**C# : Using Asynchronous Delegate**

In this la we will implement the asynchronous delegate feature. Delegate is a type which provides various features as below:

- **Executing method with its reference.**
- **Declaring Events.**
- **Performing Asynchronous Execution for Long running operations.**

Task 1: Open VS2010 and create a new console application name it as 'CS_Async_Delegate'. Add the below class in the Program.cs. This class performs string operations. Some of the operations takes less time and some of them are time consuming:

```
/// <summary>
  /// class for String Operations
  /// </summary>
  public class StringOperations
  {
      public int CalculateStringLenght(string str)
      {
          return str.Length;
      }
      public string ConvertStrintToUpper(string str)
      {
          return str.ToUpper();
      }

      /// <summary>
      /// More Completed Operations here
      /// </summary>
      /// <param name="str"></param>
      /// <param name="c"></param>
      /// <returns></returns>
      public int GetMatchWordCount(string str,char c)
      {
          int charCount = 0;
          foreach (char ch in str)
          {
              if (ch == c)
              {
                  charCount++;
              }
          }
```

```
            return charCount;
        }
    }
```

**Task 2:** At the namespace level declare the below delegate:

```
public delegate int StringHandler(string str, char c);
```

**Task 3:** In the Main method add the below code. This code shows that the method 'GetMatchWordCount' from the class created above:

```
static void Main(string[] args)
        {
            StringOperations objString =new StringOperations() ;

            //Pass the method reference to the delegate
            StringHandler handler = new
StringHandler(objString.GetMatchWordCount);

            string str = ".NET is an excellent technology
platform for developing new generation ";
            str +="Applications. It is now a days more powerfull
for RICH UI, WEB, and mobile applications";

            //Start Asynchronous operations.
            IAsyncResult ar =  handler.BeginInvoke(str, 'a',
null, null);

            Console.WriteLine();
            while (!ar.IsCompleted)
            {
                Console.WriteLine("Lenght of " + str + " is = " +
objString.CalculateStringLenght(str));
                Console.WriteLine();
                Console.WriteLine("Converting to Upper case");

Console.WriteLine(objString.ConvertStrintToUpper(str));
            }

            Console.WriteLine();
            Console.WriteLine();
```

```
            //Complete the asynchronousn operation using
IAsyncResult object (ar)
            int charCount = handler.EndInvoke(ar);
            Console.WriteLine("Number of 'a' in string " + str +
" is =" + charCount);
 1
            Console.ReadLine();
        }
```

Run the application, you will find that the methods called in the while loop gets executed and then the GetMatchWordCount () gets executed.