

System.IComparable<T> Interface

```
[ILAsm]  
.class interface public abstract IComparable`1<T>  
  
[C#]  
public interface IComparable<T>
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Summary

Defines generalized comparison methods that a value type or class implements to create a type-specific comparison method.

Library: BCL

Description

This interface is implemented by types whose values can be ordered; for example, the numeric and string classes.

IComparable<T>.CompareTo(T) Method

```
[ILAsm]  
.method public hidebysig virtual abstract int32 CompareTo(!0 obj)  
  
[C#]  
int CompareTo(T obj)
```

Summary

Returns the sort order of the current instance compared to the specified object.

Parameters

Parameter	Description
<i>obj</i>	The <i>T</i> to compare to the current instance.

Return Value

A value that reflects the sort order of the current instance as compared to *obj*. The following table defines the conditions under which the returned value is a negative number, zero, or a positive number.

Returned Value	Description
A negative value	The current instance is < <i>obj</i> .
Zero	The current instance is == <i>obj</i> .
A positive value	The current instance is > than <i>obj</i> .

Behaviors

For any objects A, B and C, the following are required to be true:

A.CompareTo(A) is required to return zero.

If A.CompareTo(B) returns zero then B.CompareTo(A) is required to return zero.

If A.CompareTo(B) is zero, then B.CompareTo(C) and A.CompareTo(C) must have the same sign (negative, zero or positive).

If B.CompareTo(C) is zero, then A.CompareTo(B) and A.CompareTo(C) must have the same sign (negative, zero or positive).

If A.CompareTo(B) returns zero and B.CompareTo(C) returns zero then A.CompareTo(C) is required to return zero.

If `A.CompareTo(B)` returns a value other than zero then `B.CompareTo(A)` is required to return a value of the opposite sign.

If `A.CompareTo(B)` returns a value x not equal to zero, and `B.CompareTo(C)` returns a value y of the same sign as x , then `A.CompareTo(C)` is required to a value of the same sign as x and y .

The exact behavior of this method is unspecified. The intent of this method is to provide a mechanism that orders instances of a class in a manner that is consistent with the mathematical definitions of the relational operators ($<$, $>$, and $=$), without regard for class-specific definitions of the operators.

Usage

Use the `System.IComparable<T>.CompareTo(T)` method to determine the ordering of instances of a class.