**Part 48 - C# Tutorial - Difference between Types and Type Members**

**In this session we will**  
**1.** Understand the difference between Types and Type Members  
**2.** Understand Organising code with regions  
  
  
In the example below **Customer** is the Type and **private fields**(\_id, \_firstName, \_lastName), **Properties**(Id, FirstName, LastName) and GetFullName() **method** are type members.

public class Customer  
{  
    #region Private Fields  
    private int \_id;  
    private string \_firstName;  
    private string \_lastName;  
    #endregion  
  
    #region Properties  
    public int Id  
    {  
        get { return \_id; }  
        set { \_id = value; }  
    }  
    public string FirstName  
    {  
        get { return \_firstName; }  
        set { \_firstName = value; }  
    }  
    public string LastName  
    {  
        get { return \_lastName; }  
        set { \_lastName = value; }  
    }  
    #endregion  
  
    #region Methods  
    public string GetFullName()  
    {  
        return this.\_firstName + " " + this.\_lastName;  
    }  
    #endregion  
}

So, in general **classes, structs, enums, interfaces, delegates** are called as **types**and **fields, properties, constructors, methods** etc., that normally reside in a type are called as **type members.**  
  
  
**In C# there are 5 different access modifiers.**  
1. Private  
2. Protected  
3. Internal  
4. Protected Internal  
5. Public  
  
  
**Type members** can have all the access modifiers, where as **types** can have only 2 (internal, public) of the 5 access modifiers  
  
  
*In the next session we will discuss about all the access modifiers in detail.*

**Customer** class makes use of regions. Using **regions** you can expand and collapse sections of your code either manually, or using visual studio **Edit** -> **Outlining** -> **Toggle All Outlining**