if e1.compareTo(e2) == 0 has the same boolean value as e1.equals(e2) for every e1 and e2 of class C. Note that null is not an instance of any class, and e.compareTo(null) should throw a NullPointerException even though e.equals(null) returns false.

It is strongly recommended (though not required) that natural orderings be consistent with equals. This is so because sorted sets (and sorted maps) without explicit comparators behave "strangely" when they are used with elements (or keys) whose natural ordering is inconsistent with equals. In particular, such a sorted set (or sorted map) violates the general contract for set (or map), which is defined in terms of the equals method.

For example, if one adds two keys a and b such that (!a.equals(b) && a.compareTo(b) == 0) to a sorted set that does not use an explicit comparator, the second add operation returns false (and the size of the sorted set does not increase) because a and b are equivalent from the sorted set's perspective.

Virtually all Java core classes that implement Comparable have natural orderings that are consistent with equals. One exception is java.math.BigDecimal, whose natural ordering equates BigDecimal objects with equal values and different precisions (such as 4.0 and 4.00).

For the mathematically inclined, the *relation* that defines the natural ordering on a given class C is:

{(x, y) such that x.compareTo(y) <= 0}.

The *quotient* for this total order is:

{(x, y) such that x.compareTo(y) == 0}.

It follows immediately from the contract for compareTo that the quotient is an *equivalence relation* on C, and that the natural ordering is a *total order* on C. When we say that a class's natural ordering is *consistent with equals*, we mean that the quotient for the natural ordering is the equivalence relation defined by the class's [equals(Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) method:

{(x, y) such that x.equals(y)}.

This interface is a member of the  [Java Collections Framework](http://docs.google.com/technotes/guides/collections/index.html).

**Since:** 1.2 **See Also:**[Comparator](http://docs.google.com/java/util/Comparator.html)

| **Method Summary** | |
| --- | --- |
| int | [**compareTo**](http://docs.google.com/java/lang/Comparable.html#compareTo(T))([T](http://docs.google.com/java/lang/Comparable.html) o)            Compares this object with the specified object for order. |

| **Method Detail** |
| --- |

### compareTo

int **compareTo**([T](http://docs.google.com/java/lang/Comparable.html) o)

Compares this object with the specified object for order. Returns a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.

The implementor must ensure sgn(x.compareTo(y)) == -sgn(y.compareTo(x)) for all x and y. (This implies that x.compareTo(y) must throw an exception iff y.compareTo(x) throws an exception.)

The implementor must also ensure that the relation is transitive: (x.compareTo(y)>0 && y.compareTo(z)>0) implies x.compareTo(z)>0.

Finally, the implementor must ensure that x.compareTo(y)==0 implies that sgn(x.compareTo(z)) == sgn(y.compareTo(z)), for all z.

It is strongly recommended, but *not* strictly required that (x.compareTo(y)==0) == (x.equals(y)). Generally speaking, any class that implements the Comparable interface and violates this condition should clearly indicate this fact. The recommended language is "Note: this class has a natural ordering that is inconsistent with equals."

In the foregoing description, the notation sgn(*expression*) designates the mathematical *signum* function, which is defined to return one of -1, 0, or 1 according to whether the value of *expression* is negative, zero or positive.

**Parameters:**o - the object to be compared. **Returns:**a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the specified object's type prevents it from being compared to this object.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Comparable.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/lang/CloneNotSupportedException.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/Compiler.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Comparable.html)    [**NO FRAMES**](http://docs.google.com/Comparable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).