Chapter 5: Equality Operator

Section 5.1: Equality kinds in c# and equality operator

In C#, there are two different kinds of equality: reference equality and value equality. Value equality is the commonly understood meaning of equality: it means that two objects contain the same values. For example, two integers with the value of 2 have value equality. Reference equality means that there are not two objects to compare. Instead, there are two object references, both of which refer to the same object.

```
object a = new object();
object b = a;
System.Object.ReferenceEquals(a, b);  //returns true
```

For predefined value types, the equality operator (==) returns true if the values of its operands are equal, false otherwise. For reference types other than string, == returns true if its two operands refer to the same object. For the string type, == compares the values of the strings.

```
// Numeric equality: True
Console.WriteLine((2 + 2) == 4);

// Reference equality: different objects,
// same boxed value: False.
object s = 1;
object t = 1;
Console.WriteLine(s == t);

// Define some strings:
string a = "hello";
string b = String.Copy(a);
string c = "hello";

// Compare string values of a constant and an instance: True
Console.WriteLine(a == b);

// Compare string references;
// a is a constant but b is an instance: False.
Console.WriteLine((object)a == (object)b);

// Compare string references, both constants
// have the same value, so string interning
// points to same reference: True.
Console.WriteLine((object)a == (object)c);
```