**Part 36 - C# Tutorial - Delegates**

A **delegate**is a type safe function pointer.That is, they hold reference(Pointer) to a function.   
  
  
The **signature**of the delegate **must match** the signature of the function, the delegate points to, otherwise you get a compiler error. This is the reason delegates are called as type safe function pointers.  
  
  
A Delegate is similar to a class. You can create an instance of it, and when you do so, you pass in the function name as a parameter to the delegate constructor, and it is to this function the delegate will point to.  
  
  
***Tip to remember delegate syntax:****Delegates syntax look very much similar to a method with a delegate keyword.*  
  
  
**Sample Delegate Program:**  
  
  
using System;  
  
  
// Delegate Declaration.   
public delegate void HelloFunctionDelegate(string Message);  
  
  
class Pragim  
{  
    public static void Main()  
    {  
        // Create the instance of the delegate and pass in the function  
        // name as the parameter to the constructor. The passed in  
        // function signature must match the signature of the delegate  
        HelloFunctionDelegate del = new HelloFunctionDelegate(Hello);  
        // Invoke the delegate, which will invoke the method  
        del("Hello from Delegte");  
    }  
  
  
    public static void Hello(string strMessge)  
    {  
        Console.WriteLine(strMessge);  
    }  
}