Events and Delegates :-

In C#, events and delegates are fundamental concepts used for implementing the observer design pattern, enabling communication between objects in an application. Here's an overview of each:

Delegates:

Delegates are a type-safe, object-oriented function pointers that allow methods to be passed as parameters to other methods. They define the signature and return type of the method(s) they can reference.

Declaration:

delegate returnType DelegateName(parameters);

public delegate void MyDelegate(string message);

Events:

Events are special types of delegates that encapsulate methods to be called when a certain action or notification occurs. They are used to implement the publisher-subscriber pattern.

Declaration:

public event DelegateName EventName;

public event EventHandler ButtonClicked;

Basic Usage:

Define Delegate: Define a delegate that matches the signature of the methods you want to encapsulate.

Declare Event: Declare an event based on the delegate.

Subscribe to Event: Subscribe methods to the event using the += operator.

Raise Event: Call the event, which will execute all subscribed methods.

```
using System;

public class Program
{
    // Step 1: Define delegate
    public delegate void EventHandler(string message);

    // Step 2: Declare event
    public event EventHandler Notify;

    // Step 3: Subscribe to event
    public void Subscribe(EventHandler method)
    {
        Notify += method;
    }
```

```
        // Step 4: Raise event
        public void RaiseEvent(string message)
        {
            Notify?.Invoke(message);
        }

        static void Main(string[] args)
        {
            Program program = new Program();

            // Subscribe methods to
            event
            program.Subscribe(Metho
            d1);
            program.Subscribe(Metho
            d2);

            // Raise event
            program.RaiseEvent("Event
            triggered!");

            Console.ReadLine();
        }

        static void Method1(string message)
        {
            Console.WriteLine("Method 1: " + message);
        }

        static void Method2(string message)
        {
            Console.WriteLine("Method 2: " + message);
        }
    }
```

This will output:

mathematica

Method 1: Event triggered!
Method 2: Event triggered!

Events and delegates are extensively used in C# for implementing various patterns like observer, command, etc., as well as in frameworks like Windows Forms and WPF for handling UI events.