**Part 57 - C# Tutorial - Why should you override ToString() method**

**Why should you override the ToString() method?**  
All types in .Net inherit from system.object directly or indirectly. Because of this inheritance, every type in .Net inherit the ToString() method from System.Object class. Consider the example below.  
  
using System;  
public class MainClass  
{  
  public static void Main()  
  {  
   int Number = 10;  
   Console.WriteLine(Number.ToString());  
  }  
}  
  
In the above example Number.ToString() method will correctly give the string representaion of int 10, when you call the ToString() method.  
  
If you have a Customer class as shown in the below example and when you call the ToString() method the output doesnot make any sense. Hence you have to override the ToString() method, that is inherited from the System.Object class.  
  
using System;  
public class Customer  
{  
 public string FirstName;  
 public string LastName;  
}  
public class MainClass  
{  
 public static void Main()  
 {  
  Customer C = new Customer();  
  C.FirstName = "David";  
  C.LastName = "Boon";  
  Console.WriteLine(C.ToString());  
 }  
}

The code sample below shows how to override the ToString() method in a class, that would give the output you want.  
  
  
using System;  
public class Customer  
{  
  public string FirstName;  
  public string LastName;  
  
  public override string ToString()  
  {  
    return LastName + ", " + FirstName;  
  }  
}  
public class MainClass  
{  
  public static void Main()  
  {  
    Customer C = new Customer();  
    C.FirstName = "David";  
    C.LastName = "Boon";  
    Console.WriteLine(C.ToString());  
  }  
}  
  
**Conclusion :** If you have a class or a struct, make sure you override the inherited ToString() method.