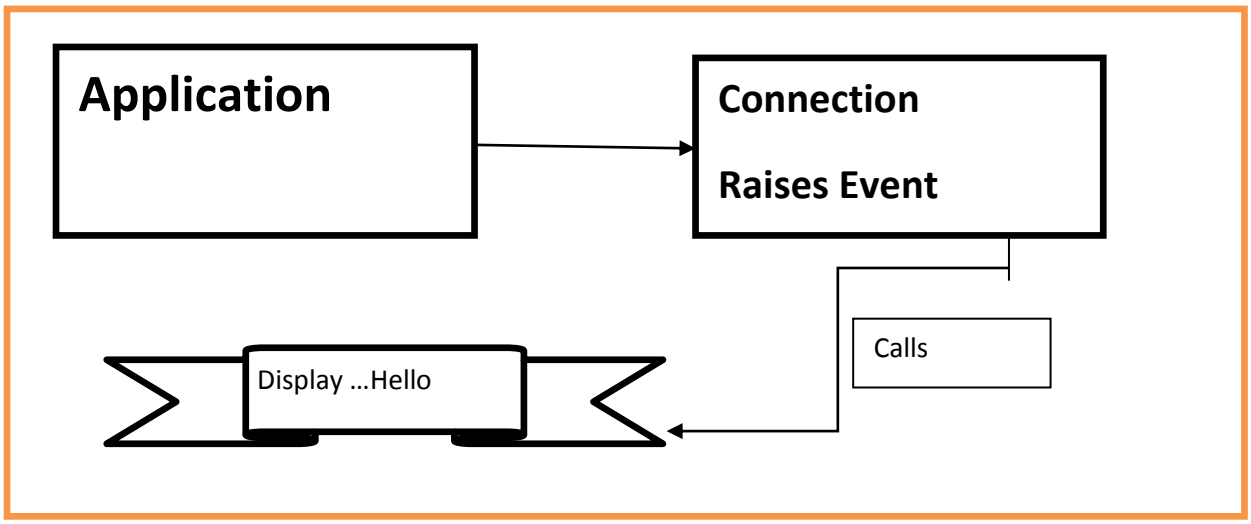
EVENTS IN C#

In this article I am going to explain you the most frequently used OOP techniques in .Net: events.

What is an Event?

Event are similar to the exceptions in that they are raised (thrown) by objects, and you can supply code that act on them. However, there are several important differences, the most important of which is that there is no equivalent to try…catch structure for handling events. Instead you must subscribe to them. Subscribing events mean providing code that must execute when event raised, in the form of event handler. Many event handlers can subscribe to a single event. The fact that delegates are used in events is one of the reasons why delegates are so useful.

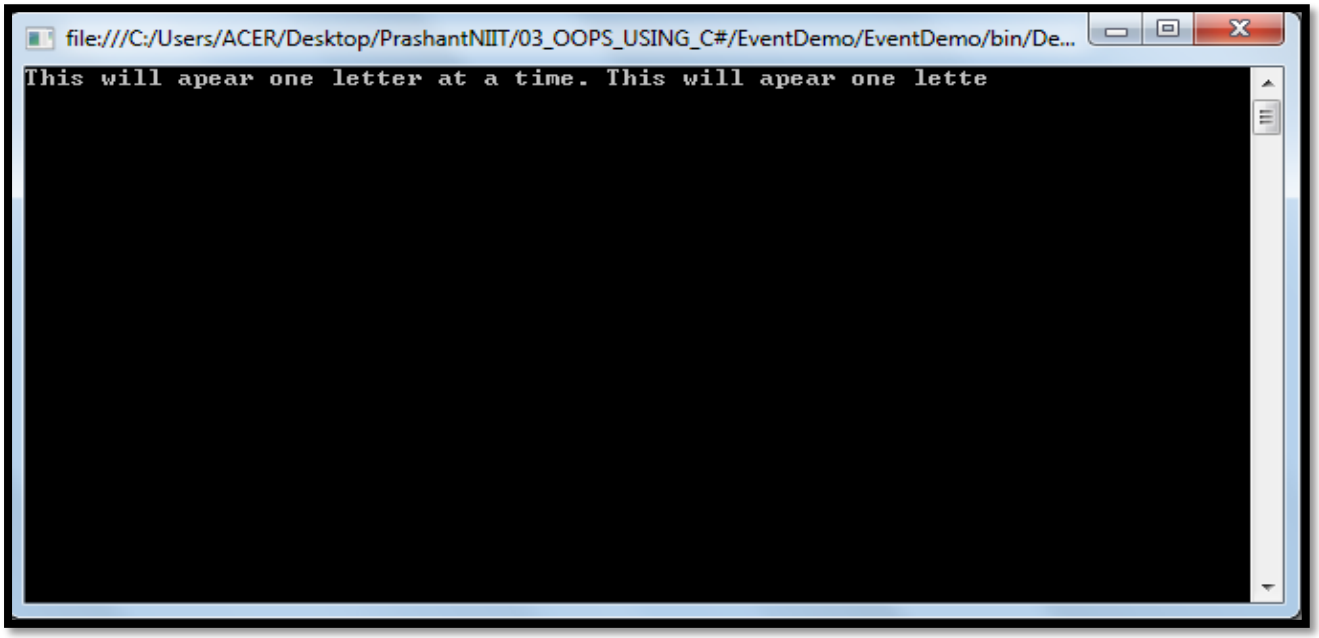


Handling Events

As previously discussed, to handle an event you need to subscribe to it by providing an event handler method whose return type and parameter match that of delegates specified for use with an event. The following example uses a simple timer object to raise events, which results in a handler method being called.

```
using System;
using System.Timers;
namespace EventDemo
{
    class Program
    {
        static int counter = 0;
        static string displayString = "This will apear one letter at a time. ";
        static void Main(string[] args)
        {
            Timer myTimer = new Timer(1000);
            myTimer.Elapsed += new ElapsedEventHandler(Writechar);
            myTimer.Start();
            Console.ReadKey();
        }
        static void Writechar(object source, ElapsedEventArgs e)
        {
            Console.Write(displayString[counter++ % displayString.Length]);
        }
    }
}
```

Run the application. After short time you will get following result (printing one character at time on console.)



How it works

The object you are using to raise events is an instance of the System.Timers.Timer class. This object gets initialized with a time period (in milliseconds). When the Timer objects is started using start() method, a stream of events is raised, spaced out in time according to the specified time period.

What is next?

In my next article of C# Event we will see how to define and use your own events.

Thank You

Prashant H Nimbare

Faculty @ NIIT