

Equality in .NET

Simon Robinson
<http://TechieSimon.com>
@TechieSimon



pluralsight 
hardcore dev and IT training



→ The virtual `object.Equals()` method.

→ The static `object.Equals()` method.

→ The static `object.ReferenceEquals()` method.

→ Strong typing and removing boxing:
- The `IEquatable<T>` interface.

→ The virtual **object**.Equals() method.

→ The static **object**.Equals() method.

→ The static **object**.ReferenceEquals() method.

→ The sta

→ Strong
- The

System.**Object**

static bool Equals()

virtual bool Equals()

static bool ReferenceEquals()

virtual int GetHashCode()

(and other methods)

Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

== and Equals()

In this module we'll only use .NET methods
to show how they work



== is provided by the C# language – not part of .NET!

.NET has no concept of operators

Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

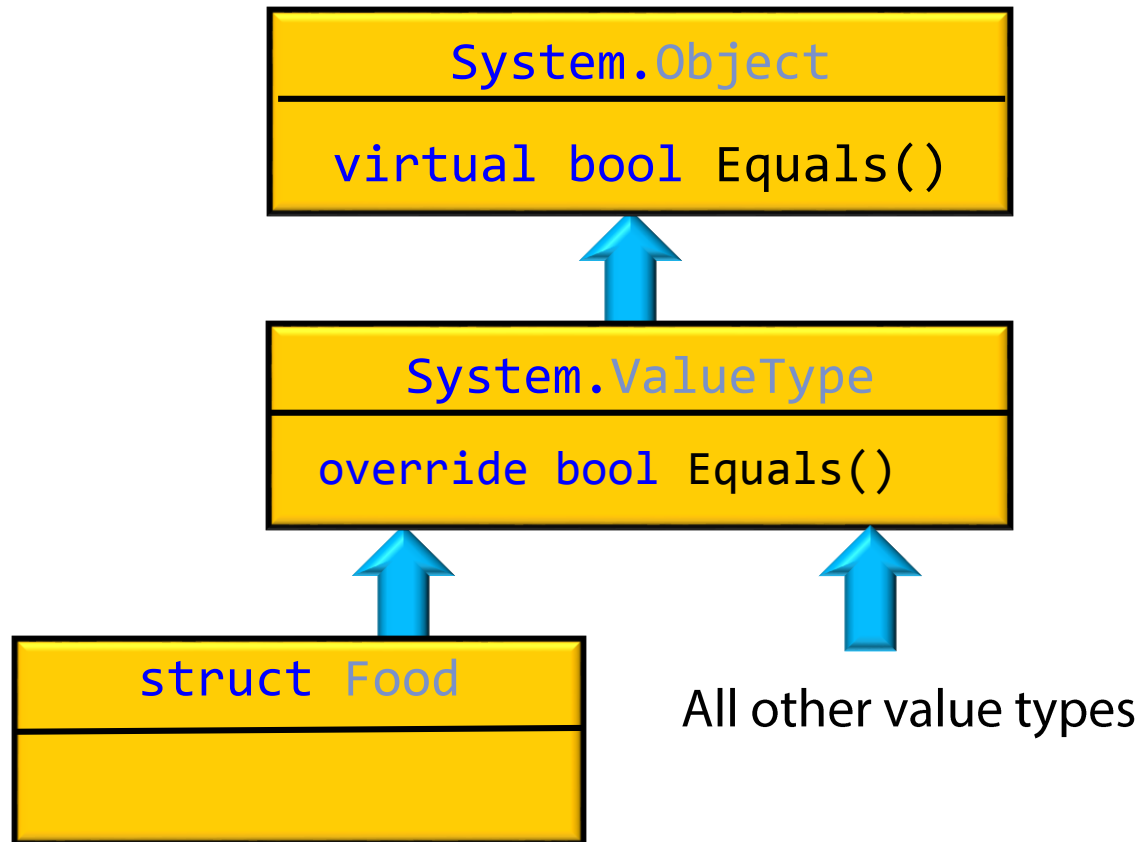
Code Demo

**Do Not Place Anything
in This Space**

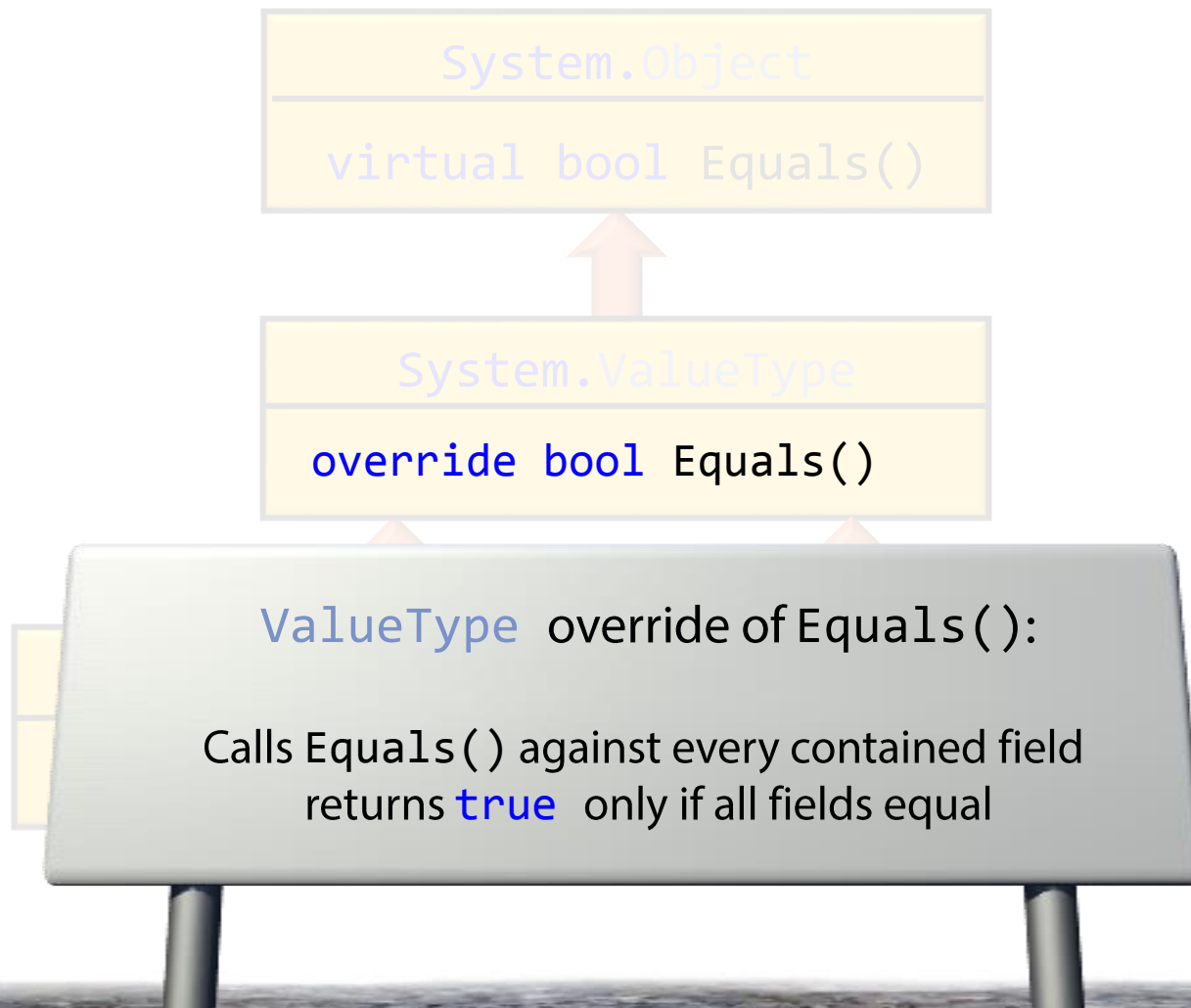
(Add watermark during
editing)

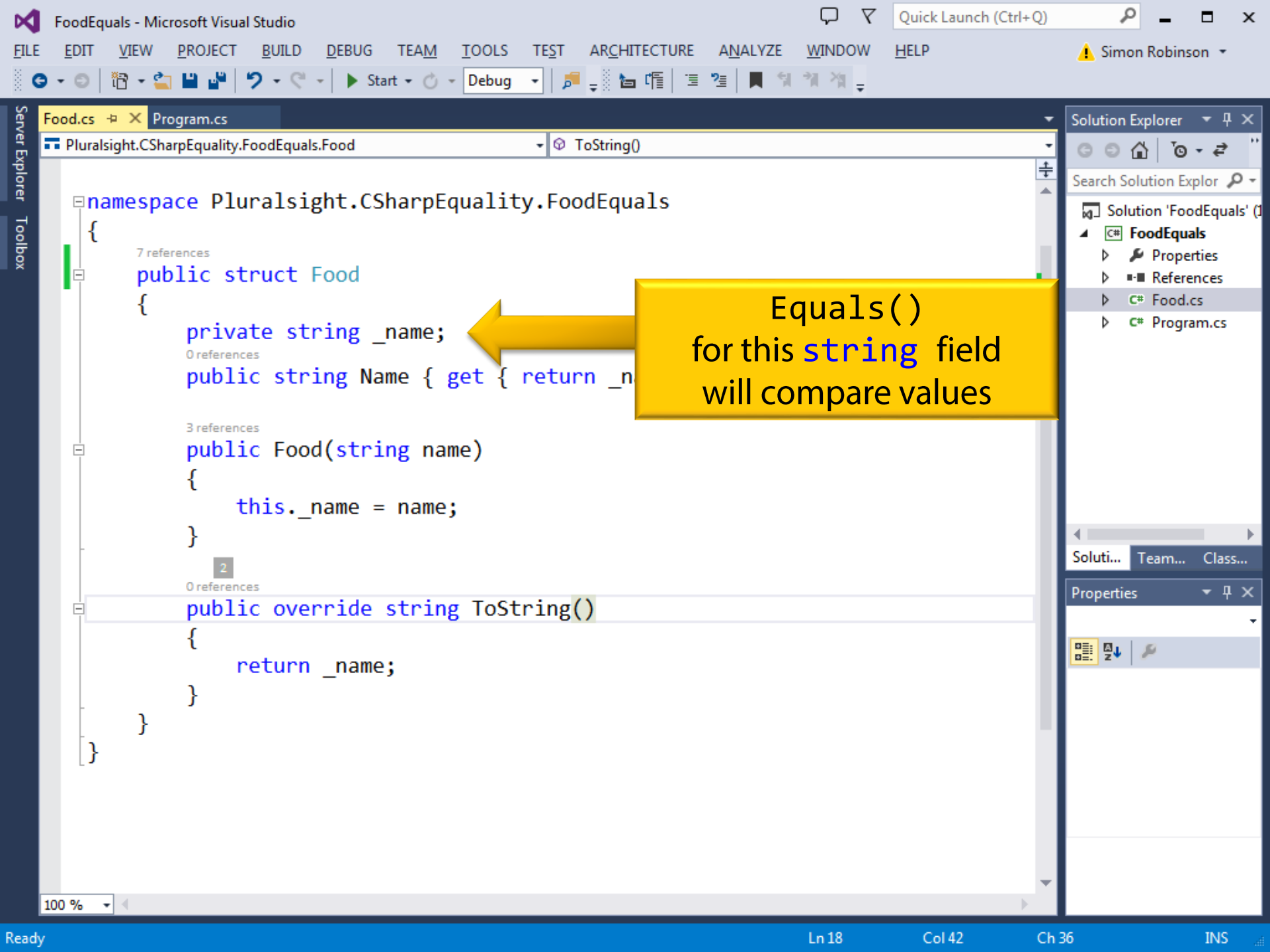
Note: Warning will not appear
during Slide Show view.

How Does Equals() Work for Structs?



How Does Equals() Work for Structs?





Food.cs Program.cs

Pluralsight.CSharpEquality.FoodEquals.Food

ToString()

```
namespace Pluralsight.CSharpEquality.FoodEquals
```

```
{
```

```
7 references
```

```
public struct Food
```

```
{
```

```
    private string _name;
```

```
0 references
```

```
    public string Name { get { return _name; } }
```

```
3 references
```

```
    public Food(string name)
```

```
{
```

```
        this._name = name;
```

```
}
```

```
2
```

```
0 references
```

```
    public override string ToString()
```

```
{
```

```
        return _name;
```

```
}
```

```
}
```

Equals()

for this **string** field
will compare values

Solution Explorer

Search Solution Explorer

Solution 'FoodEquals' (1)

C# FoodEquals

Properties

References

C# Food.cs

C# Program.cs

Solution Team Class

Properties

Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

ValueType.Equals() – the Problem...

```
System.Object  
-----  
virtual bool Equals()
```



```
System.ValueType  
-----  
override bool Equals()  
{  
    // uses reflection  
    // to find the fields  
}
```



Performance is poor

Best to override
for your own
value types
to avoid reflection

Guiding Principles

Reference Types

Value Types

`System.Object`
`virtual bool Equals()`

`System.ValueType`
`override bool Equals()`

Reference Equality
by default
(Same instance)

Any type can override the
default behaviour!

Value Equality
by default
(All fields are equal)

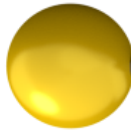


Other Equality Methods

`System.Object`

```
virtual bool Equals()  
static bool Equals()  
static bool ReferenceEquals()
```

`IEquatable<T>`



Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

What Static Equals() Does

```
public static bool Equals(  
{  
    if (obj1 == obj2)  
        return true;  
    if (obj1 == null || obj2 == null)  
        return false;  
    else  
        return obj1.Equals(obj2);  
}
```

Returns **true**
if both arguments are **null**
or they point to same instance

Returns **false**
if just one argument is **null**

Otherwise calls the
virtual method



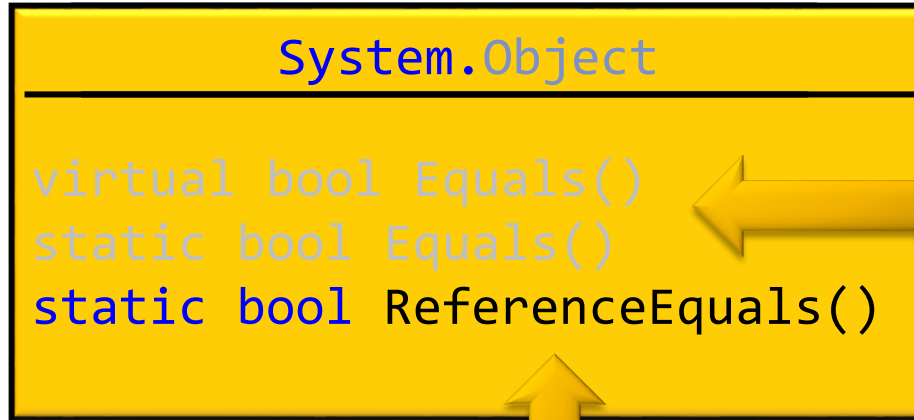
What Static Equals() Does

```
public static bool Equals(object obj1, object obj2)
{
    if (obj1 == obj2)
        return true;
    if (obj1 == null || obj2 == null)
        return false;
    else
        return obj1.Equals(obj2);
}
```



Static method gives same
results as the virtual method
(except for extra null checking)

ReferenceEquals()



Usually compare references
– But not if overridden

Used to check whether two
variables refer to the same
instance

Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Equals() and Typing

```
System.Object  
-----  
static bool Equals(object other)
```



Only parameter type that can work
with all types!



Value types will be boxed!

Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Equals(): Fixing the Problem

```
public class T : IEquatable<T>
{
    public bool Equals(T other) {
        // etc.
    }
}
```

IEquatable<T>



Type of objects being
compared!



This would solve boxing
and type safety!



But hard to make work
with inheritance



- MSDN Library
- .NET Development
- .NET Framework 4.5
- .NET Framework Class Library
- System
- IEquatable(T) Interface
 - IEquatable(T) Methods
 - Equals Method**

IEquatable<T>.Equals Method

.NET Framework 4.5 | Other Versions | 7 out of 10 rated this helpful - Rate this topic

Indicates whether the current object is equal to another object of the same type.

Namespace: System

Assembly: mscorlib (in mscorlib.dll)

Syntax

C# C++ F# VB

```
bool Equals(  
    T other  
)
```

No boxing!

Parameters

other

Type: T

An object to compare with this object.

Return Value

Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Primitive Types Implement IEquatable<T>

- MSDN Library
- .NET Development
- .NET Framework 4.5
- .NET Framework Class Library
- System

Int32 Structure

Int32 Structure

.NET Framework 4.5 | Other Versions

Represents a 32-bit signed integer.

To browse the .NET Framework source

Namespace: System

Assembly: mscorlib (in mscorlib.dll)

Syntax

C++ F# VB

```
[SerializableAttribute]  
[ComVisibleAttribute(true)]  
public struct Int32 : IComparable, IFormattable,  
    IConvertible, IComparable<int>, IEquatable<int>
```

Byte Structure

.NET Framework 4.5 | Other Versions | 2 out of 2 rated this helpful - Rating

Represents an 8-bit unsigned integer.

Namespace: System

Assembly: mscorlib (in mscorlib.dll)

Syntax

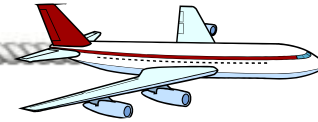
C# C++ F# VB

```
[SerializableAttribute]  
[ComVisibleAttribute(true)]  
public struct Byte : IComparable, IFormattable,  
    IConvertible, IComparable<byte>, IEquatable<byte>
```

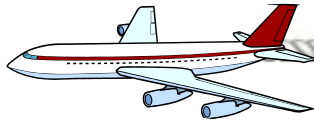
IEquatable<T>
- Very useful for
value types

IEquatable<T> Not Good for Reference Types

No boxing issue to fix!



Problems with inheritance



But
`IEquatable<T>`
is implemented by
`string`



Code Demo

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Equality Should Be Consistent!

```
public class T : IEquatable<T>
{
    public override bool Equals(object other) { ... }
    public bool Equals(T other) { ... }
    // etc.
}
```

Both methods should
do the same thing!



So if you implement `IEquatable<T>`,
then you should override
`object.Equals()` too.

Summary



Summary

- Several methods are defined on `object`.
 - Including `object.Equals()`.
- Reference equality for reference types,
value equality for value types.
- You can override `object.Equals()`.
 - Eg. strings, delegates, tuples.
- Static `Equals()` method to allow for null parameters.
- `IEquatable<T>` avoids boxing and gives type safety.