**Part 39 - C# Tutorial - Multicast Delegates**

**A Multicast delegate is a delegate that has references to more than one function. When you invoke a multicast delegate, all the functions the delegate is pointing to, are invoked.**

**There are 2 approaches to create a multicast delegate.**

**Approach 1:**  
using System;  
namespace Sample  
{  
    public delegate void SampleDelegate();  
      
    public class Sample  
    {  
        static void Main()  
        {  
            SampleDelegate del1 = new SampleDelegate(SampleMethodOne);  
            SampleDelegate del2 = new SampleDelegate(SampleMethodTwo);

  SampleDelegate del3 = new SampleDelegate(SampleMethodThree);

    // In this example del4 is a multicast delegate. You use +(plus)   
            // operator to chain delegates together and -(minus) operator to remove.  
            SampleDelegate del4 = del1 + del2 + del3 - del2;  
              
            del4();  
        }  
  
  
        public static void SampleMethodOne()  
        {  
            Console.WriteLine("SampleMethodOne Invoked");  
        }  
  
  
        public static void SampleMethodTwo()  
        {  
            Console.WriteLine("SampleMethodTwo Invoked");  
        }  
  
  
        public static void SampleMethodThree()  
        {  
            Console.WriteLine("SampleMethodThree Invoked");  
        }  
    }  
}

**Approach 2:**  
using System;  
namespace Sample  
{  
    public delegate void SampleDelegate();  
      
    public class Sample  
    {  
        static void Main()  
        {  
            // In this example del is a multicast delegate. You use += operator   
            // to chain delegates together and -= operator to remove.

     SampleDelegate del = new SampleDelegate(SampleMethodOne);  
            del += SampleMethodTwo;  
            del += SampleMethodThree;  
            del -= SampleMethodTwo;  
              
            del();  
        }  
  
  
        public static void SampleMethodOne()  
        {  
            Console.WriteLine("SampleMethodOne Invoked");  
        }  
  
  
        public static void SampleMethodTwo()  
        {  
            Console.WriteLine("SampleMethodTwo Invoked");  
        }  
  
  
        public static void SampleMethodThree()  
        {  
            Console.WriteLine("SampleMethodThree Invoked");  
        }  
    }  
}  
  
  
***Note:*** A multicast delegate, invokes the methods in the invocation list, in the same order in which they are added.  
  
  
**If the delegate has a return type other than void and if the delegate is a multicast delegate, only the value of the last invoked method will be returned. Along the same lines, if the delegate has an out parameter, the value of the output parameter, will be the value assigned by the last method.**

**Example:** Multicast delegate with an int return type  
using System;  
namespace Sample  
{  
    // Deletegate's return type is int  
    public delegate int SampleDelegate();  
      
    public class Sample  
    {  
        static void Main()  
        {  
            SampleDelegate del = new SampleDelegate(SampleMethodOne);  
            del += SampleMethodTwo;  
  
  
            // The ValueReturnedByDelegate will be 2, returned by the SampleMethodTwo(),  
            // as it is the last method in the invocation list.  
            int ValueReturnedByDelegate = del();  
  
  
            Console.WriteLine("Returned Value = {0}", ValueReturnedByDelegate);  
        }  
  
  
        // This method returns one  
        public static int SampleMethodOne()  
        {  
            return 1;              
        }  
  
  
        // This method returns two  
        public static int SampleMethodTwo()  
        {  
            return 2;  
        }  
    }  
}  
  
  
**Example:** Multicast delegate with an integer output parameter.  
using System;  
namespace Sample  
{  
    // Deletegate has an int output parameter  
    public delegate void SampleDelegate(out int Integer);  
      
    public class Sample  
    {  
        static void Main()  
        {  
            SampleDelegate del = new SampleDelegate(SampleMethodOne);  
            del += SampleMethodTwo;  
  
  
            // The ValueFromOutPutParameter will be 2, initialized by SampleMethodTwo(),  
            // as it is the last method in the invocation list.  
            int ValueFromOutPutParameter = -1;  
            del(out ValueFromOutPutParameter);  
  
  
            Console.WriteLine("Returned Value = {0}", ValueFromOutPutParameter);  
        }  
  
  
        // This method sets ouput parameter Number to 1  
        public static void SampleMethodOne(out int Number)  
        {  
            Number = 1;  
        }  
  
  
        // This method sets ouput parameter Number to 2  
        public static void SampleMethodTwo(out int Number)  
        {  
            Number = 2;  
        }  
    }  
}  
  
  
**Where do you use multicast delegates? This is a very common interview question.**  
**Answer:**Multicast delegate makes implementation of observer design pattern very simple. Observer pattern is also called as publish/subscribe pattern.