Web Forms

ASP.NET

Active Server Pages (.asp)

- Used before ASP.NET and may still be in use.
- Merges the HTML with scripting on the server.
- Easier than CGI.
- Performance is poor since code is interpreted each time the page is loaded.
- No real component model.
- Relies on HTML forms, client side scripting, and embedded script.
- Other implementations such as PHP.

ASP Example – calc.asp

(Visual Basic)

```
<%@ Language="VBScript" %>

<html>
  <body>
    <form>
      <input type="text" name="op1" value="<%= Request ("op1") %>"/>
      +
      <input type="text" name="op2" value="<%= Request ("op2") %>" />
      <input type="submit" value="  =  " />
      <%
        If Request ("op1") <> "" And Request ("op2") <> "" Then
            a = CInt (Request ("op1"))
            b = CInt (Request ("op2"))
            Response.Write (CStr (a + b))
        End If
      %>
    </form>
  </body>
</html>
```

DEMO

Denoting Script in an Active Server Page

- Use the Script tag.
  <Script Language="C#" runat="server">
  ---program code---
  </Script>
- Small segments of script can be enclosed with <% and %>
  - Language must be specified for the whole page using <%@ Language = "…" %> tag.

ASP.NET Web Forms

- Use the extension .aspx rather than .asp.
- Can use C#.
- Has full object oriented model for controls virtually identical to Windows Forms.
- Code can be in a separate file and compiled. (code behind)
  - Better performance

ASP.NET Example - calc.aspx

```
<html>
  <body>
    <form runat="server">
      <asp:TextBox ID="op1" RunAt="server" />
      +
      <asp:TextBox ID="op2" RunAt="server" />
      <asp:Button Text="  =  "OnClick="OnAdd" RunAt="server" />
      <asp:Label ID="Sum" RunAt="server" />
    </form>
  </body>
</html>

<script language="C#" runat="server">
  void OnAdd (Object sender, EventArgs e)
  {
      int a = Convert.ToInt32 (op1.Text);
      int b = Convert.ToInt32 (op2.Text);
      Sum.Text = (a + b).ToString ();
  }
</script>
```

DEMO

Deficiencies (in this program)

- No error checking.
  Try entering a non-integer
- We can cause a run time exception.
- A try/catch block should be added and feedback to the user given for errors. We will do this when we start using code behind.

Running ASP.NET Applications

- Demo – implement calc using code behind
  - Create in VS2017
  - Run using built in web server - IIS Express
  - IIS Express is new replacement for the old development server which has been removed
  - View source in browser
- In order to run an ASPX file under IIS you must make the folder an application directory. (Advanced Topic)

ASP.NET Application Models

- C# source is compiled to an intermediate language (MSIL)
- The MSIL is compiled to native code when it is accessed (JIT)
- Two options in VS 2017
  - Web application project (precompiled code)
  - Web site (compiled on demand)
- We will use the second option for EC512

Differences from ASP

- Form fields are now represented by actual C# controls.
- Use runat="server" to indicate that a form field executes on the server.
  - HTML sent to browser looks like regular form fields. (DEMO)
- Use properties and events to manipulate the controls on the server side.
- Controls persist their values across postbacks. (magic of asp.net)

Postbacks

- The same .aspx page bounces back and forth between the server and client.
- The server side code is able to easily determine if the page is executing initially or on behalf of a postback.
  - The IsPostBack property can be used to find out.
- A hidden field passes state information from the browser back to the server.

Web Forms Programming Model

- Design the user interface as a combination of HTML and server controls.
- Server controls fire events that are processed by server side handlers.

  There is the illusion that events are fired and handled on the same machine (client side).
- Server side programming is compiled and run under the .NET CLR. (JIT)

Controls

- Two families of controls
  1. Web controls

     Designed for .NET. More robust and more of them.
  2. HTML controls

     Designed to be compatible with HTML form fields. **Allow for retrofitting existing HTML pages for ASP.NET with less editing. <u>We will not be using them.</u>**
- Look in the toolbox to see what controls are available.

Page Level Events

- Derived from System.Web.UI.Page class
- Page.Init event is fired when the page is being created. This is where the designer initializes the components.
- Page.Load event is fired After Page.Init but before the page is sent to the browser. This is where to customize the contents of controls etc. VS 2017 generates an empty method for you.
- Note – Page.Init is implemented as an override whereas Page.Load is implemented as an event handler. (See VS2017 code generated)

Learning C#

- For those that do not know C# these are critical concepts to learn:
  1. Reference vs. value types
  2. Inheritance
  3. Polymorphism
     - Overrides
     - Virtual methods
  4. Use of the FCL
  5. Concept of namespaces and assemblies
  6. Exceptions and how to handle them