[0843] IComparable

The ICompareable interface allows you to specify the sort order for your Class' objects. Objects may or may not have a well-defined, natural sort order. By implementing ICompareable, you can define that order yourself.

Here's an example:

```
public class Card : ICloneable, IComparable
{
    public Rank Rank { get; private set; }
    public Suit Suit { get; private set; }

    public Card(Suit suit, Rank rank)
    {
        this.Suit = suit;
        this.Rank = rank;
    }

    public object Clone()
    {
        return this.MemberwiseClone();
    }

    int IComparable.CompareTo(object obj) {
        Card temp = obj as Card;
        if (temp != null)
        {
            if (this.Suit > temp.Suit) return 1;
            if (this.Suit < temp.Suit) return -1;

            // Suits are equal, so compare Ranks
            if (this.Rank > temp.Rank) return 1;
            if (this.Rank < temp.Rank) return -1;
            return 0;
        }
        throw new ArgumentException("Invalid comparison to Card. Object is not a Card.");
    }

    public override string ToString()
    {
        return ("[" + Rank + " of " + Suit + "]");
    }

}
```

In this case, Card objects are first sorted by suit. If the suit's are the same, then rank is used to determine the order.