

# System.Collections.Generic.IEqualityComparer<T> Interface

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

Provides a mechanism to customize equality in sort ordering of a generic collection.

**Library:** BCL

# IEqualityComparer<T>.Equals(T, T) Method

```
[ILAsm]  
.method public hidebysig virtual abstract bool Equals(!0 x, !0 y)  
  
[C#]  
bool Equals(T x, T y)
```

## Summary

Determines whether the specified objects are equal.

## Parameters

Parameter	Description
<i>x</i>	First T to compare.
<i>y</i>	Second T to compare.

## Return Value

true if the specified objects are equal; otherwise, false.

## Description

An implementation of Equals(T,T) shall satisfy the following: The equality function shall be reflexive, so `x.Equals(x)` is true; symmetric, so `x.Equals(y)` and `y.Equals(x)`; and transitive, so `x.Equals(y)` and `y.Equals(z)` implies `x.Equals(z)`; for any values `x`, `y` and `z` for which these expressions are defined.

# IEqualityComparer<T>.GetHashCode(T) Method

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

## Summary

Returns a hash code for the specified object.

## Parameters

Parameter	Description
<i>obj</i>	The object for which the hash code is to be returned.

## Return Value

A hash code for the specified object.

## Description

To produce a hash function for the given object. [*Note:* A hash function is used to quickly generate a number (a hash code) corresponding to the value of an object. Hash functions are used with `hashtables`. A good hash function algorithm rarely generates hash codes that collide. For more information about hash functions, see *The Art of Computer Programming*, Vol. 3, by Donald E. Knuth.]

## Behaviors

All implementations are required to ensure that if `x.Equals(y) == true`, then `x.GetHashCode() equals y.GetHashCode()`, for any `x` and `y` values for which these expressions are defined.

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	The type of <i>obj</i> is a reference type and <i>obj</i> is <code>null</code> .

