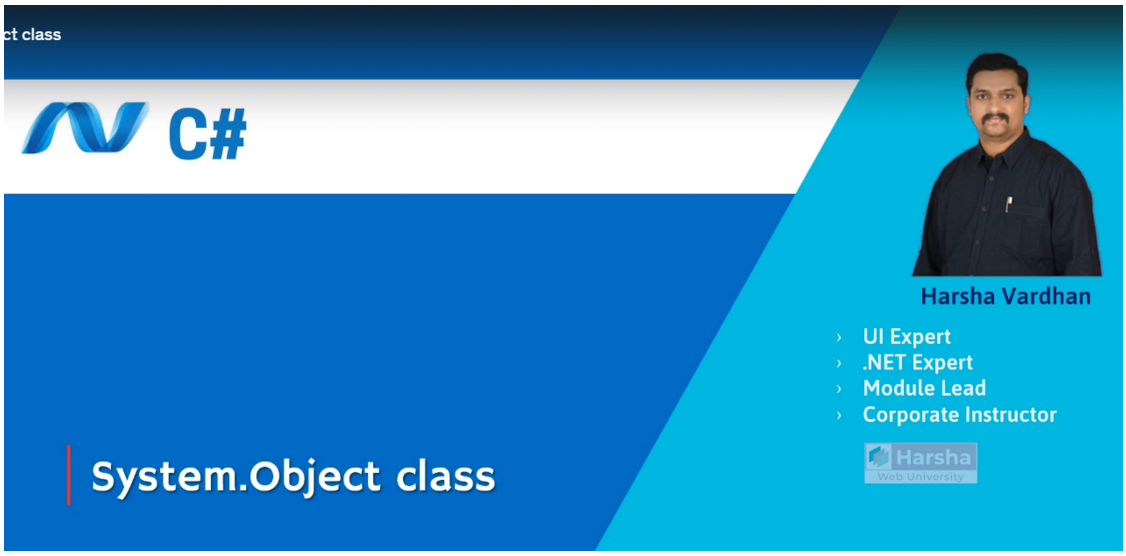
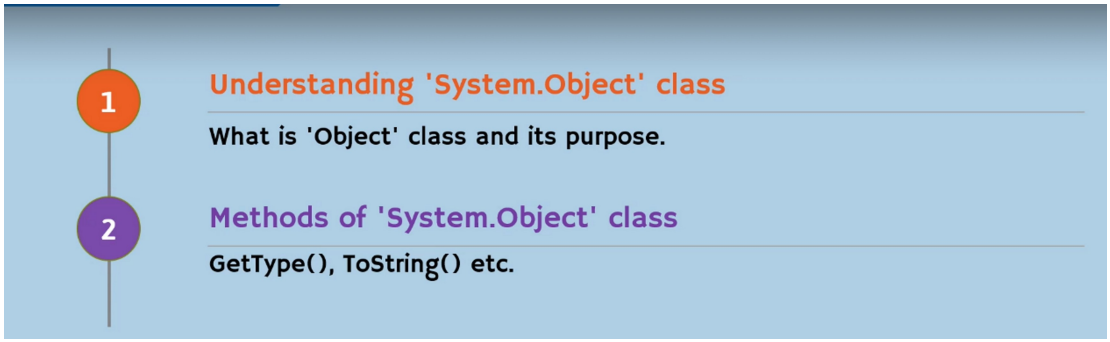
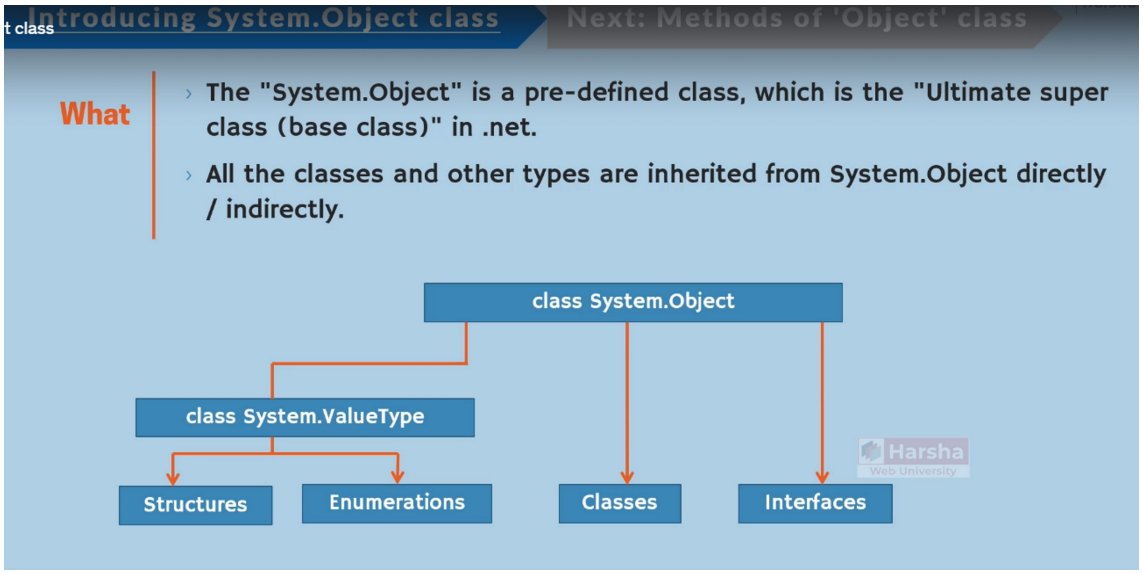
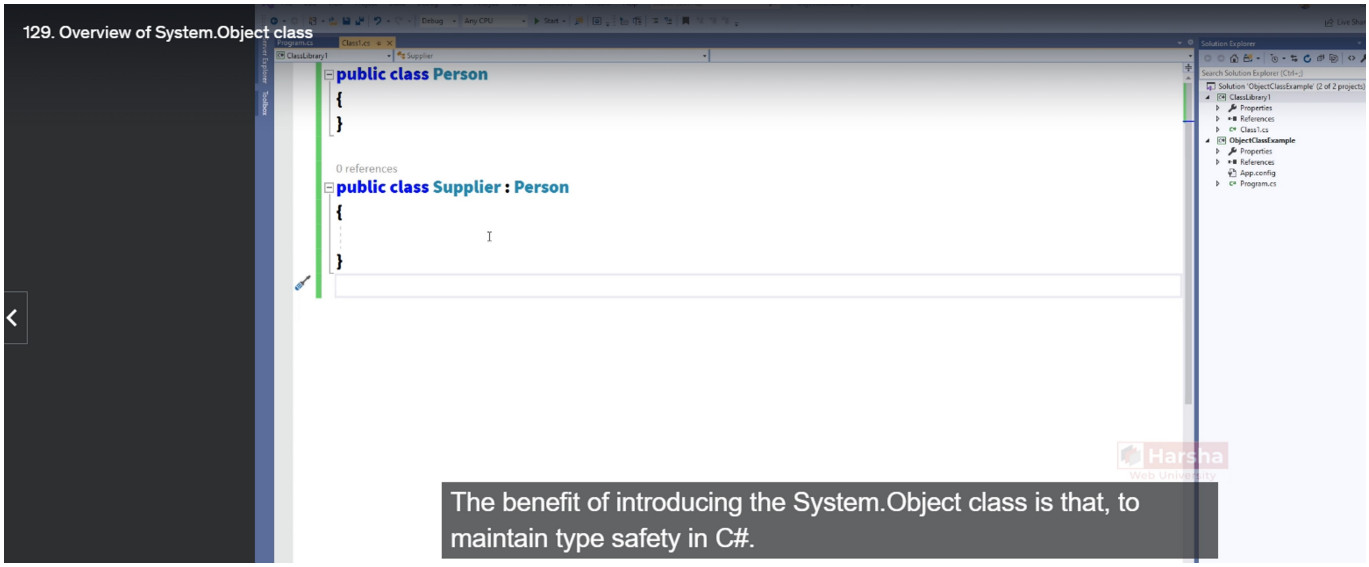
Sec 15 : System.Object Class

14 July 2024 16:29



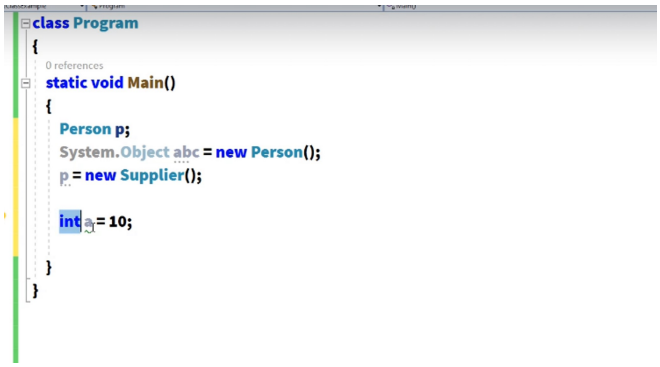


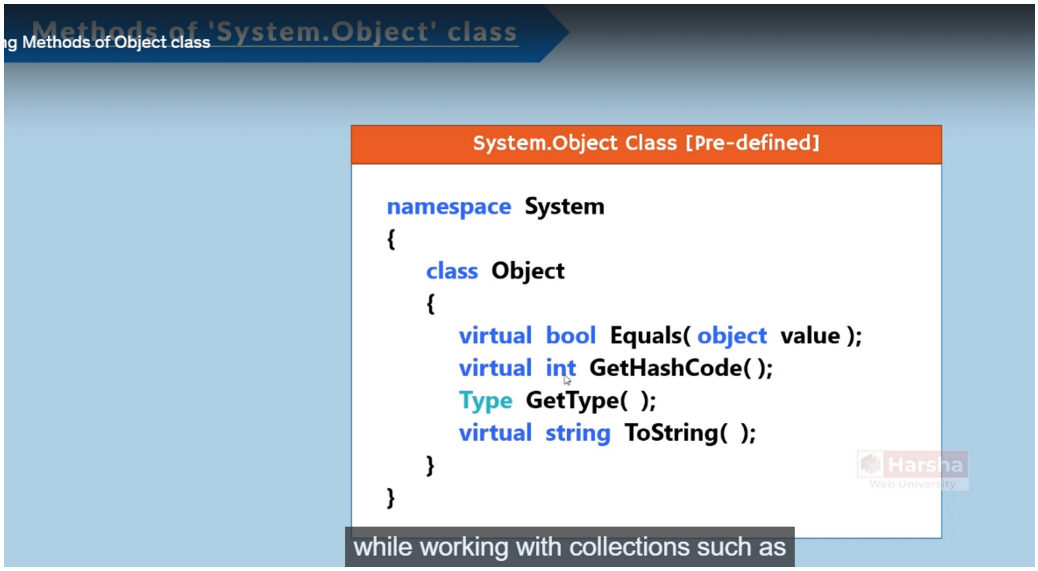


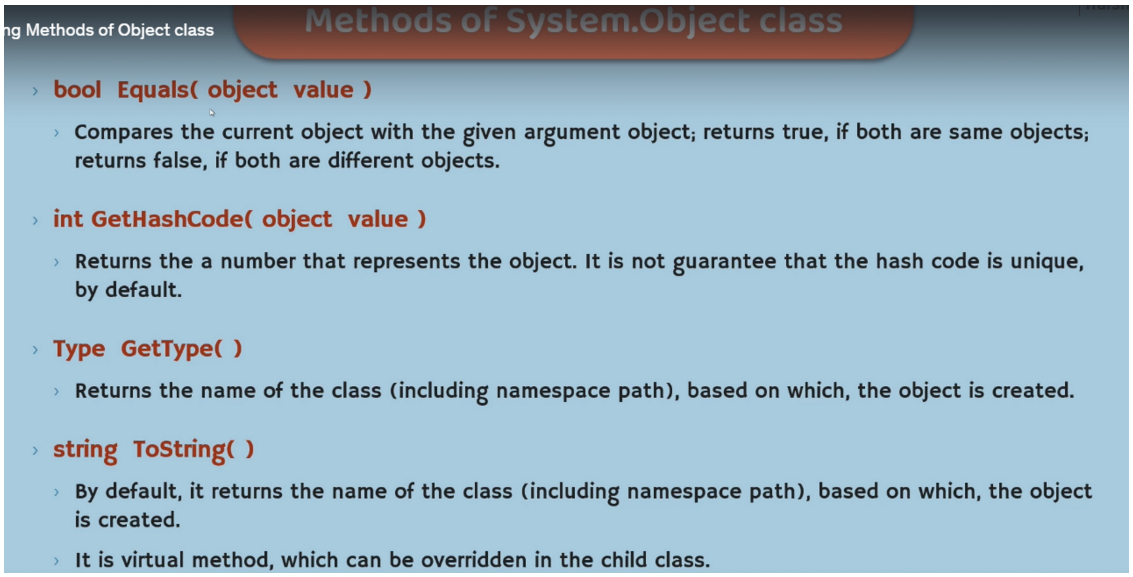


That means in a variable of particular type you must be able to assign only the values of the particular type. You should not be able to assign the values other than the particular type. To maintain this kind of type safety, the System.Object class is introduced in C#.

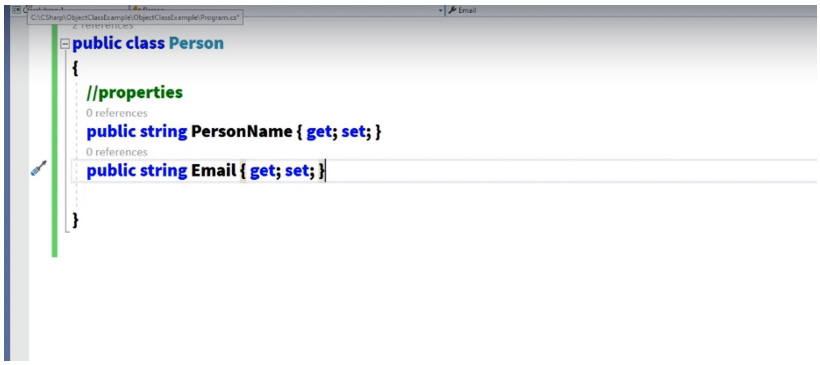
So all over in the entire language there is no any type which is not child of System.Object class. Because it must be child of System.Object class either directly or indirectly.

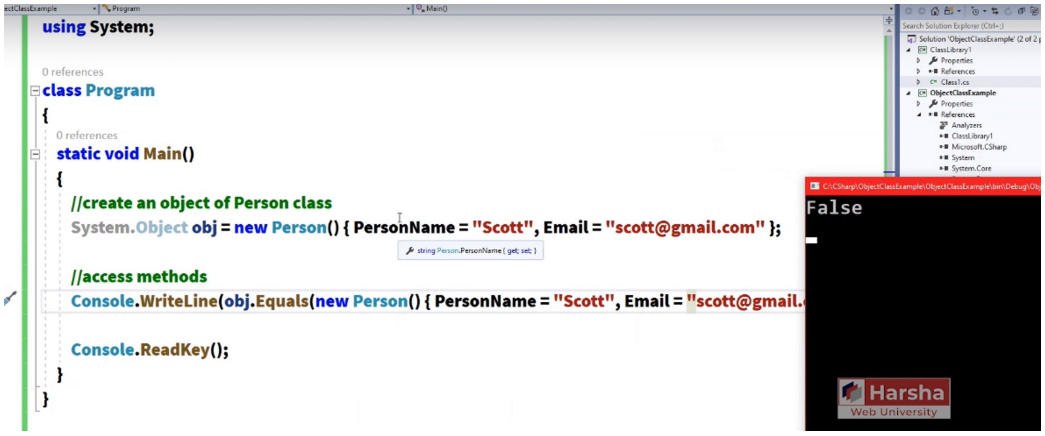




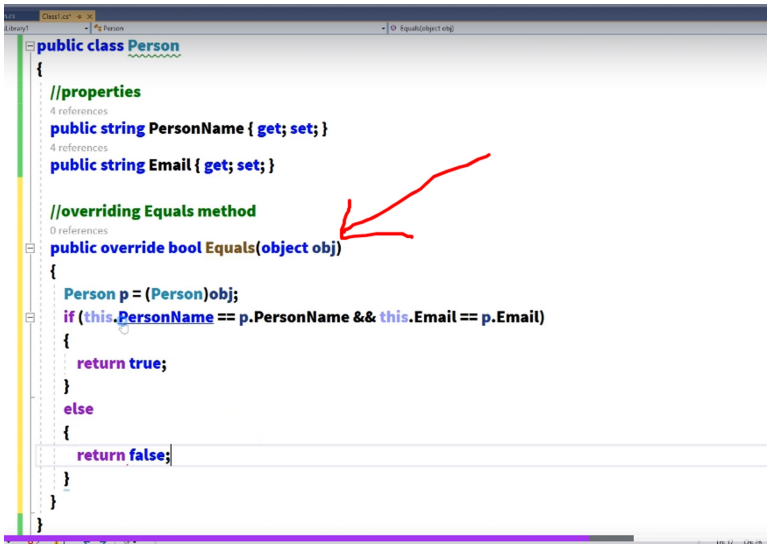


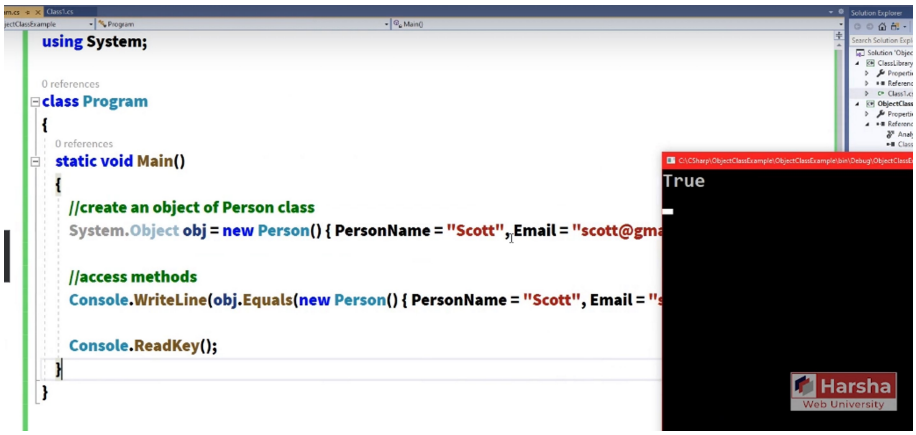
Understanding and Overriding Methods of Object Class

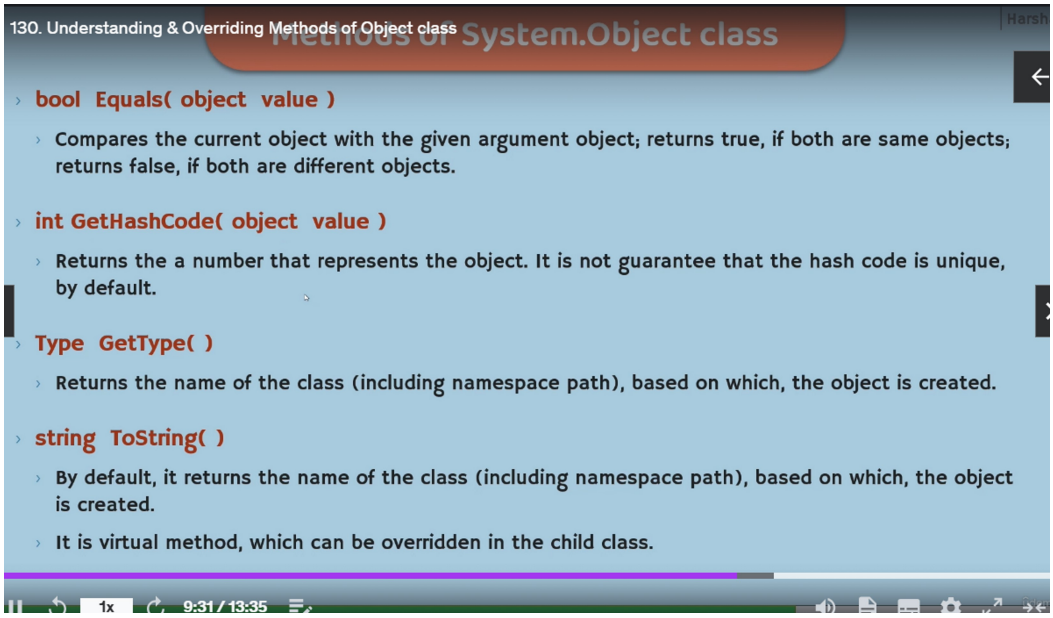


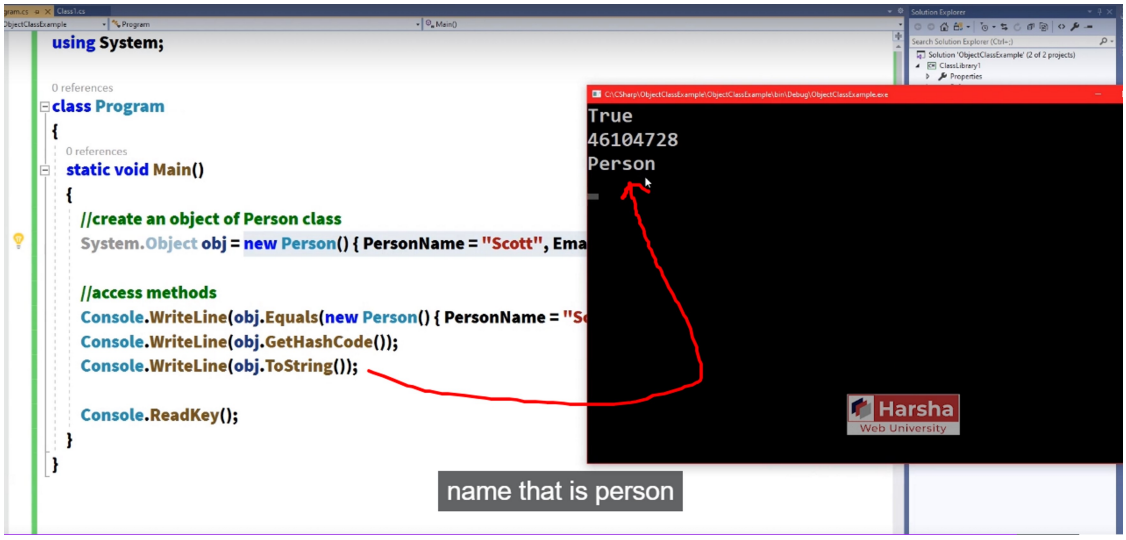


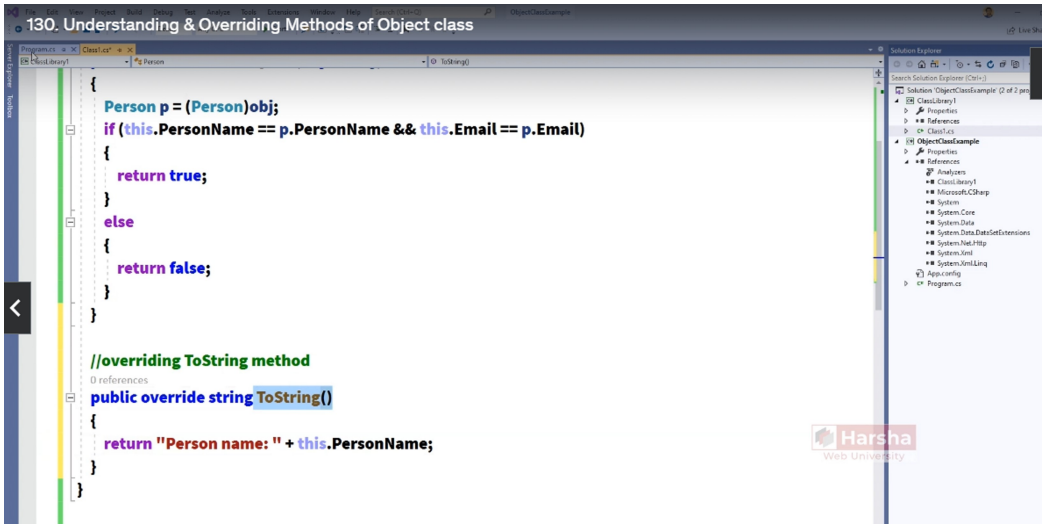
We're getting false as they are 2 separate object even though they have same value. but we can change it.

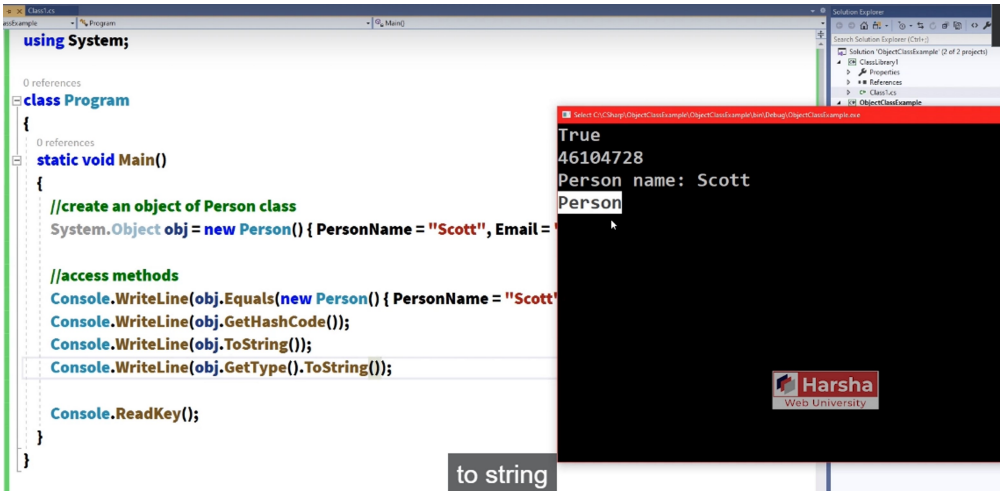




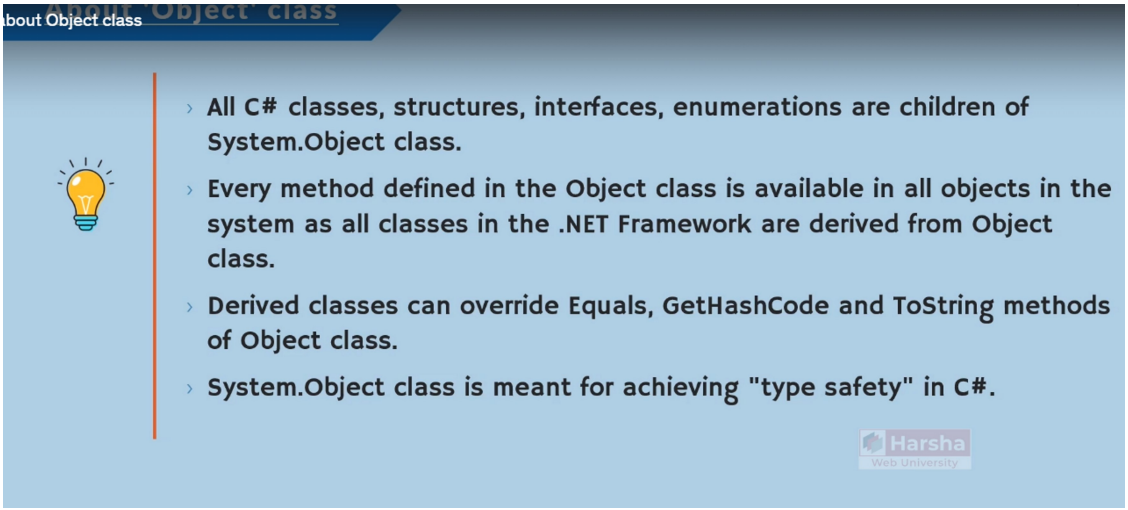

130. Understanding & Overriding Methods of Object class
Methods of System.Object class
bool Equals(object value)
Compares the current object with the given argument object; returns true, if both are same objects; returns false, if both are different objects.
int GetHashCode(object value)
Returns the 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.
1x
9:31 / 13:35


using System;
class Program
{
static void Main()
{
//create an object of Person class
System.Object obj = new Person() { PersonName = "Scott", Ema
//access methods
Console.WriteLine(obj.Equals(new Person() { PersonName = "S
Console.WriteLine(obj.GetHashCode());
Console.WriteLine(obj.ToString());
Console.ReadKey();
}
}
True
46104728
Person
Harsha
Web University
name that is person


130. Understanding & Overriding Methods of Object class
{
Person p = (Person)obj;
if (this.PersonName == p.PersonName && this.Email == p.Email)
{
return true;
}
else
{
return false;
}
}
//overriding ToString method
public override string ToString()
{
return "Person name: " + this.PersonName;
}
}
Harsha
Web University



ToString() can be overridden but GetType().ToString() always return class name and can not be overridden.



C#

Harsha Vardhan

- UI Expert
- .NET Expert
- Module Lead
- Corporate Instructor

Harsha Web University

Boxing & Unboxing

Agenda

1. **Boxing**
   Conversion from Value-type to Reference-type.
2. **Unboxing**
   Conversion from Reference-type to Value-type.

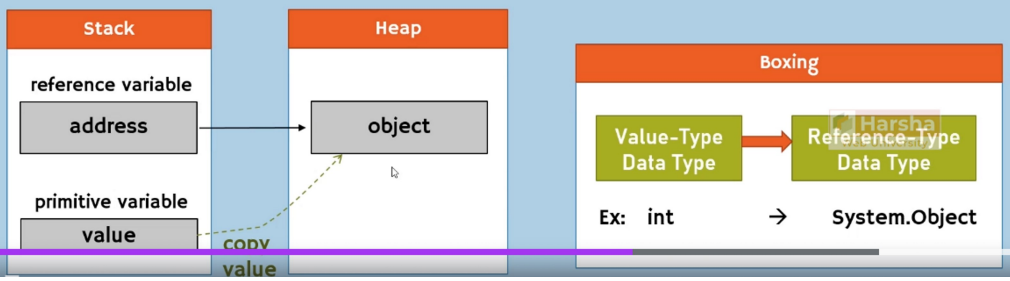
Boxing | Next: Unboxing

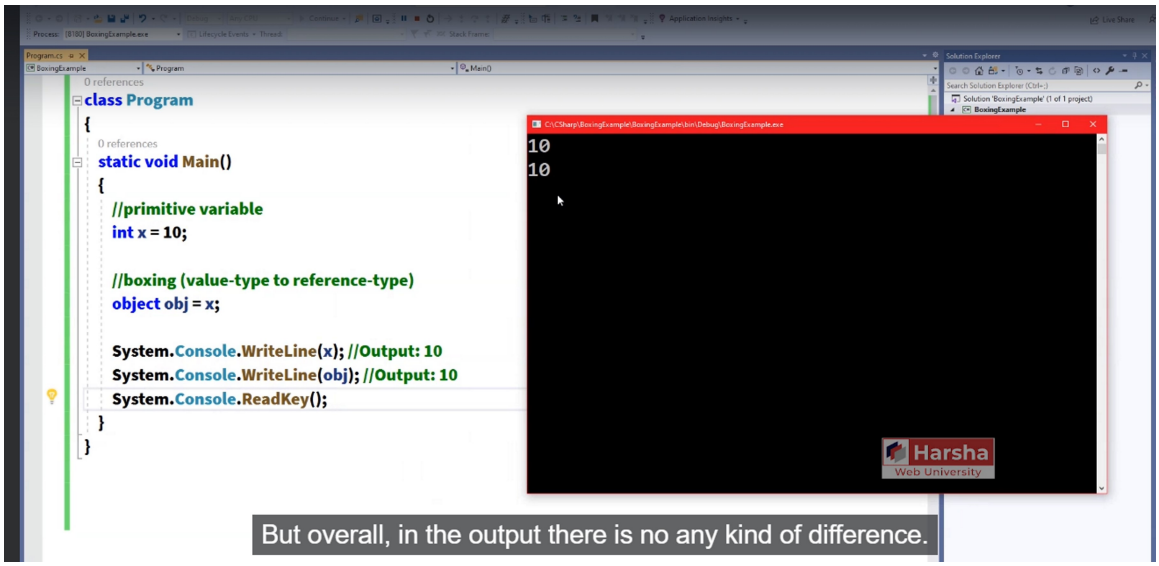
What

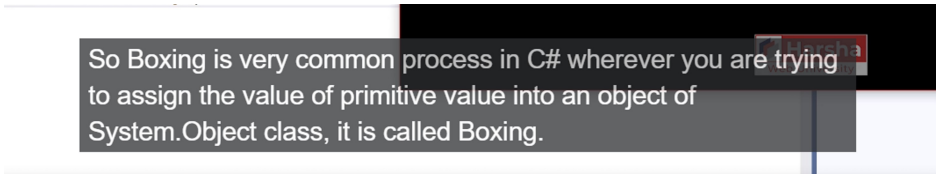
- It is a process of converting a value from "Value-Type Data Type" to "Reference-Type Data Type", if they are compatible data types.

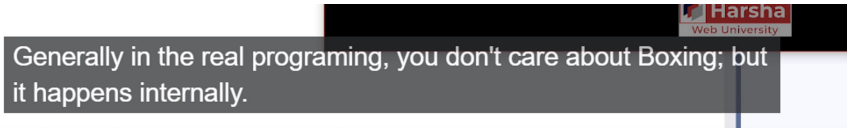
How

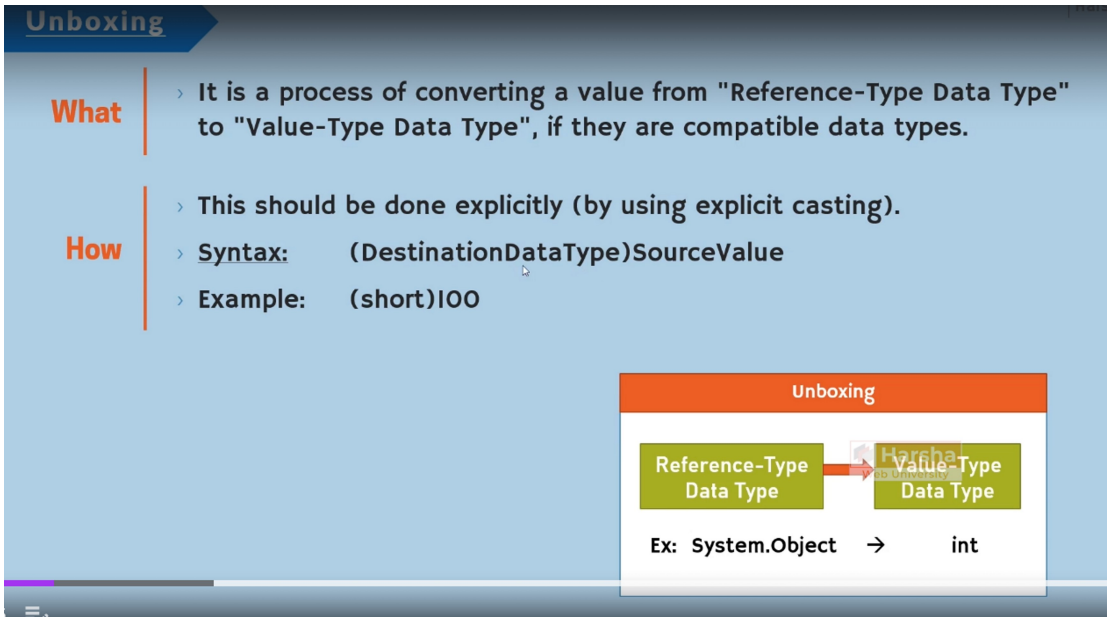
- This can be done automatically [no need of any syntax].

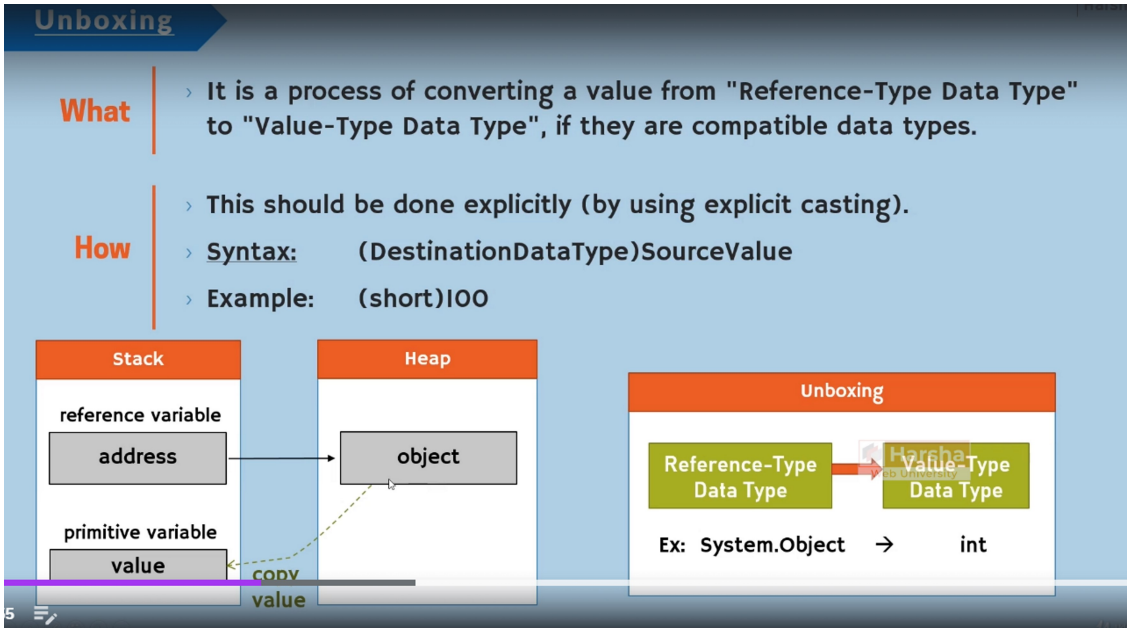



0 references
class Program
{
0 references
static void Main()
{
//primitive variable
int x = 10;
//boxing (value-type to reference-type)
object obj = x;
System.Console.WriteLine(x); //Output: 10
System.Console.WriteLine(obj); //Output: 10
System.Console.ReadKey();
}
}
10
10
Harsha
Web University
But overall, in the output there is no any kind of difference.


So Boxing is very common process in C# wherever you are trying to assign the value of primitive value into an object of System.Object class, it is called Boxing.


Harsha
Web University
Generally in the real programing, you don't care about Boxing; but it happens internally.


Unboxing
What
It is a process of converting a value from "Reference-Type Data Type" to "Value-Type Data Type", if they are compatible data types.
How
This should be done explicitly (by using explicit casting).
Syntax: (DestinationDataType)SourceValue
Example: (short)100
Unboxing
Reference-Type Data Type
Value-Type Data Type
Ex: System.Object → int



If the value has been copied from the heap into the stack, then it is called as Unboxing.