

IComparer

What

- › The `System.Collections.Generic.IComparer` interface has a method called "Compare", which determines order of two objects i.e. current object and parameter object.
- › It can be implemented by a separate class to make the objects sortable.
- › It is useful to invoke **List.Sort** method to sort the collection of objects.
- › It is an alternative to `IComparable`; useful for the classes that doesn't implement `IComparable`.

```
interface System.Collections.Generic.IComparer
```

```
public interface IComparer<T>
{
    int Compare(T x, T y);
}
```

Return value

- › 0 : both x and y are equal; so will be kept in the same position.
- › <0 : x comes first; y comes next.
- › >0 : y comes first; x comes next.

```
interface System.Collections.Generic.IComparer
```

```
public class ClassName : IComparer<T>
{
    public int Compare(T x, T y)
    {
        return value;
    }
}
```

Implementation of IComparer interface

```
public class ClassName : IComparer<T>
{
    public int Compare(ClassName x, ClassName y)
    {
        if (x.field1 == y.field1)
            return 0; //equal
        else if (x.field1 < y.field1)
            return -1; //x object comes next.
        else
            return 1; //y object comes next.
    }
}
```