IComparer interface is used in conjunction with the Array.Sort and

Array.BinarySearch methods. It provides a way to customize the sort order of a

collection.

Compares two objects and returns a value indicating whether one is less than, equal to, or greater than the other.

**Namespace:**  [System.Collections](https://msdn.microsoft.com/en-us/library/system.collections%28v=vs.110%29.aspx)  
**Assembly:**  mscorlib (in mscorlib.dll)

[**Syntax**](javascript:void(0))

C#

int Compare(

Object x,

Object y

)

**Parameters**

x

Type: [System.Object](https://msdn.microsoft.com/en-us/library/system.object%28v=vs.110%29.aspx)

The first object to compare.

y

Type: [System.Object](https://msdn.microsoft.com/en-us/library/system.object%28v=vs.110%29.aspx)

The second object to compare.

**Return Value**

Type: [System.Int32](https://msdn.microsoft.com/en-us/library/system.int32%28v=vs.110%29.aspx)  
A signed integer that indicates the relative values of x and y, as shown in the following table.

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| Less than zero | x is less than y. |
| Zero | x equals y. |
| Greater than zero | x is greater than y. |

[**Exceptions**](javascript:void(0))

|  |  |
| --- | --- |
| **Exception** | **Condition** |
| [ArgumentException](https://msdn.microsoft.com/en-us/library/system.argumentexception%28v=vs.110%29.aspx) | Neither x nor y implements the [IComparable](https://msdn.microsoft.com/en-us/library/system.icomparable%28v=vs.110%29.aspx) interface.  -or-  x and y are of different types and neither one can handle comparisons with the other. |