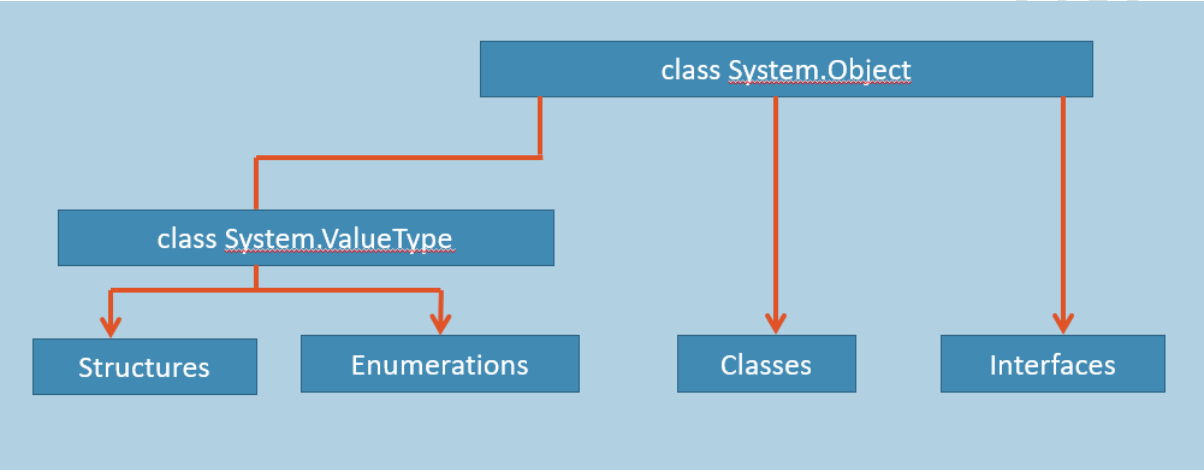
C# - Ultimate Guide - Beginner to Advanced | Master class

Section 15 - System.Object Class

System.Object class

The "System.Object" is a pre-defined class, which is the "Ultimate super class (base class)" in .net.

All the classes and other types are inherited from System.Object directly / indirectly.



All C# classes, structures, interfaces, enumerations are children of System.Object class.

Every method defined in the Object class is available in all objects in the system as all classes in the .NET Framework are derived from Object class.

Derived classes can override Equals, GetHashCode and ToString methods of Object class.

System.Object class is meant for achieving "type safety" in C#.

Methods of System.Object class

System.Object Class [Pre-defined]

```
namespace System
{
  class Object
  {
    virtual bool Equals( object value );
    virtual int GetHashCode( );
    Type GetType( );
    virtual string ToString( );
  }
}
```

› **bool Equals(object value)**

Compares the current object with the given argument object; returns true, if both are same objects; returns false, if both are different objects.

› **int GetHashCode(object value)**

Returns the a number that represents the object. It is not guarantee that the hash code is unique, by default.

› **Type GetType( )**

Returns the name of the class (including namespace path), based on which, the object is created.

› **string ToString( )**

By default, it returns the name of the class (including namespace path), based on which, the object is created.

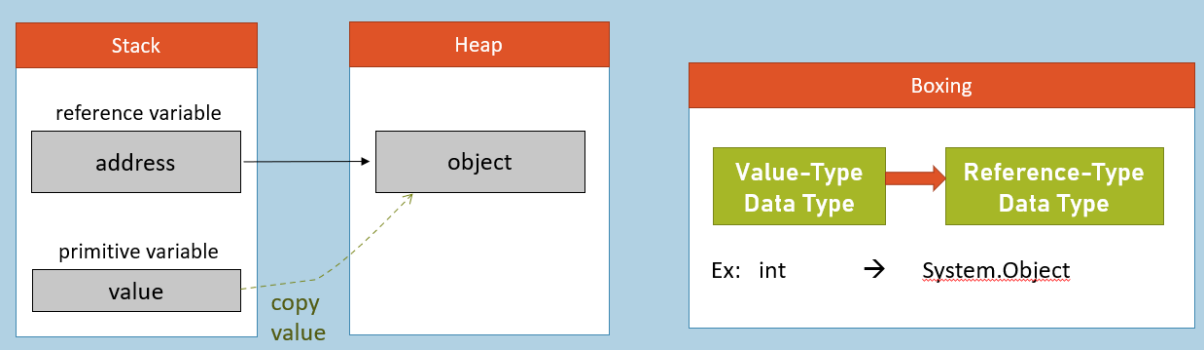
It is virtual method, which can be overridden in the child class.

Boxing & Unboxing

Boxing

It is a process of converting a value from "Value-Type Data Type" to "Reference-Type Data Type", if they are compatible data types.

This can be done automatically [no need of any syntax].



Unboxing

It is a process of converting a value from "Reference-Type Data Type" to "Value-Type Data Type", if they are compatible data types.

This should be done explicitly (by using explicit casting).

Syntax: (DestinationDataType)SourceValue

Example: (short)100