M11 L1
Web Computing Models

Ira A. Fulton Schools of Engineering
Arizona State University

Lecture Outline

- **Different Web Computing Models**
  - Client-side (frontend) computing
  - Server-side (backend) computing

$\left.\begin{array}{l}\text{Client-side (frontend) computing}\\\text{Server-side (backend) computing}\end{array}\right\}$ Full Stack Development

- **Pure HTML with Server Process Support**
- **Client-side Implementation**

Domains of Web Computing and Applications

Web computing-based applications have penetrated all areas and are reshaping the world:

1. Business
2. Computing and Communication
3. Education
4. Financial Services
5. Healthcare and Health Plans
6. Manufacturing
7. Retail
8. Social and Media
9. LLMs: OpenAI, Microsoft Copilot, and NVidia NeMo use web services and micro services as their components
10. …

Web Computing Models

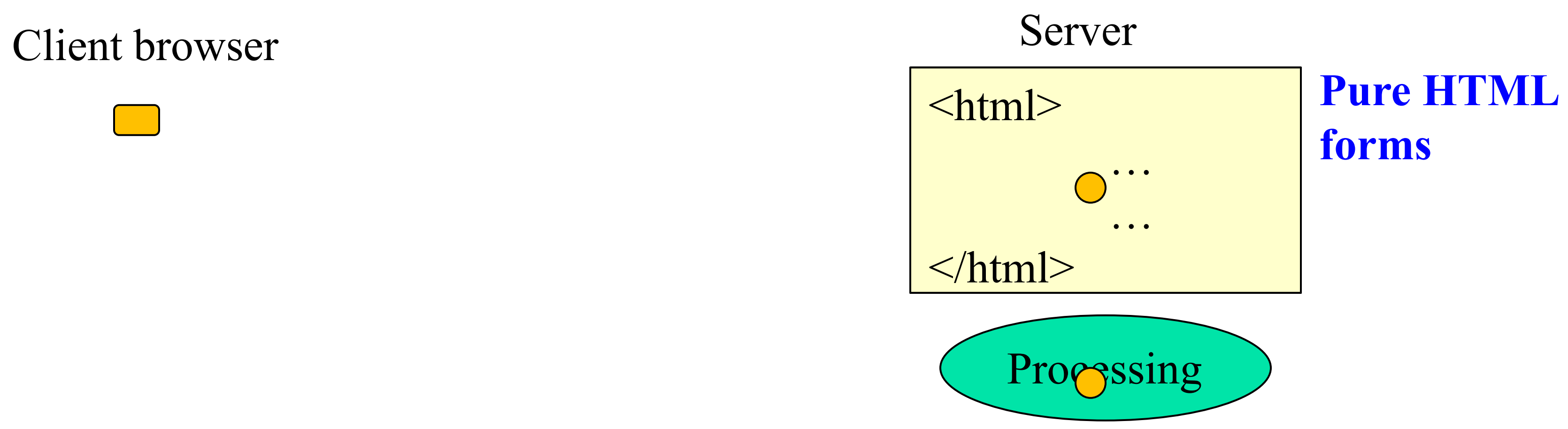
Frontend computing

Backend computing

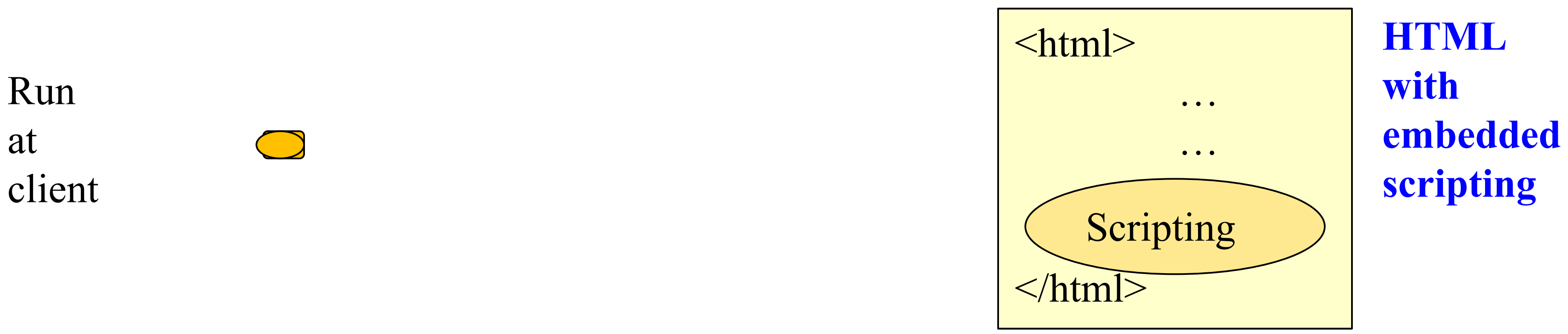
1. Pure HTML Web with Server Process Support
2. HTML with Embedded Client-Side Scripting
   - Dynamic HTML (DHTML) concept
   - E.g., JavaScript, VBScript
   - HTML5, but allowing standard library on local machine
3. Server-Side Scripting and Code Behind Presentation
   - Any programming language supported by the server
4. Page Postback vs. Partial Page Update
   - AJAX
5. Client-Side Out-Of-Browser Computing
   - Adobe Flash
   - Silverlight

HTML5

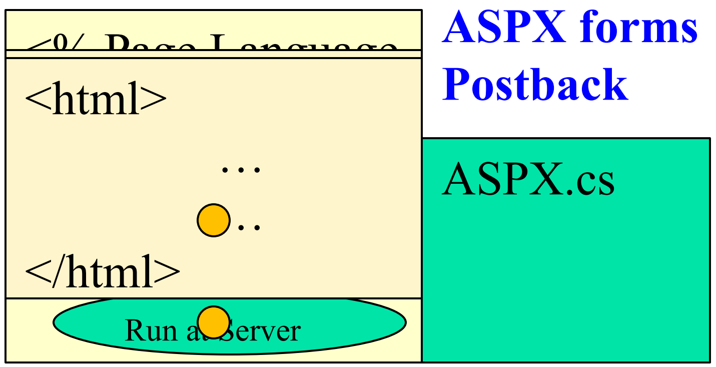
Web Computing Model 1 (Backend)



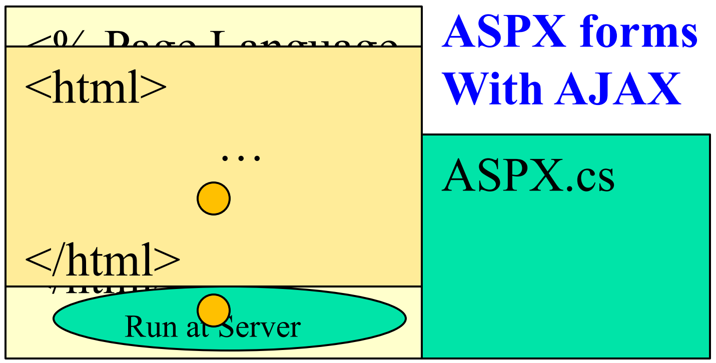
(Frontend) Web Computing Model 2



Web Computing Model 3 (Backend)

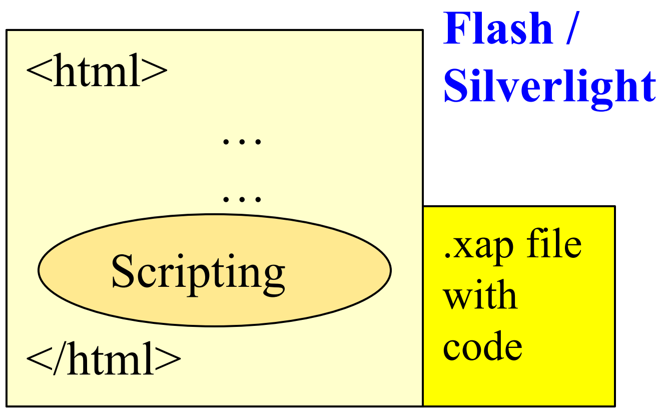


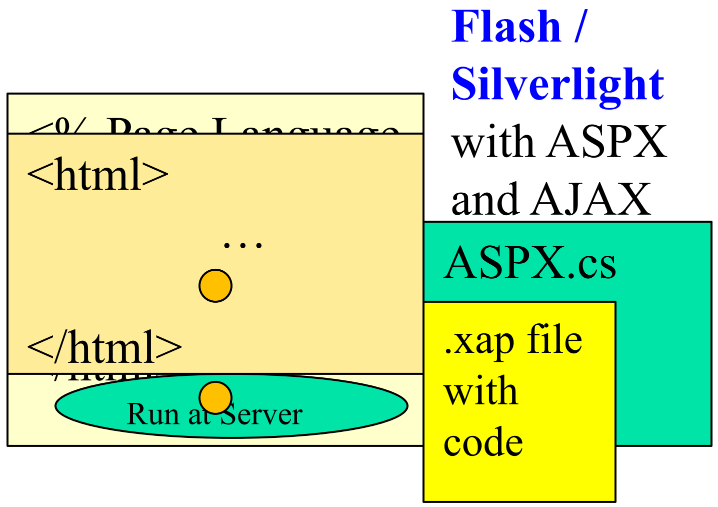
Web Computing Model 4 (Backend)



(Frontend) Web Computing Model 5

Run
at
client



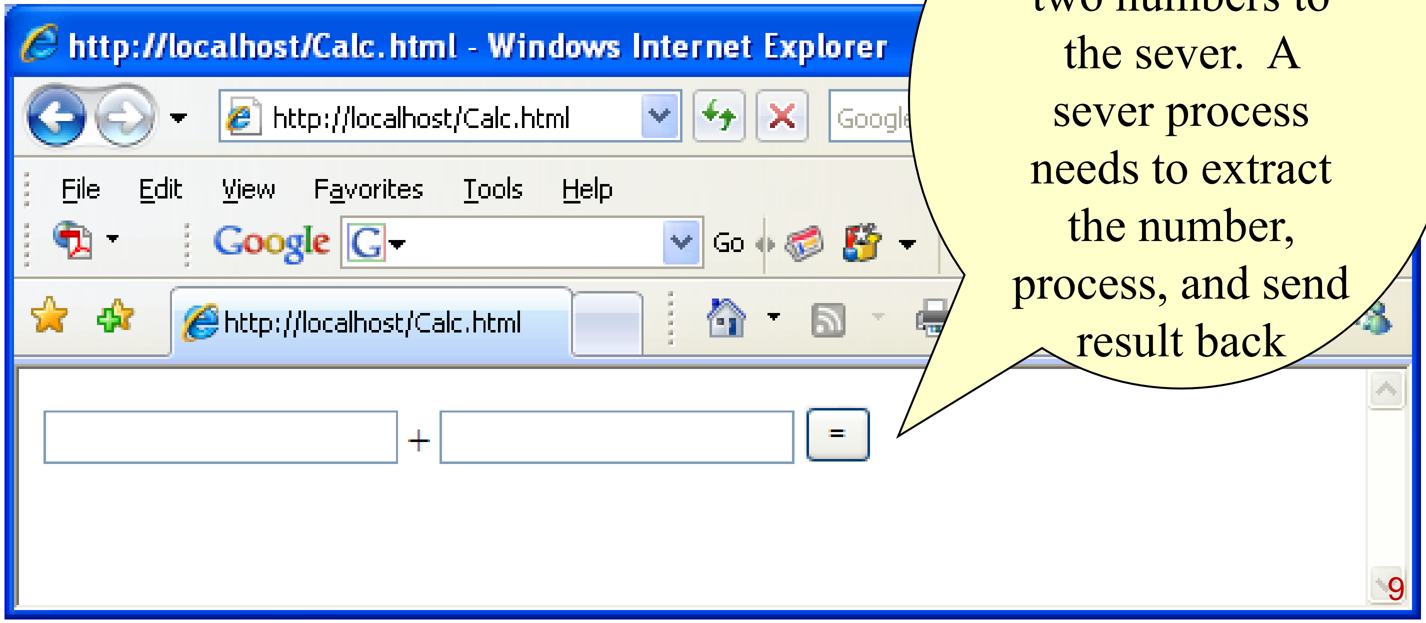


Implementing Web Computing Models Using Web Controls and Components

Examples of Web Computing Models and Web Applications

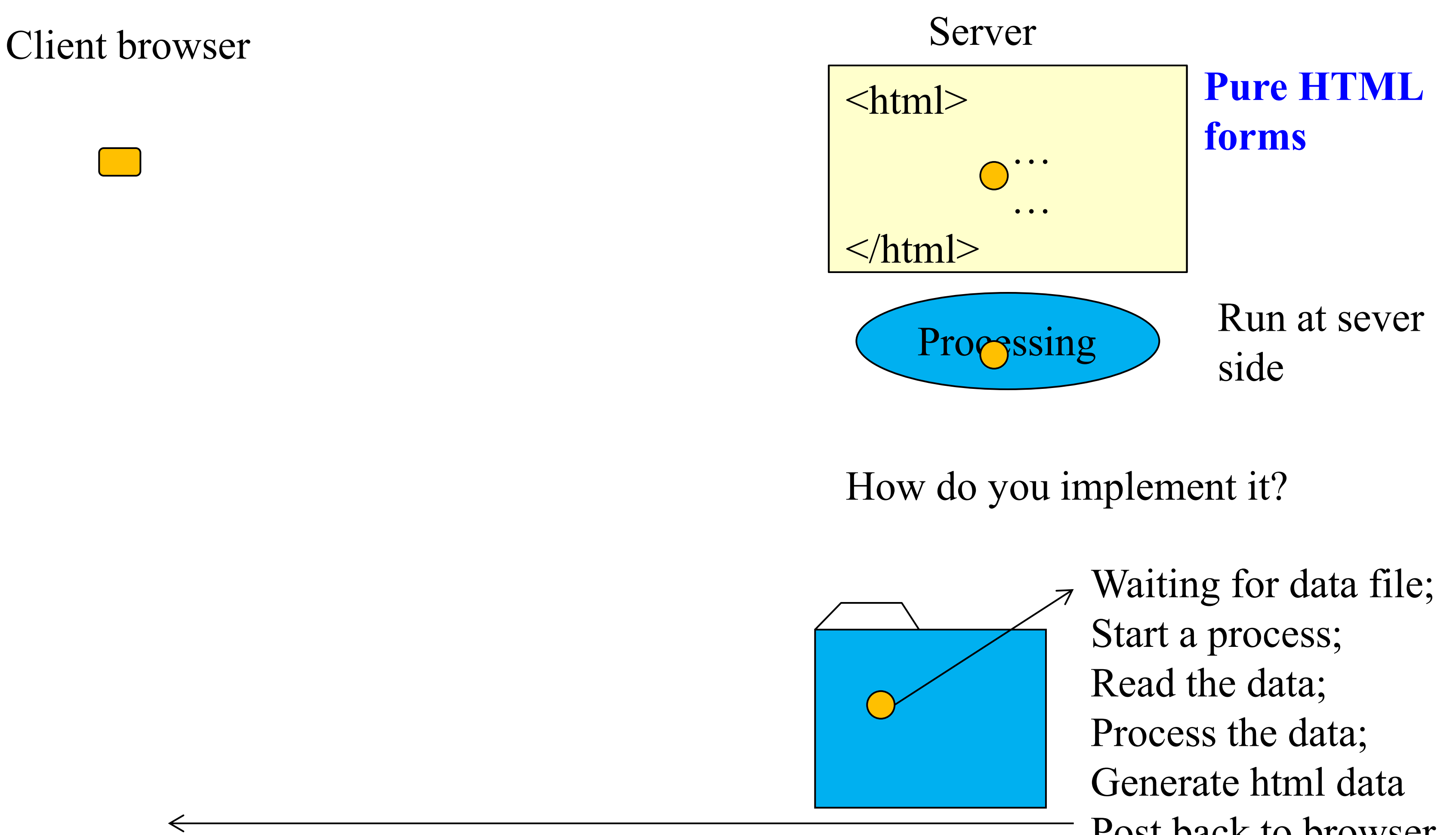
(1) Pure HTML Form for an “Adder” Page

```
<html>
    <body>
        <form>
                <input type="text" name="op1" />
                +
                <input type="text" name="op2" />
                <input type="submit" value=" = " />
        </form>
    </body>
</html>
```


http://localhost/Calc.html - Windows Internet Explorer
http://localhost/Calc.html
File Edit View Favorites Tools Help
Google
Go
+
=

The form sends two numbers to the sever. A sever process needs to extract the number, process, and send result back

Example of Web Computing Model with Pure HTML



How do you implement it?

Waiting for data file;
Start a process;
Read the data;
Process the data;
Generate html data
Post back to browser
Delete the process

How does the sever process handle the input numbers?

- CGI (Common Gateway Interface) for pure html
  - Under UNIX
  - Using Perl, PHP (other languages are possible)
  - It could be slow because, most implementations launch a new process to handle each request, even if just doing a+b

Example: Perl CGI Code for Generating HTML Form

Source: http://perlmeme.org/tutorials/cgi_form.html

```
#!/usr/bin/perl
  use strict; use warnir
   my $q = new CGI;  -
  print $q->header();
  # Output stylesheet,
  output_top($q);
  if ($q->param()) {    <
     # Parameters are
     display_results($q
  } else {
     # We're here for th
     output_form($q);
  }
  # Output footer and e
  output_end($q);
  exit 0;
  # Outputs the start ht
  sub output_top {
     my ($q) = @_;
     print $q->start_htn
        -title => 'A Ques
        -bgcolor => 'whi
        -style => {
                   -co
           /* Styleshe
              body {
              font-fam
           }
```

```
h2 {
                  color: darkblue;
                  border-bottom: 1pt solid
                  width: 100%;
               }
               div {
                  text-align: right;
                  color: steelblue;
                  border-top: darkblue 1pt
                  margin-top: 4pt;
               }
               th {
                  text-align: right;
                  padding: 2pt;
                  vertical-align: top;
               }
               td {padding: 2pt; vertical-a
Stylesheet code */
           ',
                       },
    );
    print $q->h2("A Questionaire");
  }
  # Outputs a footer line and end html
sub output_end {
    my ($q) = @_;
    print $q->div("My Web Form");
    print $q->end_html;
  }
```

```
# Displays the results of the fc
  sub display_results {
     my ($q) = @_;
     my $username = $q->par
     my $userage = $q->para
     my $usergender = $q->p
     my @favourite_language
     my %sex = ('F' => 'girl',
     print $q->h4("Hi $userna
     print $q->p("You are a $
     print $q->p("Your favou
     print $q->table(
        {-border => 1, -cellpa
        $q->Tr($q->td(\@favo
     );
  }
  # Outputs a web form
  sub output_form {
     my ($q) = @_;
     print $q->start_form(
        -name => 'main',
        -method => 'POST',
print $q->start_table;
     print $q->Tr(
       $q->td('Name:'),
       $q->td(
         $q->textfield(-name =
     ) )
```

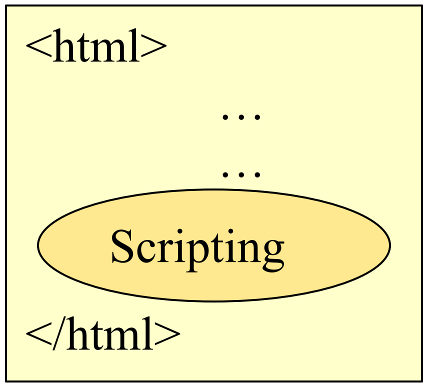
```
print $q->Tr(
     $q->td('Age:'),
     $q->td(
       $q->radio_group(
         -name => 'age',
         -values => [
            '0-12', '13-18', '18-30', '30-40', '40-50', '50-60', '60-7
         ],
         -rows => 4,
       )))
my %genders = ('F' => 'Female', 'M' => 'Male');
     print $q->Tr(
       $q->td('Gender:'),
       $q->td(
         $q->popup_menu( -name => 'gender', -values => [keys
\%genders,)));
     print $q->Tr(
       $q->td('Favourite Languages:'),
       $q->td(
         $q->checkbox_group(
           -name => 'language',
           -values => ['Perl', 'C', 'C++', 'C#', 'Java', 'VB', 'Python'
           -defaults => ['Perl'],
           -columns => 2,
         )))
print $q->Tr($q->td($q->submit(-value => 'Submit')), $q->td('
     print $q->end_table;
     print $q->end_form;
   }
```

How does a sever process handle the input numbers?

- CGI (Common Gateway Interface) for pure html
  - Under UNIX
  - Using Perl, PHP (other languages are possible)
  - It could be slow because, most implementations launch a new process to handle each request, even if just doing a+b
- ISAPI (Internet Server API) extension of DLL for pure html
  - Under Windows OS
  - Low level programming required.
    For a simple Calculator application, it is about 80 lines of code to extract numbers, process them, and post back.
- Scripting: Javascript in html and Active Server Pages (.asp)
- ASP .Net (.aspx)
  - Using html control
  - Using Web control

(2) Implementing Web Computing Model with Client-Side Scripting (Frontend Development)

Run
at
client ⬭ ▭

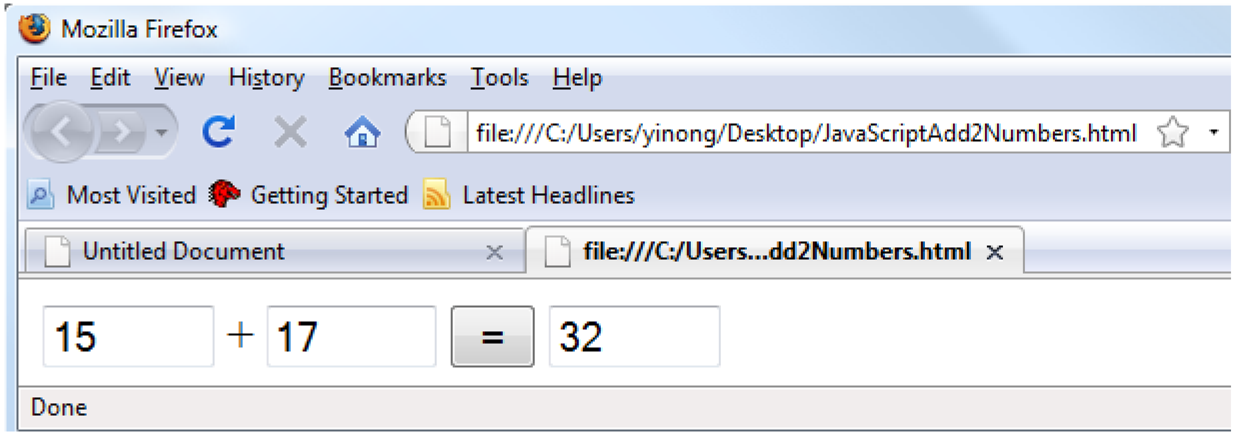


HTML with embedded scripting

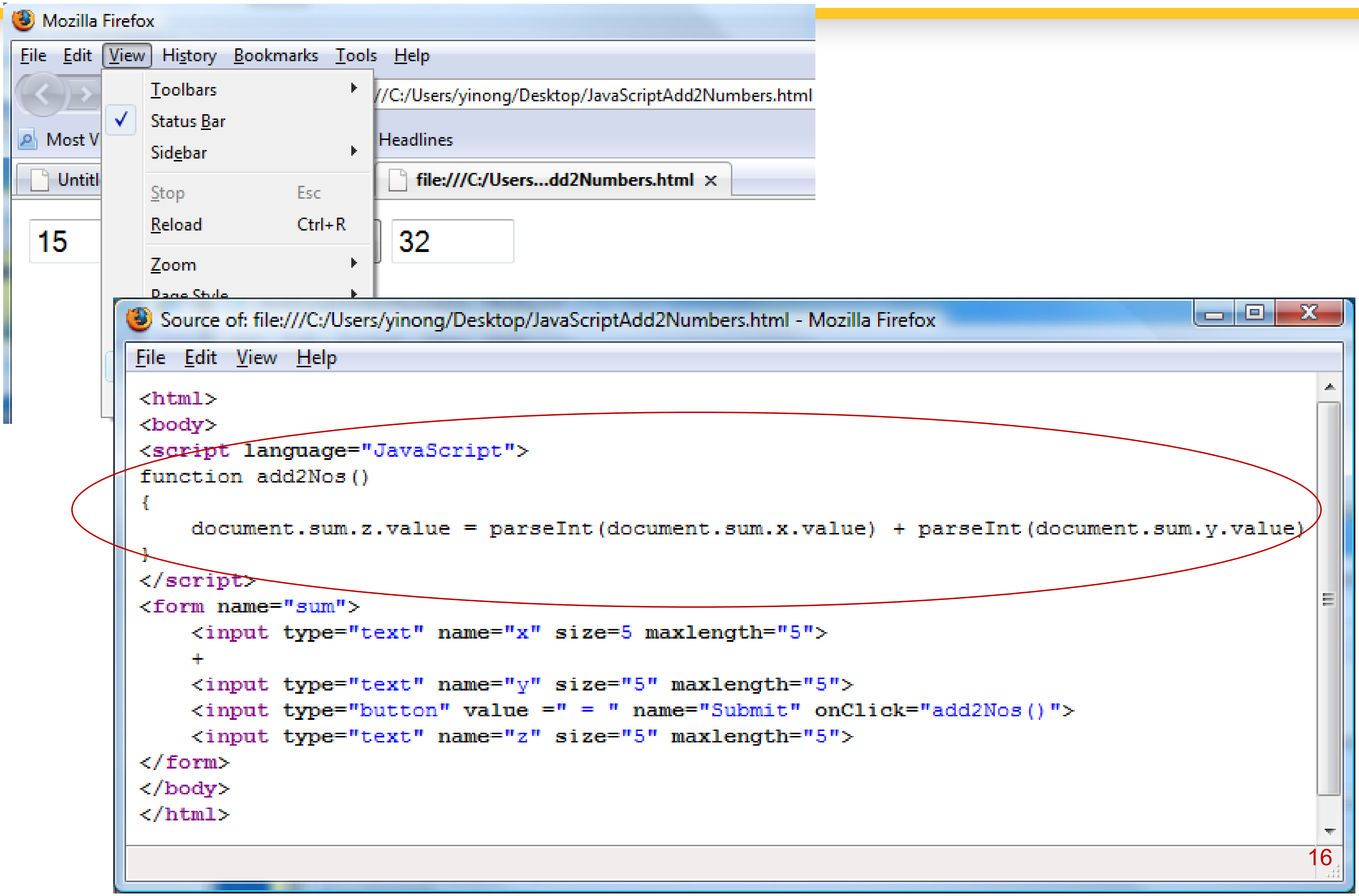
Example of Client-Side Scripting (Frontend)

Script is embedded in html

```
<html>  <body>
<script type="text/JavaScript">
   function add2Nos() {
      document.sum.z.value = parseInt(document.sum.x.value) +
           parseInt(document.sum.y.value)
   }
</script>
<form name="sum">
   <input type="text" name="x" size=5 maxlength="5">
   +
   <input type="text" name="y" size="5" maxlength="5">
   <input type="button" value =" = " name="Submit" onClick="add2Nos()">
   <input type="text" name="z" size="5" maxlength="5">
</form>
</body>
</html>
```


Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Users/yinong/Desktop/JavaScriptAdd2Numbers.html
Most Visited Getting Started Latest Headlines
Untitled Document
file:///C:/Users...dd2Numbers.html
15 + 17 = 32
Done

What can be seen by View Source?


Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbars
Status Bar
Sidebar
Stop Esc
Reload Ctrl+R
Zoom
Page Style
//C:/Users/yinong/Desktop/JavaScriptAdd2Numbers.html
Most V
Headlines
Untitl
file:///C:/Users...dd2Numbers.html
15
32
Source of: file:///C:/Users/yinong/Desktop/JavaScriptAdd2Numbers.html - Mozilla Firefox
File Edit View Help

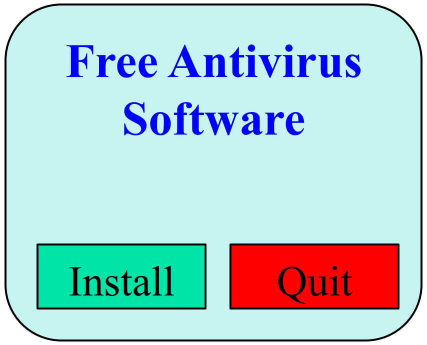
```
<html>
<body>
<script language="JavaScript">
function add2Nos()
{
    document.sum.z.value = parseInt(document.sum.x.value) + parseInt(document.sum.y.value)
}
</script>
<form name="sum">
    <input type="text" name="x" size=5 maxlength="5">
    +
    <input type="text" name="y" size="5" maxlength="5">
    <input type="button" value =" = " name="Submit" onClick="add2Nos()">
    <input type="text" name="z" size="5" maxlength="5">
</form>
</body>
</html>
```

Good and Bad of Client-Side Scripting

- Computing is done in the client browser. Reduce the overhead of sending data back and forth;
- Script programs are interpreted. Interpretation is slower than executing compiled code, if code is heavy;
- Script code is not reusable: Cannot be called from other pages.
- Code can get messy if one tries to implement thick client
- Security and proprietary issues:
  - Allowing the code running on the client machine presents a security hazard to the client machine;
  - Code is visible via "View Source".

Security and Proprietary with Client-Side Scripting

- Security Issues:
  - Code is visible via "View Source", allowing the attackers to explore the data processing process
  - Cross-Site Scripting (XSS): Script programs are transferred from attacker's page to the visitor's browser and are executed on client side
- Proprietary issues:
  - Code is visible via "View Source", allowing competitors to explore the business logic behind the programs.
- What is about the other client-side computing model: Silverlight or flash-based out-of-browser computing model?



Advancement in HTML5

- HTML5 is largely based on the script computing on the client side.
- It is more than the script computing.
- Development efficiency improved by library availability and framework support. The library functions (API calls) can be pre-compiled, instead of scripting, running on local machine, instead of in browser. JavaScript code still running in browser.
- Source code visible still a problem for many applications
  - Security issues
  - Proprietary issues

ASU Ira A. Fulton Schools of Engineering

Arizona State University

M11 L2
Web Computing Models: Server-Side Implementation

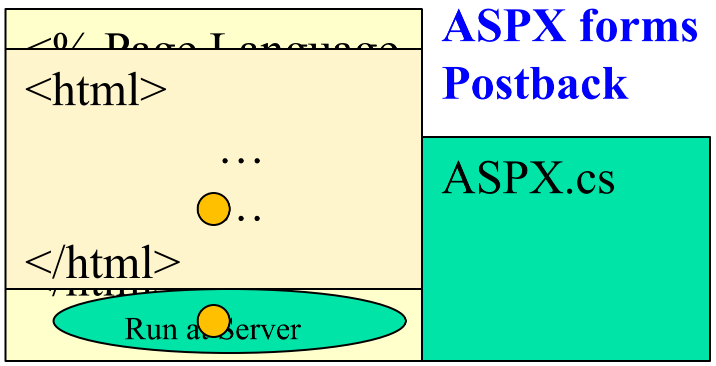
Ira A. Fulton Schools of Engineering
Arizona State University

Lecture Outline

- Server-Side Implementation (Backend)
- Out-of-Browser Computing (Frontend)
- Reverse Engineering

(3) Implementing Web Computing Model Server-Side Computing (Backend)

Use of Server Controls



Server-Side Scripting Using HTML Control

```
<html>
<head runat="server">
    <title>Server Side Script</title>
</head> <body >
    <form id="Form1" RunAt = "Server" >
        <input id="Text1" type="text" RunAt = "Server" />
        +
        <input id="Text2" type="text" RunAt = "Server" />
        <input type="submit" value= " = "
            ID = "z"
            OnServerClick = "OnAdd"
            RunAt = "Server" />
    </form>
</body> </html>
<script language= "C#" runat="server">
    void OnAdd (Object sender, EventArgs e) {
    Int32 x = Convert.ToInt32(Text1.Value);
    Int32 y = Convert.ToInt32(Text2.Value);
    z.Value = Convert.ToString(x + y);
    }
</script>
```

Integrate C# code in html file, But code separated from the html part

View Page Source: Program not Visible

```
<html>
<head><title>Server Side Script</title></head> <body >
    <form name="Form1" method="post" action="Default.aspx" id="Form1">
        <div>
            <input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
            <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
            <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
    value="/wEPDwUKLTkzMzQ0OTIwNg9kFgICAw9kFgICBQ8WAh4FdmFsdWUFAjU1ZGSVLB7EHEiZszM7bxtX2itVFSK2oA==" />
        </div>
    <script type="text/javascript">
        //<![CDATA[var theForm = document.forms['Form1'];
        if (!theForm) {
            theForm = document.Form1;
        }
        function __doPostBack(eventTarget, eventArgument) {
            if (!theForm.onsubmit || (theForm.onsubmit() != false)) {
                theForm.__EVENTTARGET.value = eventTarget;
            theForm.__EVENTARGUMENT.value = eventArgument;
            theForm.submit();
            }
        }
        //]]>
    </script>
                        <div>
                        <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
                        value="/wEWBALd0uq2CQLzlKGwCgL2lKGwCgKmuYiBBSKn5+tsAi8rGZvqAijK+zePk9vB" />
                        </div>
                                    <input name="Text1" type="text" id="Text1" />
                                    +
                                    <input name="Text2" type="text" id="Text2" />
                                    <input name=“z" type="submit" id=“z" value=" = " />
                                </form>
                            </body>
                        </html>
```

Server-Side Scripting Using **Web Controls**

Default.aspx

```
<html>
<head runat="server"><title>Server Side Scripting</title></head>
   <body>
       <form runat="Server">
          <asp:TextBox ID= "x" RunAt="server" />
          +
          <asp:TextBox ID= "y" RunAt="server" />
          <asp:Button Text= " = " OnClick = "OnAdd" RunAt= "server" />
          <asp:Label ID= "z" RunAt="server" />
       </form>
   </body>
</html>
<script language= "C#" runat="server">
   void OnAdd (Object sender, EventArgs e) {
       Int32 a = Convert.ToInt32(x.Text);
       Int32 b = Convert.ToInt32(y.Text);
       z.Text = (a + b).ToString();
   }
</script>
```

Another possible way

Default.aspx.cs

What You Can View in the Browser?

You can only view html file when you use View Source Command to view .aspx file -- an html file will be generated:

```
<html>
<head><title>Server Side Script</title></head>
      <body>
            <form name="ctl01" method="post" action="Default.aspx" id="ctl01">
<div>
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKM
</div>
<div>
      <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
</div>
                  <input name="x" type="text" id="x" />
                  +
                  <input name="y" type="text" id="y" />
                  <input type="submit" name="ctl03" value=" = " />
                  <span id="z"></span>
            </form>
      </body>
</html>
```

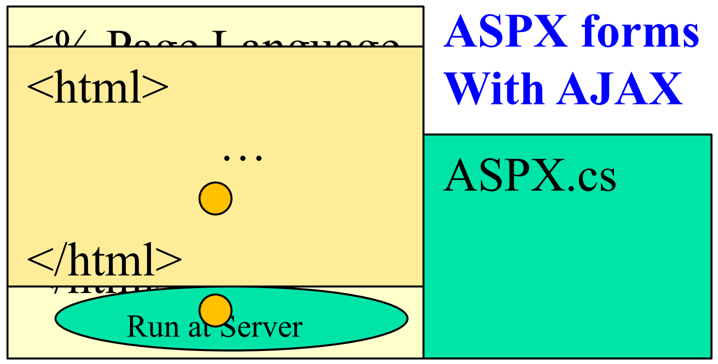
This approach is similar to abstract data type/class: Data are private and have to be accessed via a method.

Adding exception handler in the .aspx page

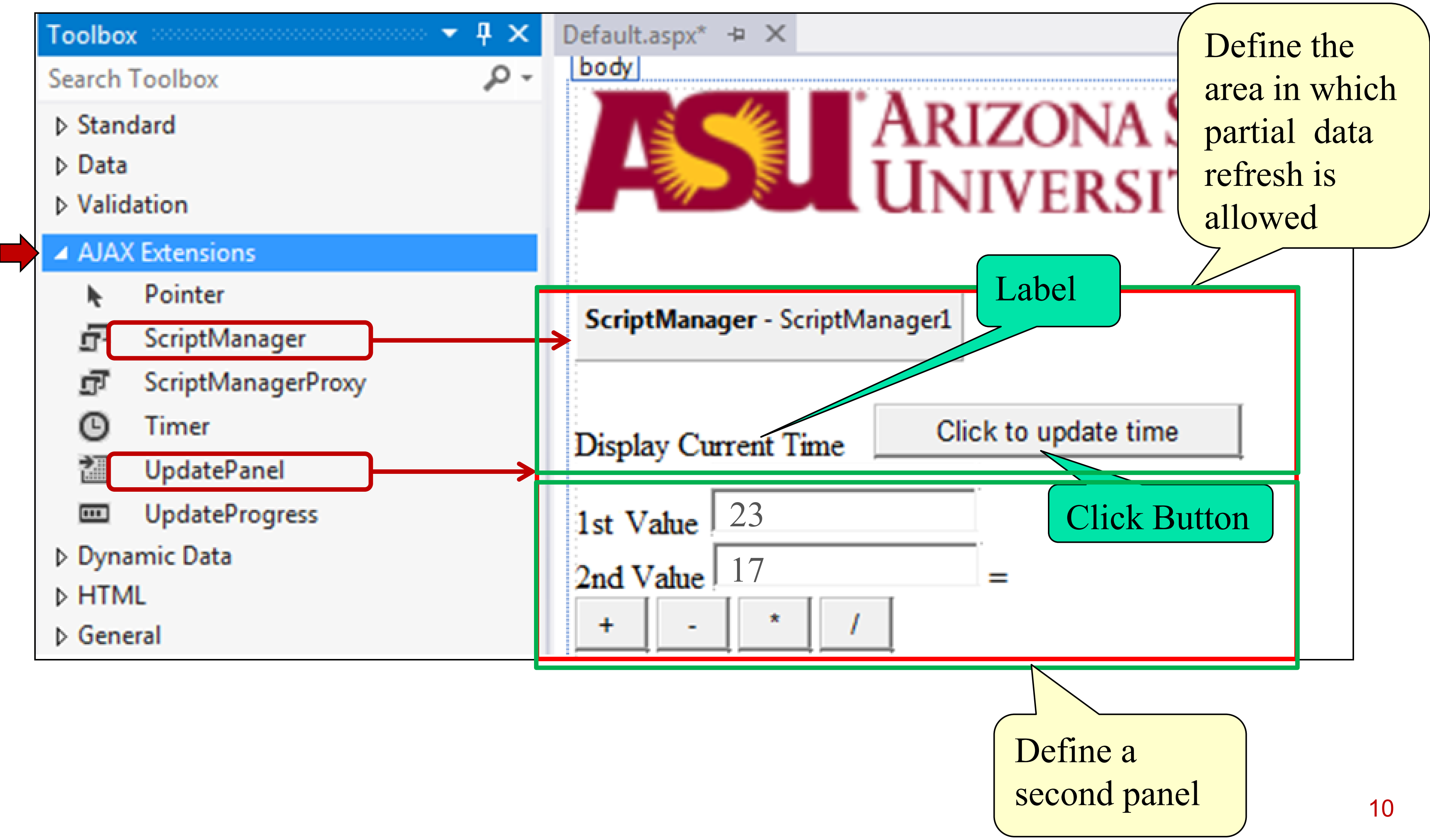
In the previous form, if a noninteger is entered, the form will return an "uncaught exception". In aspx, one can add an exception handler to catch the error:

```
<script language= "C#" runat="server">
    void OnAdd (Object sender, EventArgs e)  {
            try {
                Int32 a = Convert.ToInt32(x.Text);
                Int32 b = Convert.ToInt32(y.Text);
                z.Text = (a + b).ToString();
            }
            catch (FormatException) {
                z.Text = "Please enter integers only";
            }
      }
</script>
```

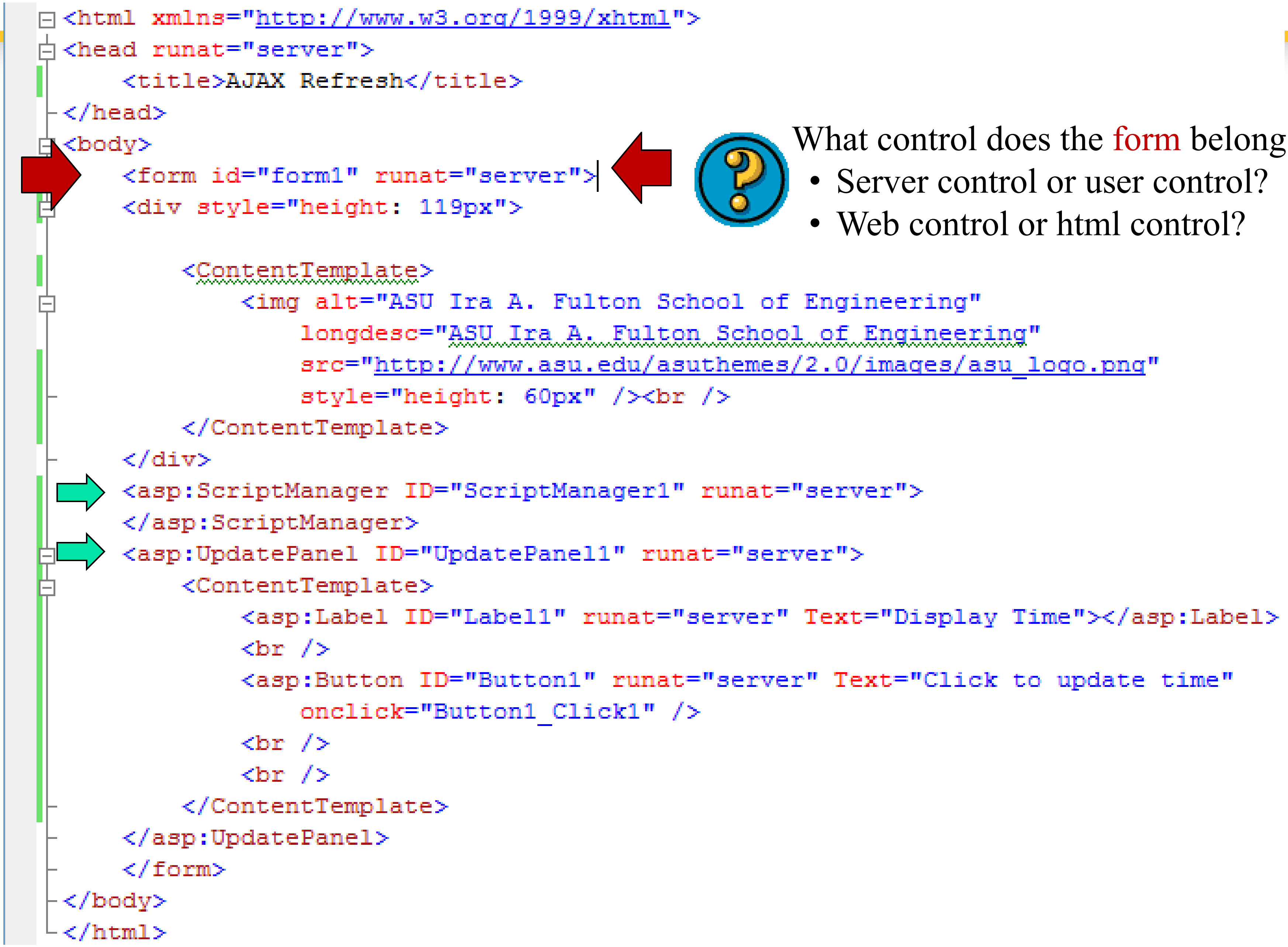
(4) Implementing Web Computing Model AJAX



Example: AJAX **Web Controls** (Backend)



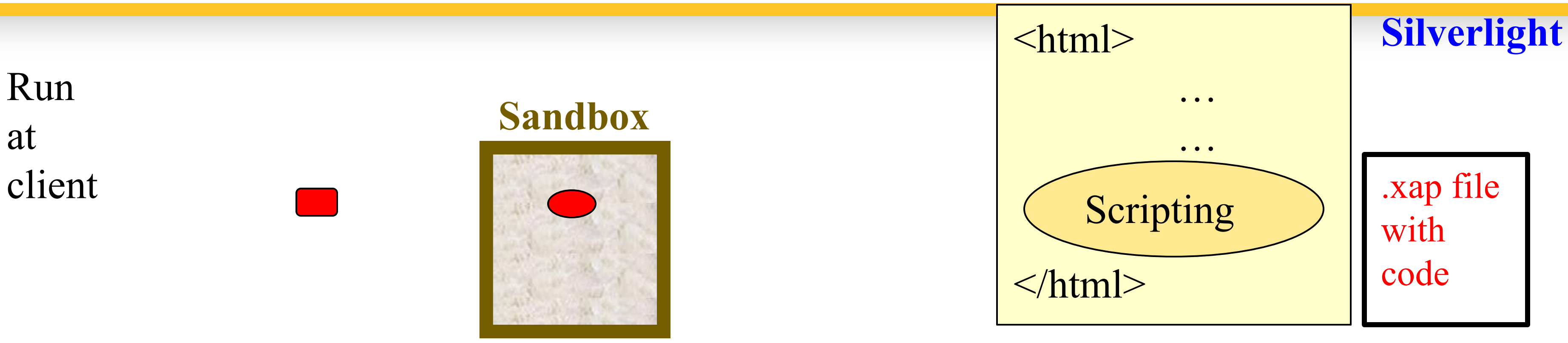
Example: AJAX Web Controls



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>AJAX Refresh</title>
</head>
<body>
    <form id="form1" runat="server">
    <div style="height: 119px">

        <ContentTemplate>
            <img alt="ASU Ira A. Fulton School of Engineering"
                longdesc="ASU Ira A. Fulton School of Engineering"
                src="http://www.asu.edu/asuthemes/2.0/images/asu_logo.png"
                style="height: 60px" /><br />
        </ContentTemplate>
    </div>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>
            <asp:Label ID="Label1" runat="server" Text="Display Time"></asp:Label>
            <br />
            <asp:Button ID="Button1" runat="server" Text="Click to update time"
                onclick="Button1_Click1" />
            <br />
            <br />
        </ContentTemplate>
    </asp:UpdatePanel>
    </form>
</body>
</html>
```

What control does the form belong to?
- Server control or user control?
- Web control or html control?

(5) Example of Out-Of-Browser Model



- Computing is done on client side (frontend). Reduce the overhead of sending data back and forth;
- Code is structured like server-side programming (backend development style);
- **Security** and **proprietary** issues for any client-side computing model:
  - Allowing the code running on the client machine presents a security hazard to the client machine.
    Silverlight Sandbox Solution: Not allowed to access any native applications, unless through public APIs or services.
  - Code is visible via "reverse engineering".

“Reverse Engineering” of Silverlight Code

- A deployed Silverlight application has two files:
  - An html file and a .xap file. The latter is a zip file
- Use “view source” to view HTML source;
- An element in the HTML code pointing to the xap file:

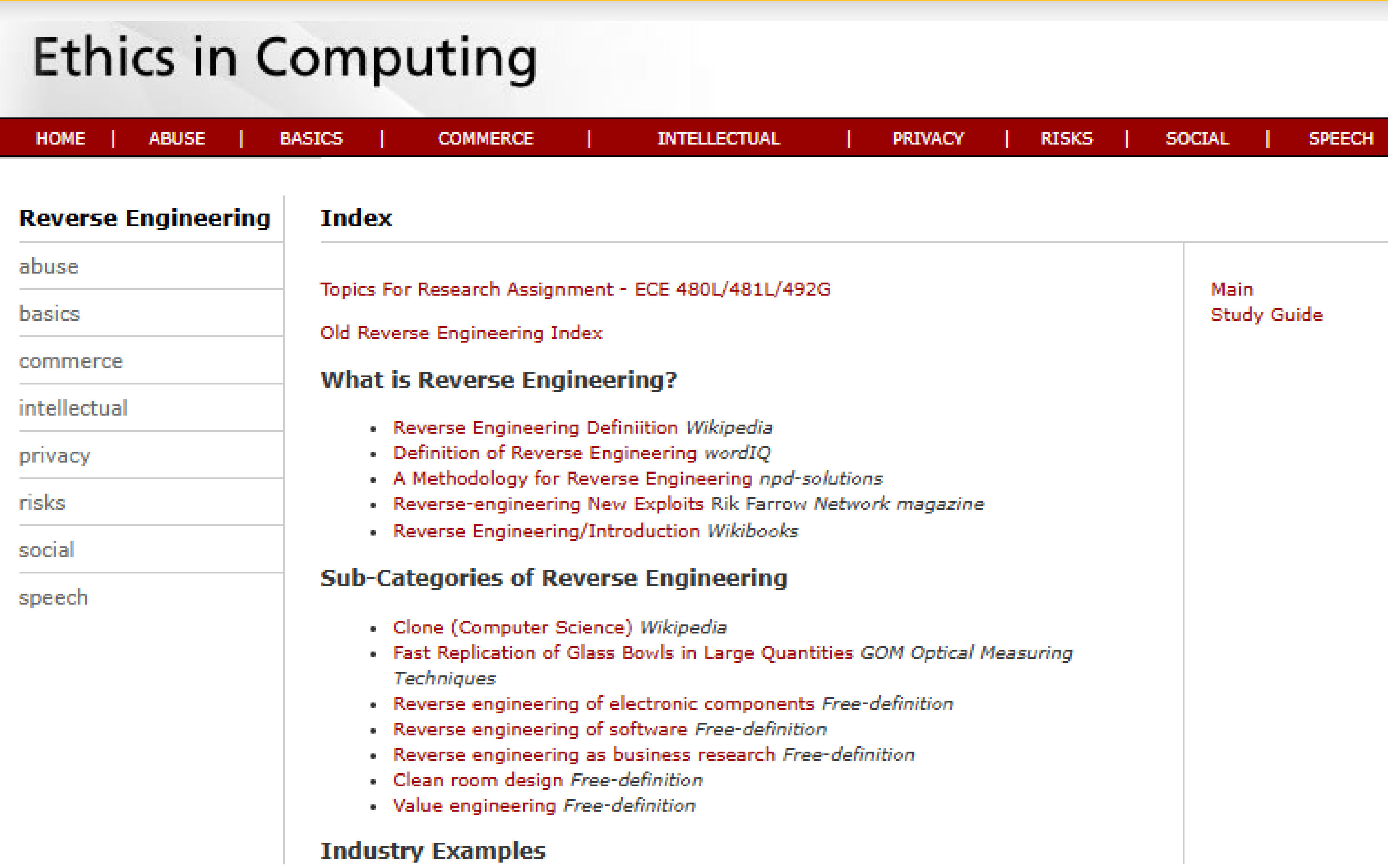
  `<param name=“source” value=“myCalculator.xap” />`
- In the address bar of the browser, replace myCalculator.xap for HTML file name, e.g., index.html
- Download/Save the file, & rename the file to a .zip file;
- Unzip the file, you obtain the MSIL assembly code;
- Use a tool, e.g., Reflector, you can view the MSIL code;
- There are tools that decompile the MSIL code to C#.

Ethics and Laws on "Reverse Engineering"

- Reverse engineering is acceptable, only if
  - You own the source code;
  - You have explicit permission to view the code; or
  - It is open-source code.
- Otherwise, reverse engineering is
  - unethical for reading the code
  - a crime for using the code
- Laws related to reverse engineering, e.g.,:
  http://ethics.csc.ncsu.edu/intellectual/reverse/

- As a software engineer, prevent "reverse engineering":
  - Server-side code for security and proprietary code
  - Use an obfuscation tool to make it harder. VS comes with a tool named Dotfuscation. Other tools exist.

Ethics and Laws on "Reverse Engineering"
http://ethics.csc.ncsu.edu/intellectual/reverse/



Index

Topics For Research Assignment - ECE 480L/481L/492G

Old Reverse Engineering Index

What is Reverse Engineering?

- Reverse Engineering Definiition *Wikipedia*
- Definition of Reverse Engineering *wordIQ*
- A Methodology for Reverse Engineering *npd-solutions*
- Reverse-engineering New Exploits Rik Farrow *Network magazine*
- Reverse Engineering/Introduction *Wikibooks*

Sub-Categories of Reverse Engineering

- Clone (Computer Science) *Wikipedia*
- Fast Replication of Glass Bowls in Large Quantities *GOM Optical Measuring Techniques*
- Reverse engineering of electronic components *Free-definition*
- Reverse engineering of software *Free-definition*
- Reverse engineering as business research *Free-definition*
- Clean room design *Free-definition*
- Value engineering *Free-definition*

Industry Examples

Ira A. Fulton Schools of Engineering

Arizona State University

M11 L3
Web Application Architecture

Ira A. Fulton Schools of Engineering
Arizona State University

Lecture Outline

- **Web Architecture vs. Web Computing Models**
  - Thin Client
  - Thick Client
- **Application Domains in Web Server**
- **Structure of ASP .Net Web Applications**

Web Architecture vs. Web Computing Models Thin and Thick Client

- Thin Client Architecture
  - Pure HTML Web with Server Process Support
  - Client-Side Scripting with Lightweight Programming
  - Server-Side Scripting and Code Behind Presentation
- Thick Client Architecture
  - Client-Side Scripting with Heavyweight Programming
    - HTML5, with adequate library support and with framework support
  - Client-Side Out-Of-Browser Computing
    - Good use of thick client
    - Examples: Adobe Flash and Silverlight in gaming

Useful in specific domains, such as games.

Web Application vs. Desktop Application

- A traditional desktop (console) application
  - has a unique entry point – main method;
  - follows control flow computing model;
  - can be compiled into a stand-alone executable file;
  - can consist of many files, but a project file organizes them into a well-defined application domain.
- A Web application
  - consists of a collection of Web pages (groups of classes);
  - can be entered from any of the pages, even if the designer has a default “entry” page in mind;
  - follows event-driven computing model
  - has a coherent mission and shares common resources.

Web Applications & Distributed Computing

- A Web application within its application domain is distributed
  - It consists of multiple “pages”, each of which is an autonomous “object”;
  - The pages share common resources in the application domain;
  - The pages can communicate with each other in a loosely couple manner: shared memory, asynchronous callback;
  - Multiple instances (threads) can be created from one application or service to support multiple users in parallel.
- A Web application exists beyond its application domain
  - Use remote web services / remote APIs as its functional units
  - Communicate with other Web applications

Web Applications & Future Computing

- Web applications are rapidly expanding
  - For every desktop application, a web version has been or is being developed;
  - Web 2.0 (Web is the computing platform);
  - Web integration using programming languages
  - Web integration based on workflow and visual programming
  - Application and database integration
  - Big data, ontology, AI, and machine learning
  - Cloud computing: Moving from desktop computing to Web-based computing and enabling programs and data access anywhere and anytime through:
    - SaaS (Software as a Service): service-oriented computing
    - PaaS (Platform as a Service): IDE, e.g., VS, Eclipse
    - IaaS (Infrastructure as a Service): Hardware as services

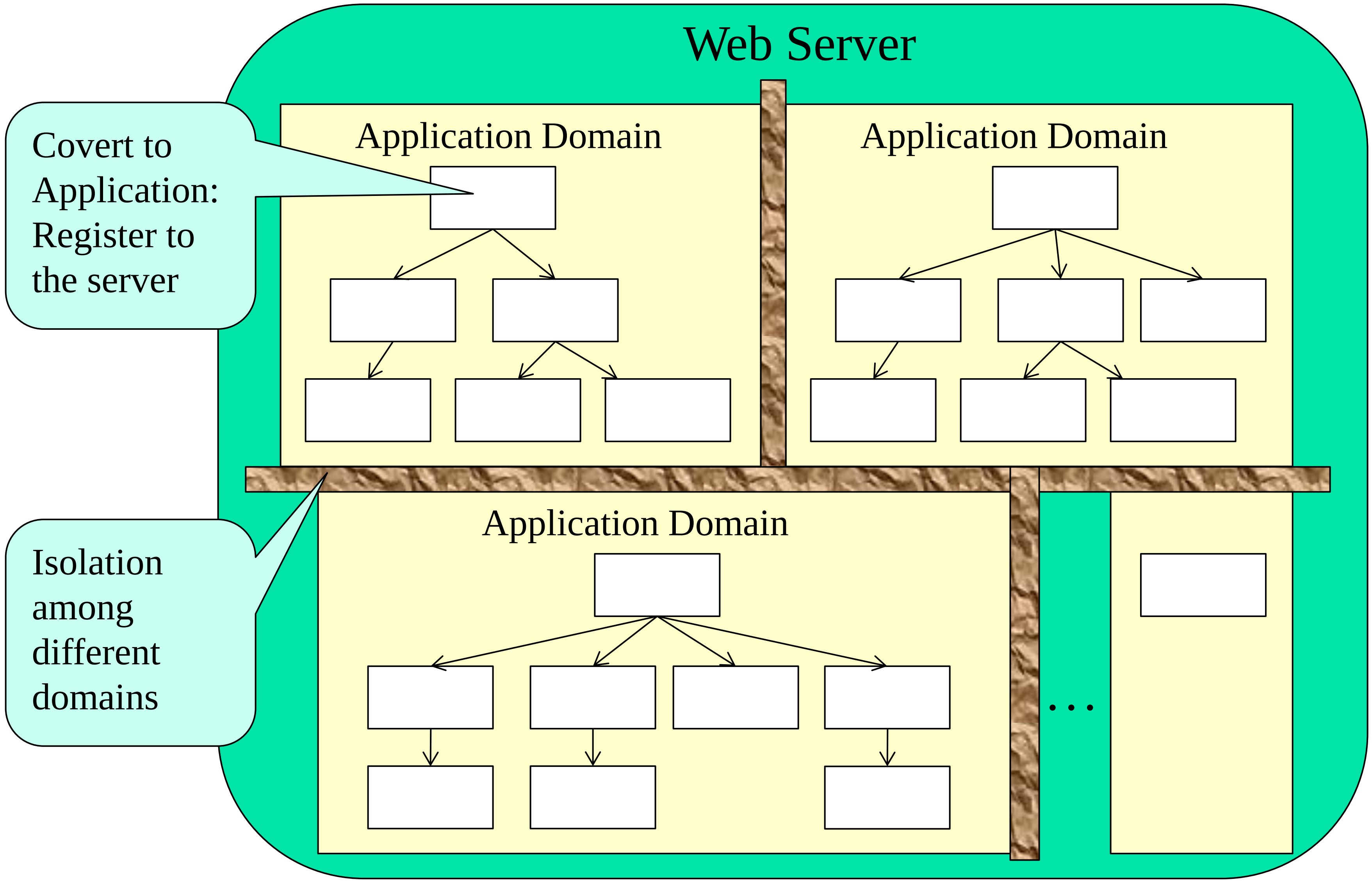
AWS
Google
Microsoft

Web

Part II of Text

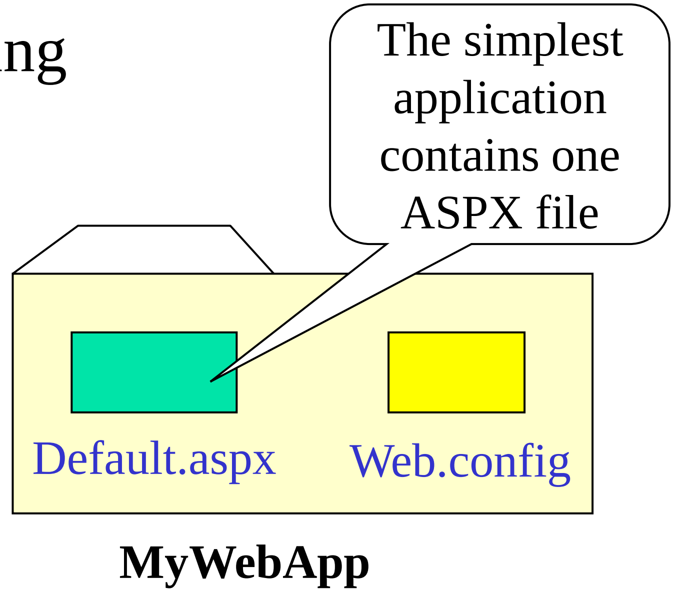
Application Domains in a Web Server

Web applications are hosted in Web server. Applications are independent of each others.


Web Server
Application Domain
Application Domain
Application Domain
Covert to Application: Register to the server
Isolation among different domains
. . .

Structure of ASP .Net Web Applications

- An ASP.Net application consists of all the files in the application directory and its sub-directories, and possibly sibling directories.
- An ASP.Net application includes following types of files:
  - ASPX files and html/Web controls.
  - ASCX files forming user controls.
  - Web.config files containing configuration settings.
  - A single Global.asax file containing global application class elements
  - DLL (dynamic link library) files containing custom types employed by the application.
  - Service files: services used, typically not but can be in the application folder (directory).

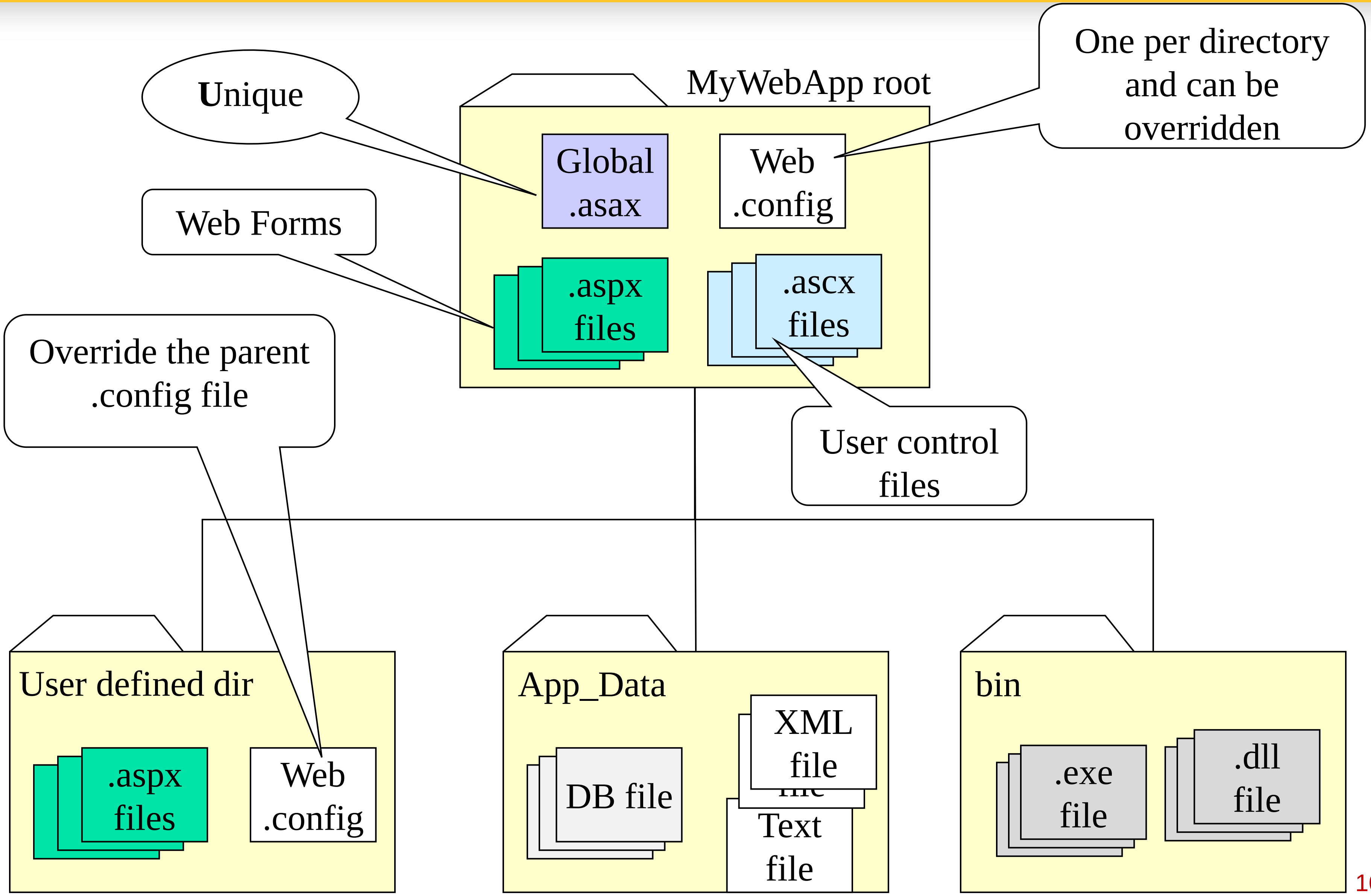


Directories Created in Application

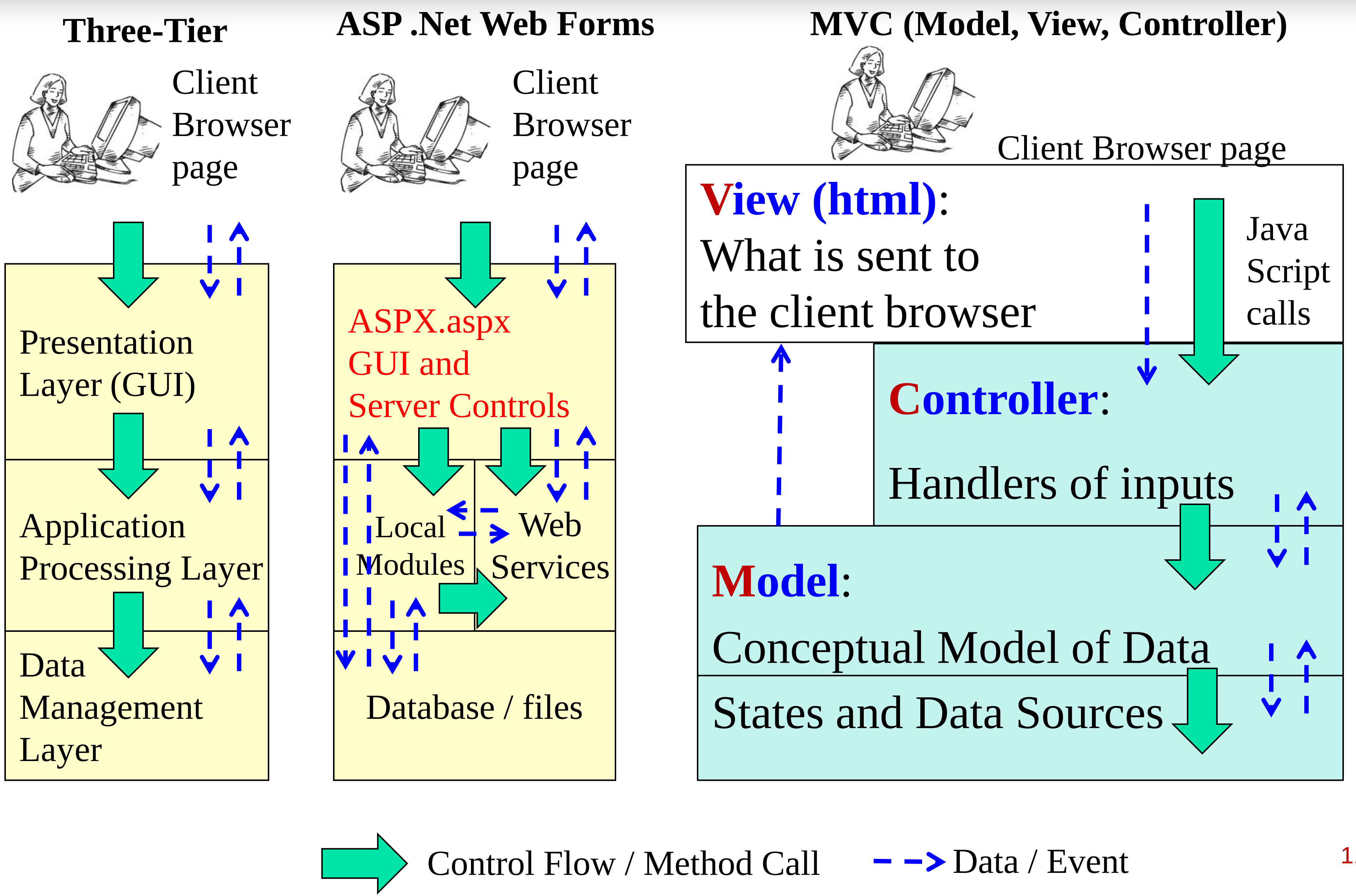
- Bin: Contains executable code (pre-compiled) and DLL files
- App_Code: Contains source code of programs (not compiled)
- App_Data: Store text files, XML files, and database files
- App_WebReferences: Store references to Web services that are added to the application
- …

You do not need to convert these components to “Application”, as they are not directly access from browser.

A More Complex ASP.Net Web Application



Three-Tier, ASP .Net Web Form, and MVC


Three-Tier
Client Browser page
Presentation Layer (GUI)
Application Processing Layer
Data Management Layer
ASP .Net Web Forms
Client Browser page
ASPX.aspx GUI and Server Controls
Local Modules
Web Services
Database / files
MVC (Model, View, Controller)
Client Browser page
View (html): What is sent to the client browser
Java Script calls
Controller: Handlers of inputs
Model: Conceptual Model of Data
States and Data Sources
Control Flow / Method Call
Data / Event

ASU Ira A. Fulton Schools of Engineering

Arizona State University

M11 L4
Web Application Controls and Components

Ira A. Fulton Schools of Engineering
Arizona State University

Lecture Outline

- **Server Controls**
  - HTML Controls
  - Web Controls
- **Applying Server Controls to Display XML File**

ASPX Web Page and Server Controls

- An ASPX page's Graphic User Interface (GUI) can be built using a set of Server Controls;
- A server control
  - is an "object", which forms a component in the ASPX page;
  - can take input and emit an event upon an input, and thus, we can write an event handler to process the input;
  - can generate an output and render the output into html format, so that the output can be displayed in a web page.

Server Controls Available

- There are two kinds of Server controls for GUI design:
  - HTML controls: They are objects on server side that generate the similar-looking html components (tags), so that all html functions can be handled in one-to-one mapping.
    - Handlers can be linked behind the controls, and thus, the data can be processed inside the application;
    - In pure HTML case, the data has to be sent to the server and the handler has to be located outside;
  - Web controls: They provide additional components that do not exist in html tags. These components are often more powerful.

ASPX Web Form Design and Controls

- **html controls** Each html tag has a counterpart in html control. For example:

  ```
  <input type="text" RunAt="server" />
  <form RunAt "Server">
  ```

- **Web controls**: add more complex components to the GUI design. A Web control is defined by a library class

  Web controls come from classes defined in the namespace System.Web.UI.WebControls. They are declared by prefixing the class name with asp and has an attribute RunAt = "server". For example:

  ```
  <asp:TextBox ID = "op1" RunAt="server" />
  ```

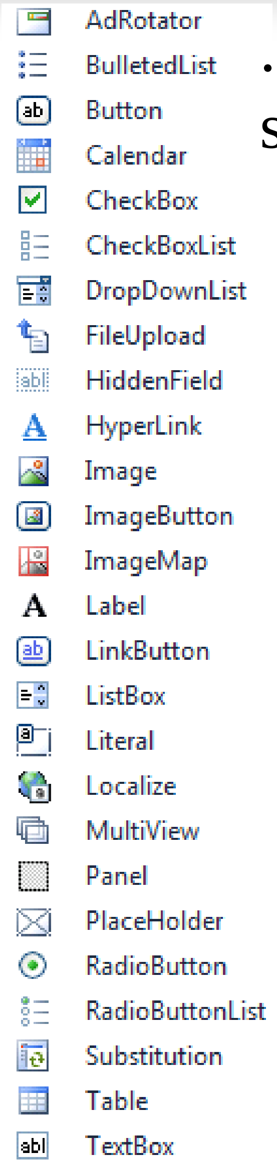
HTML Controls on Server Side

- To facility direct translation of existing html file into web form, for each action tag, .NET FCL provides a corresponding html Control:

Tag

<button runat= “server”>	*HtmlButton*	**HTML Input Control**
<input type=“button” runat=“server”>	*HtmlInputButton*	
<input type=“reset” runat=“server”>	*HtmlInputButton*	
<input type=“submit” runat=“server”>	*HtmlInputButton*	
<input type=“checkbox” runat=“server”>	*HtmlInputcbeckBox*	
<input type=“file” runat=”server”>	*HtmlInputFile*	
<input type=“hidden” runat=“server”>	*HtmlInputHidden*	
<input type= “image” runat=“server”>	*HtmlInputImage*	
<input type=“radio” runat=“server”>	*HtmlInputRadioButton*	
<input type=“password” runat=“server”>	*HtmlInputText*	
<input type=“text” runat=“server”>	*HtmlInputText*	
<a runat= “server”>	*HtmlAnchor*	**HTML Container Control**
<form runat= “server”>	*HtmlForm*	
<select runat=“server”>	*HtmlSelect*	
<table runat= “server”>	*HtmlTable*	
<td runat=“server”>	*HtmlTableCell*	
<th runat=“server”>	*HtmlTableCell*	
<tr runat=“server”>	*HtmlTableRow*	
<textarea runat=“server”>	*HtmlTextArea*	
<img runat=“server”>	*HtmlImage*	
Any other html tag with *runat=“server”*	*HtmlGenericControl*	

Web Controls on Server Side



.NET Framework Class Library (FCL) provides the following six categories of Web Controls

- "Simple" controls, so called because (in general) they wrap simple HTML control tags
- Button controls, which create various types of buttons in Web forms
- List controls, which display simple lists of items
- Data-bound controls, which use data binding to display information obtained from databases and other data sources
- Calendar control, whose sole member, *Calendar*, adds interactive calendars to Web forms
- Validation controls, which validate user input before and after forms are submitted to the server

Example of Using HTML Control

```
<html>
<head runat="server">
    <title>Server Side Script</title>
</head> <body >
    <form id="Form1" RunAt = "Server" >
          <input id="Text1" type="text" RunAt = "Server" />
          +
          <input id="Text2" type="text" RunAt = "Server" />
          <input type="submit" value= " = "
                ID = "z"
                OnServerClick = "OnAdd"
                RunAt = "Server" />
    </form>
</body> </html>
<script language= "C#" runat="server">
    void OnAdd (Object sender, EventArgs e) {
    Int32 x = Convert.ToInt32(Text1.Value);
    Int32 y = Convert.ToInt32(Text2.Value);
    z.Value = Convert.ToString(x + y);
    }
</script>
```

Code separated from the html part

Example Using Web Controls

Default.aspx

```
<html>
<head runat="server"><title>Server Side Scripting</title></head>
   <body>
       <form runat="Server">
          <asp:TextBox ID= "x" RunAt="server" />
          +
          <asp:TextBox ID= "y" RunAt="server" />
          <asp:Button Text= " = " OnClick = "OnAdd" RunAt= "server" />
          <asp:Label ID= "z" RunAt="server" />
       </form>
   </body>
</html>
<script language= "C#" runat="server">
   void OnAdd (Object sender, EventArgs e) {
       Int32 a = Convert.ToInt32(x.Text);
       Int32 b = Convert.ToInt32(y.Text);
       z.Text = (a + b).ToString();
   }
</script>
```

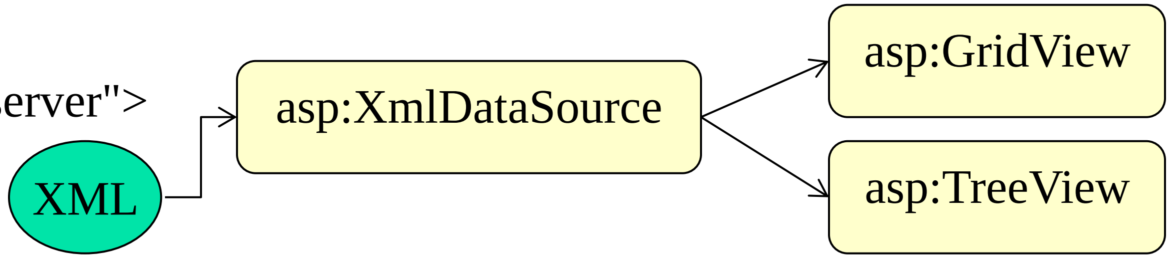
Another possible way

Default.aspx.cs

Can we display XML format without Writing an XSL program?

How?
Use Server Controls!
Are they better than my XSL program?

Using Web Controls to Display XML File



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"> <title>Display XML File</title> </head>
<body>
   <form id="form1" runat="server">
   <div>
   </div>
   <asp:XmlDataSource ID="XmlDataSource1" runat="server"
      DataFile="http://venus.sod.asu.edu/WSRepository/xml/Courses.xml"
      XPath="/Courses/Course/*"></asp:XmlDataSource>
   <asp:GridView ID="GridView1" runat="server" DataSourceID="XmlDataSource1">
   </asp:GridView>
   <asp:TreeView ID="TreeView1" runat="server" DataSourceID="XmlDataSource1">
   </asp:TreeView>
   </form>
</body>
</html>
```

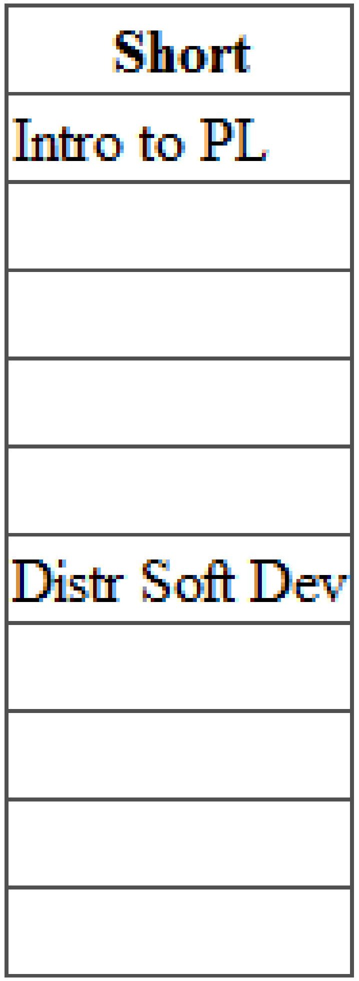
GridView and TreeView Web Controls

```
 <asp:GridView ID="GridView1" runat="server"
               DataSourceID="XmlDataSource1">
</asp:GridView>
```

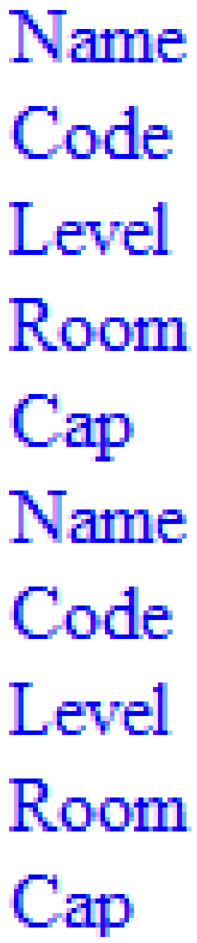
Display the contents at the top level of the tree

```
<asp:TreeView ID="TreeView1" runat="server"
               DataSourceID="XmlDataSource1">
</asp:TreeView>
```

Display the element names of the XML file



Short
Intro to PL
Distr Soft Dev

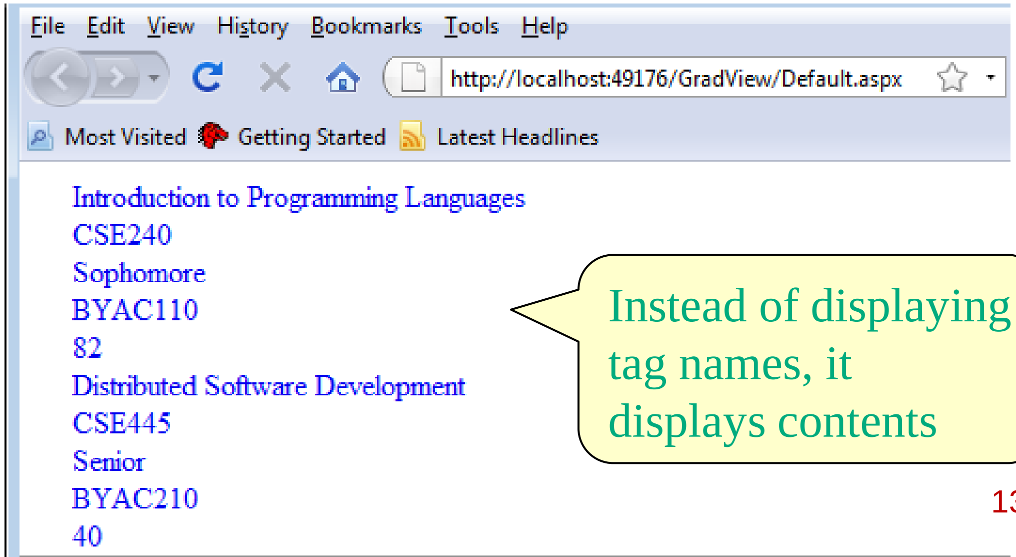


Name
Code
Level
Room
Cap
Name
Code
Level
Room
Cap

Using DataBinding Control to Display Contents

```
 <asp:XmlDataSource ID="XmlDataSource1" runat="server"
       DataFile="http://venus.sod.asu.edu/WSRepository/xml/Courses.xml"
       XPath="/Courses/Course/*"></asp:XmlDataSource>
<asp:TreeView ID="TreeView1" runat="server" DataSourceID="XmlDataSource1">
     <DataBindings>
           <asp:TreeNodeBinding DataMember="Name" TextField = "#InnerText" />
           <asp:TreeNodeBinding DataMember="Code" TextField = "#InnerText"/>
           <asp:TreeNodeBinding DataMember="Level" TextField = "#InnerText"/>
           <asp:TreeNodeBinding DataMember="Room" TextField = "#InnerText"/>
           <asp:TreeNodeBinding DataMember="Cap" TextField = "#InnerText"/>
     </DataBindings>
</asp:TreeView>
```

How can you display the tags and contents?


File Edit View History Bookmarks Tools Help
http://localhost:49176/GradView/Default.aspx
Most Visited Getting Started Latest Headlines

```
Introduction to Programming Languages
CSE240
Sophomore
BYAC110
82
Distributed Software Development
CSE445
Senior
BYAC210
40
```

Instead of displaying tag names, it displays contents

Ira A. Fulton Schools of Engineering

Arizona State University

M11 L5
Web User Controls

Ira A. Fulton Schools of Engineering
Arizona State University

Lecture Outline

- Web User controls
- ASCX Files
- ASCX File as Patch in ASPX File
- Examples

What is a (Web) User Control?

What can be used for building a Web application?

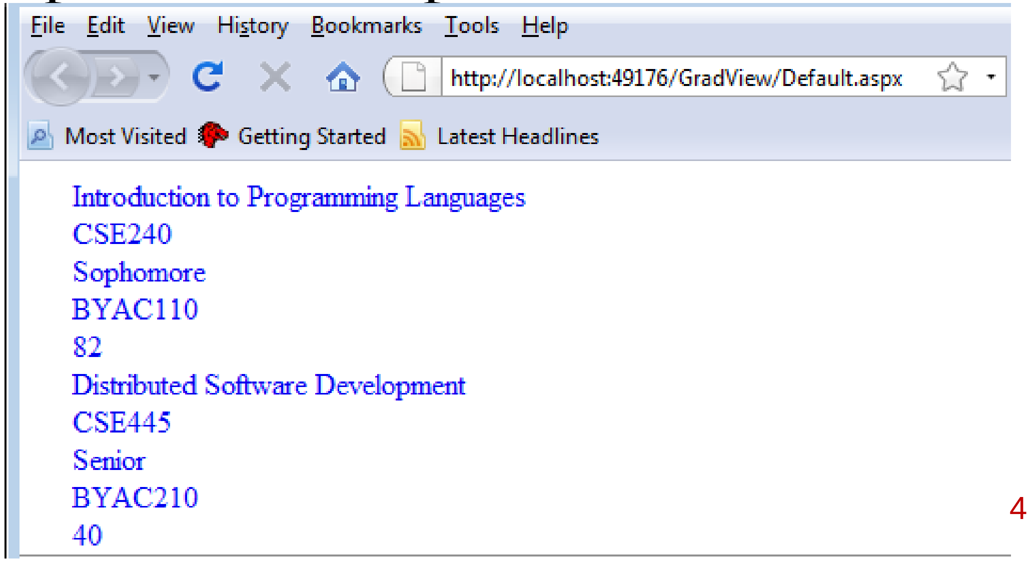
- Remote components
  - Web Services (WSDL and RESTful)
  - Other remote APIs
- Local components
  - Server Controls: GUI and code behind
  - (Web) User Control: GUI and Code component
    - Combine multiple server controls and even DLL functions
    - Functions and GUI items needed frequently
  - DLL: Code component only

What is a User Control in ASCX File

- A user control is a new "server control" that you build;
- It is a custom control (object) built from user code, HTML controls, and Web controls;
- It is a mechanism for building **reusable** ASP.NET components that can be shared by multiple pages.
- Without using users controls, you may need to repeat building the same component multiple times.

How can you display the tags and contents?

Write your user control to do it.


File Edit View History Bookmarks Tools Help
http://localhost:49176/GradView/Default.aspx
Most Visited
Getting Started
Latest Headlines

Introduction to Programming Languages
CSE240
Sophomore
BYAC110
82
Distributed Software Development
CSE445
Senior
BYAC210
40

How are User Controls Different?

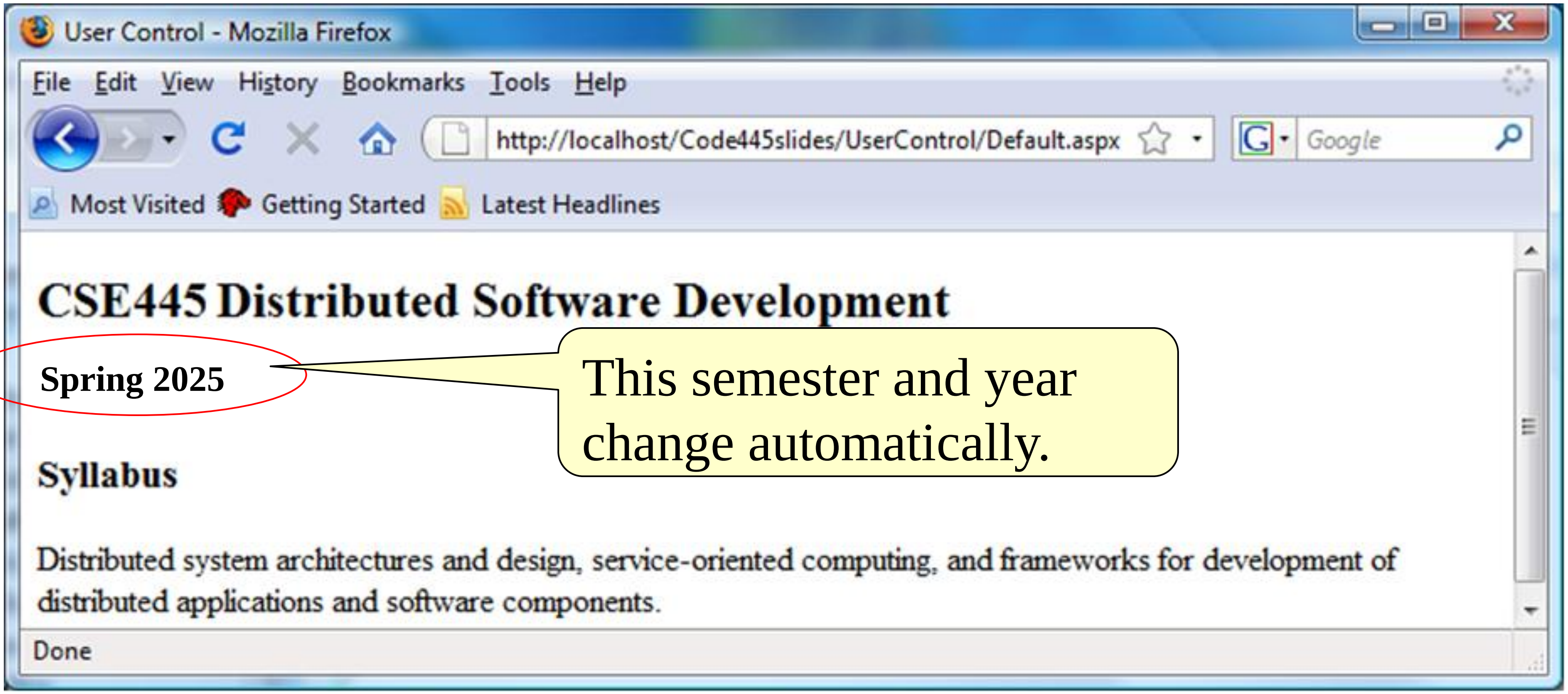
A User Control consists of a GUI file (.ascx) and a code behind file (.ascx.cs).

- Compared with .aspx form:
  A user control does not create a "form", and it cannot be directly accessed from a browser; Instead, it adds a user-controlled item (a patch) into an ASPX page.
- Compared with a Server Control:
  A Server Control does not have separate GUI and code files. They are parts within .aspx and .aspx.cs files.
- Compared with a DLL function:
  A user control is not pre-compiled, and the just-in-time compilation model can apply. A DLL function does not create GUI.

Example: Creating a User Control Item in an ASPX Page

1. Define what you want.
2. Add a User Control Item into your project.
3. Design User Control’s GUI : ascx Item.
4. Write Code behind ascx.cs Item.
5. Add the ascx into a .aspx page.

The Purpose: Make a Patch


User Control - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/Code445slides/UserControl/Default.aspx
Google
Most Visited Getting Started Latest Headlines
Done

CSE445 Distributed Software Development

Spring 2025

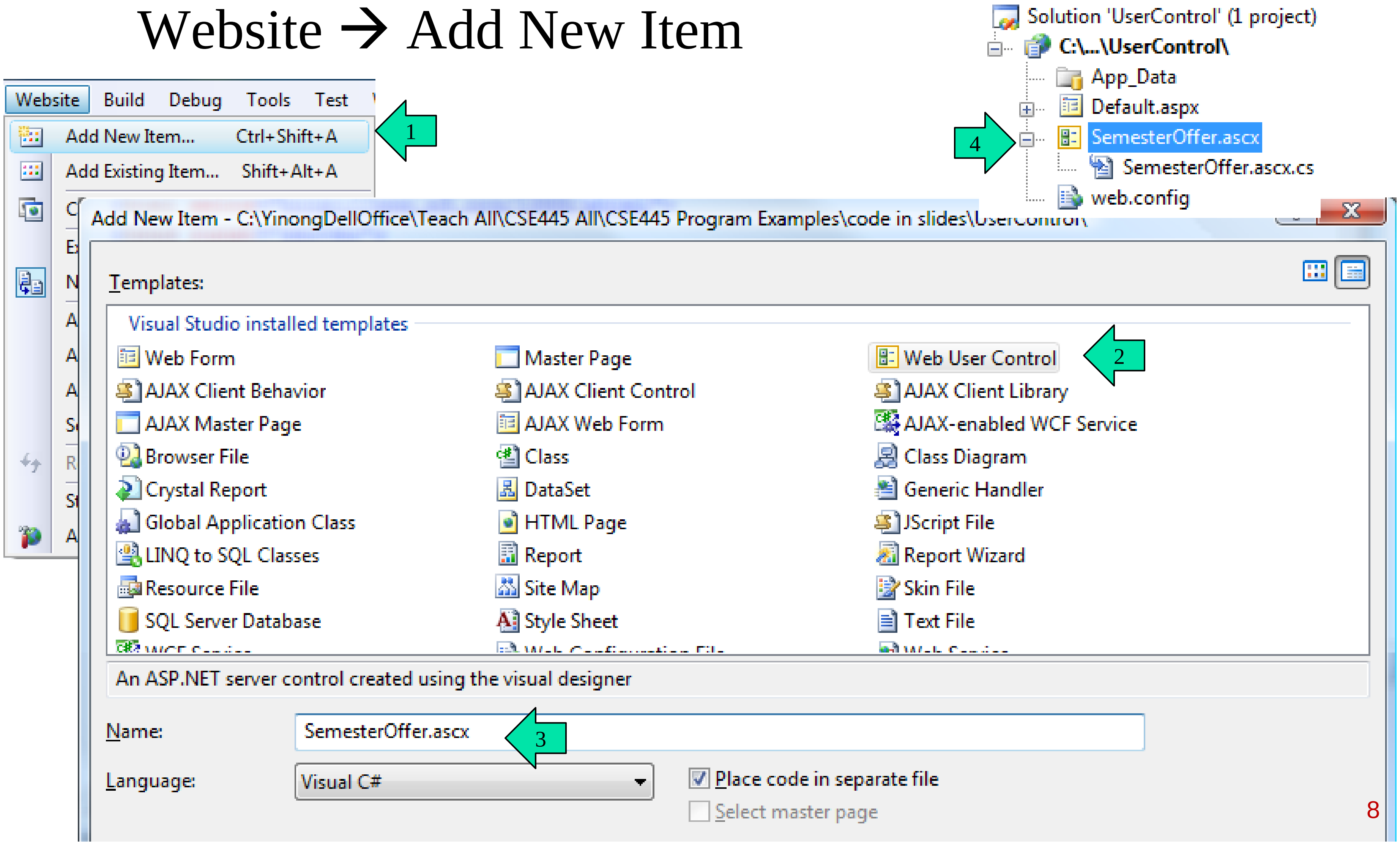
This semester and year change automatically.

Syllabus

Distributed system architectures and design, service-oriented computing, and frameworks for development of distributed applications and software components.

1. Adding a User Control Item

- After you opened your Website project, use VS menu: Website → Add New Item



An ASP.NET server control created using the visual designer

Name: SemesterOffer.ascx

Language: Visual C#

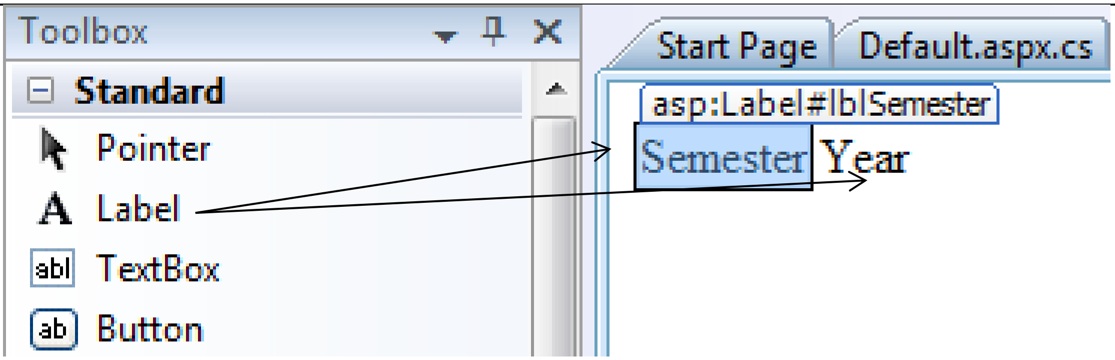
☑ Place code in separate file

☐ Select master page

2. Design User Control GUI : ascx Item

SemesterOffer.ascx

```
<%@ Control Language="C#"
    AutoEventWireup="true"
    CodeFile="WebUserControl.ascx.cs"
    Inherits="SemesterOffer" %>
<asp:Label ID="lblSemester" runat="server"
    Text="Semester"></asp:Label>
<asp:Label ID="lblYear" runat="server"
    Text="Year"></asp:Label>
```



3. Write Code behind ascx Item

SemesterOffer.ascx.cs

```
public partial class SemesterOffer : System.Web.UI.UserControl
{
      protected void Page_Load(object sender, EventArgs e)
      {
            String semester, year;
            Int32 m = DateTime.Now.Month;
            if (m <= 5) semester = "Spring";
            else if (m <= 7) semester = "Summer";
            else semester = "Fall";
            Int32 yr = DateTime.Now.Year;
            year = yr.ToString();
            lblSemester.Text = semester;
            lblYear.Text = year;
      }
}
```

4. Patch an **ascx** Item into an **aspx** Page

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<%@ Register TagPrefix = "cse" TagName="semester"
                    src="SemesterOffer.ascx" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"> <title>User Control</title> </head>
<body>
     <h2>CSE445/598 Distributed Software Development</h2>
          <form id="Form1" runat = "server">
                    <cse:semester runat = "server" />
          </form>
     <h3>Syllabus</h3>
Distributed system architectures and design, service-oriented computing, and
frameworks for development of distributed applications and software components.
     </body>
</html>
```

Putting All Together Example

Deploying a Web Application with a Web User Control

to a Server

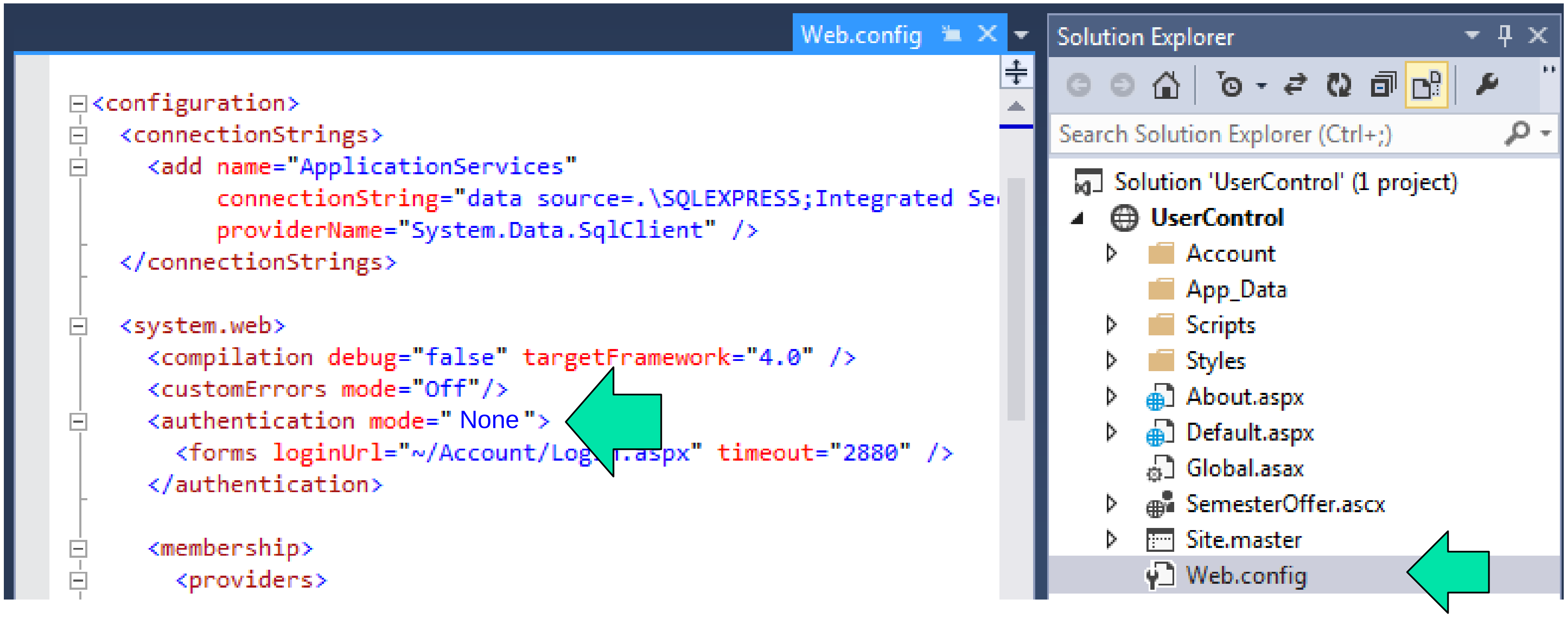
Before Deploying the Application into IIS

1. Open the Web.config file in Visual Studio and check
2. Change from

        <authentication mode=“Windows"/>
        <authentication mode="Forms"/>

   Chapter 6

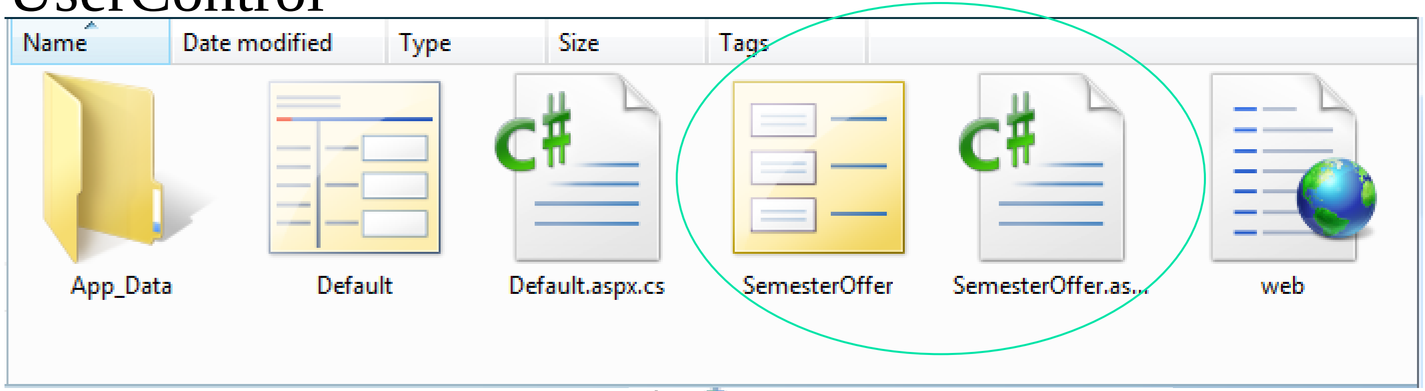
   To: <authentication mode="None"/>



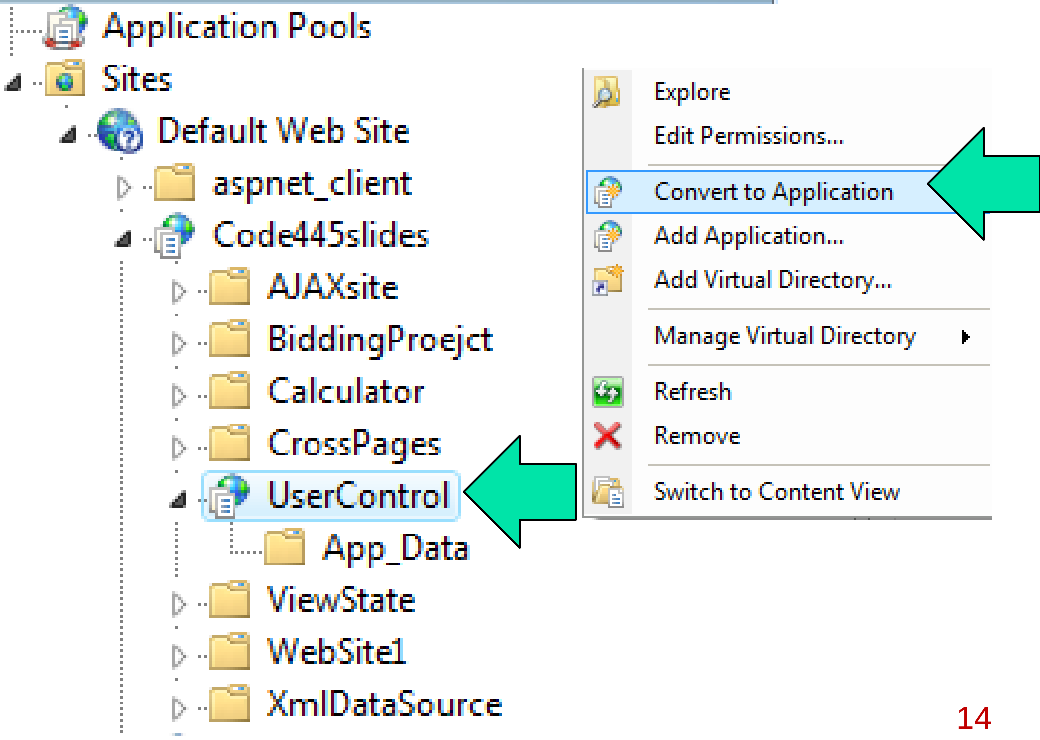
```
<configuration>
  <connectionStrings>
    <add name="ApplicationServices"
         connectionString="data source=.\SQLEXPRESS;Integrated Se
         providerName="System.Data.SqlClient" />
  </connectionStrings>

  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <customErrors mode="Off"/>
    <authentication mode=" None ">
      <forms loginUrl="~/Account/Log  .aspx" timeout="2880" />
    </authentication>

    <membership>
      <providers>
```

Folders and Files in the Application

UserControl

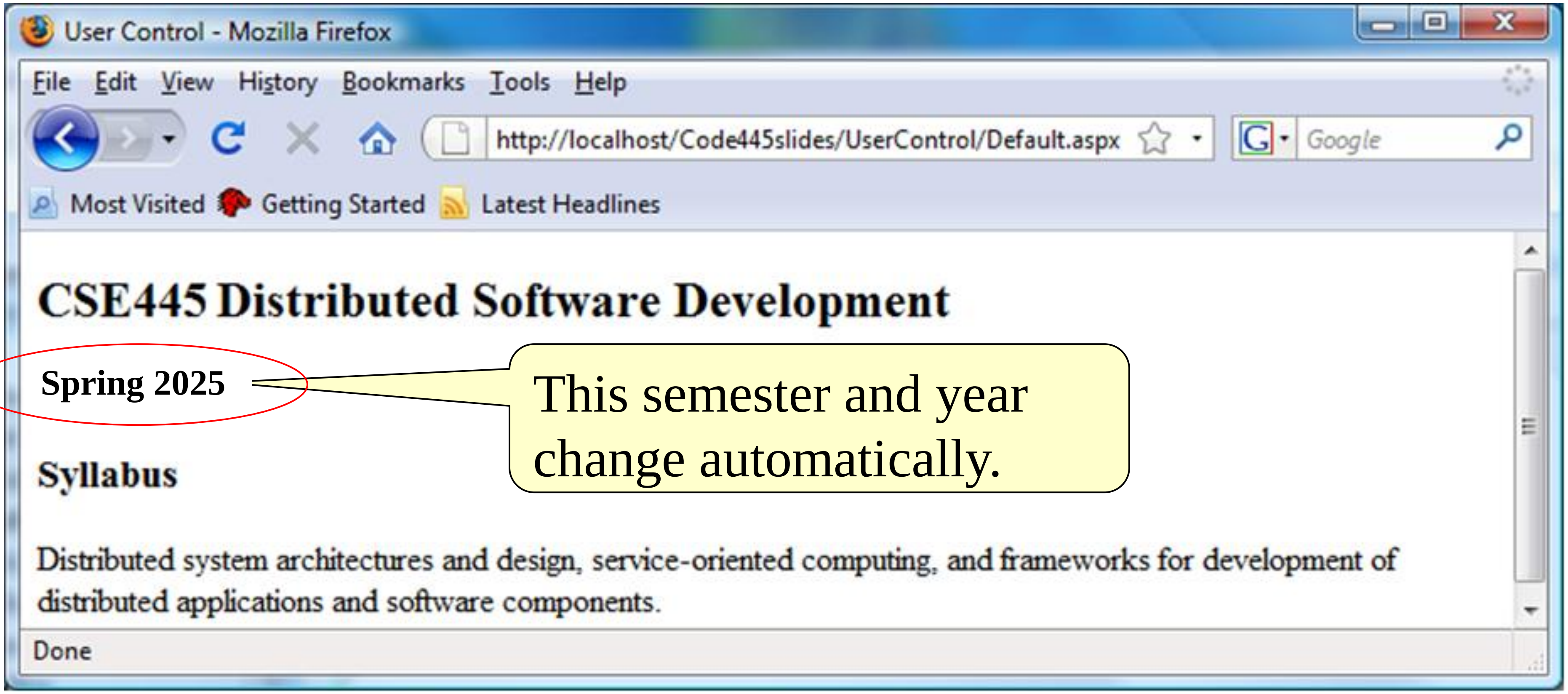


- Create a virtual directory in IIS and link it to this directory “UserControl” Application.
- In IIS: Right-click UserControl: and select: Convert to Application



Testing the .ascx Item in the aspx Page

In a Web browser, open:
http://localhost/Code445slides/UserControl/Default.aspx


User Control - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/Code445slides/UserControl/Default.aspx
Google
Most Visited
Getting Started
Latest Headlines
Done

CSE445 Distributed Software Development

Spring 2025

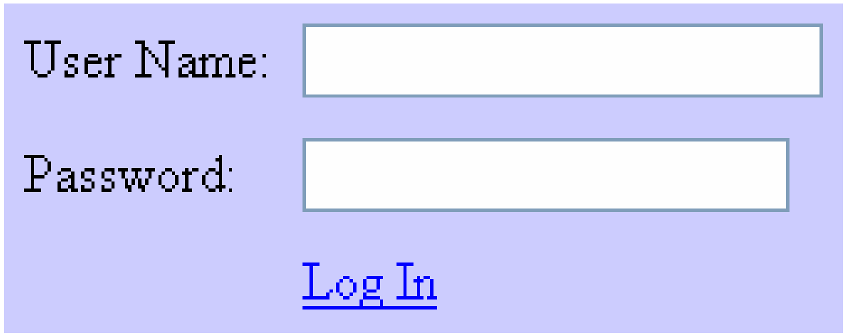
Syllabus

Distributed system architectures and design, service-oriented computing, and frameworks for development of distributed applications and software components.

This semester and year change automatically.

More Examples:
Creating more Complex User Control Item in an ASPX page

1. Creating a Login Page that can be shared in different ASPX pages.
2. How to share a User Control in different ASPX pages?
3. Creating a dynamic graph item and position it within a User Control



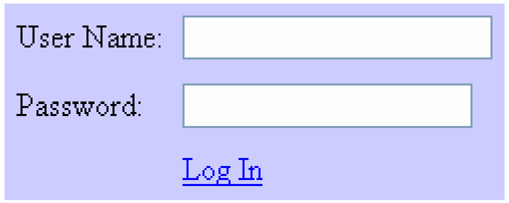
ASPX File Creating the Web Form *LoginPage.aspx*

```
<%@ Register TagPrefix=“user” TagName=“LoginControl” src=“LoginControl.ascx” %>
<html>
	<body>
		<h1>User Control Demonstration</h1>
		<hr>
		 <form runat="server">
			<user:LoginControl ID=“MyLogin” BackColor=“#ccccff” RunAt=“server” />
		</form>
		<hr>
		<h3><asp:Label ID=“Output” RunAt=“server” /></h3>
	</body>
</html>
<script language=“C#” runat=“server”>
	void Page_Load (Object sender, EventArgs e) {
		if (IsPostBack)
		Output.Text = “Hello, ” + MyLogin.UserName;
	}
</script>
```

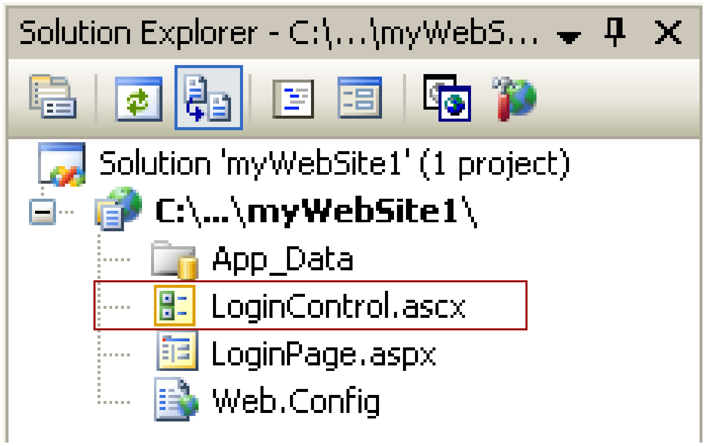
Another Example: *LoginControl.ascx*

LoginControl.ascx

```
<table id="MyTable" cellpadding="4" RunAt="server">
        <tr> <td>User Name:</td>
                <td><asp:TextBox ID=“MyUserName" RunAt="server" />
                </td>
        </tr>
         <tr> <td>Password:</td>
                <td><asp:TextBox ID=“MyPassword" TextMode="password" RunAt="server" />
                </td>
        </tr>
        <tr> <td></td> <td><asp:LinkButton Text="Log In" RunAt="server" /></td> </tr>
</table>
<script language="C#" RunAt="server">
        public string BackColor   {      get { return MyTable.BgColor; }
                                         set { MyTable.BgColor = value; }          }
        public string UserName    {      get { return MyUserName.Text; }
                                         set { MyUserName.Text = value; }          }
        public string Password    {      get { return MyPassword.Text; }
                                         set { MyPassword.Text = value; }          }
</script>
```



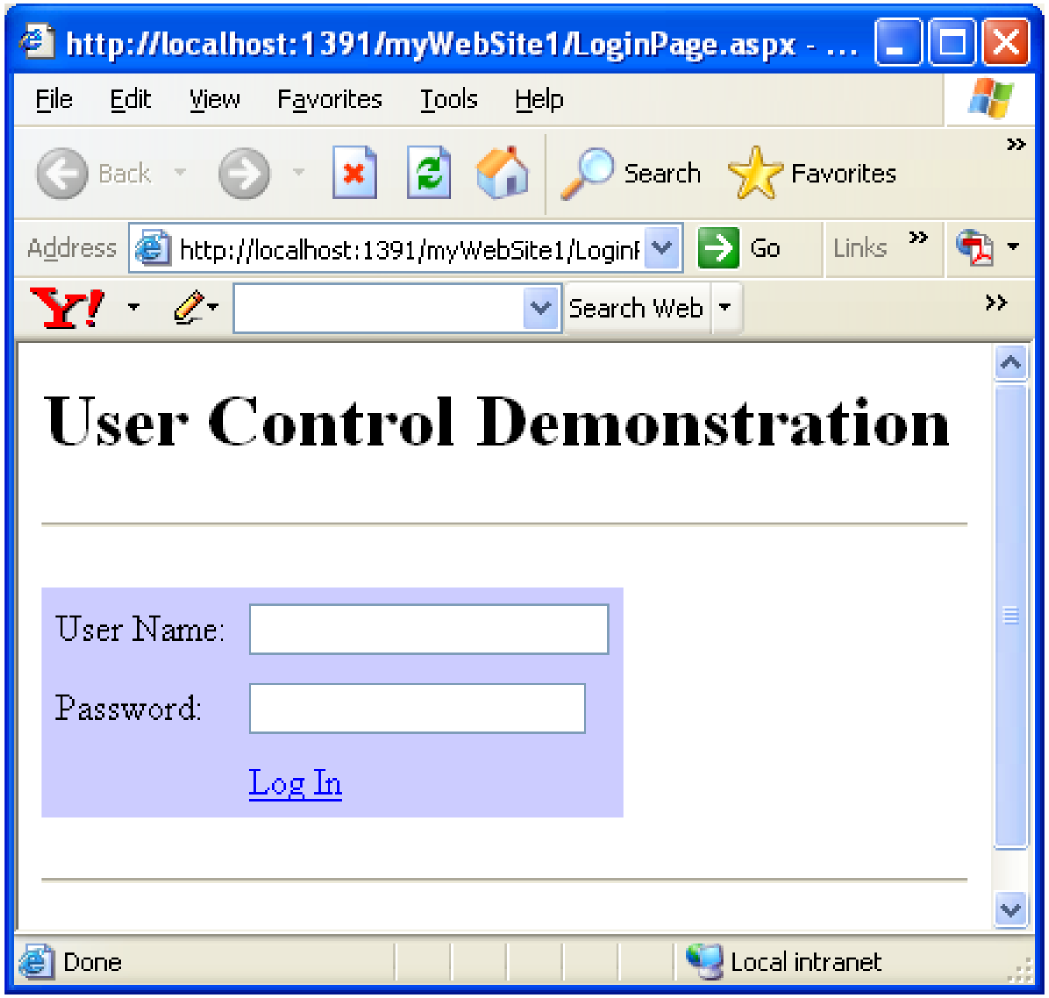
Putting ascx and aspx files into Website Project



Solution Explorer - C:\...\myWebS...

Solution 'myWebSite1' (1 project)
- C:\...\myWebSite1\
  - App_Data
  - LoginControl.ascx
  - LoginPage.aspx
  - Web.Config

Build
Start



http://localhost:1391/myWebSite1/LoginPage.aspx - ...

File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://localhost:1391/myWebSite1/LoginP
Go
Links
Search Web

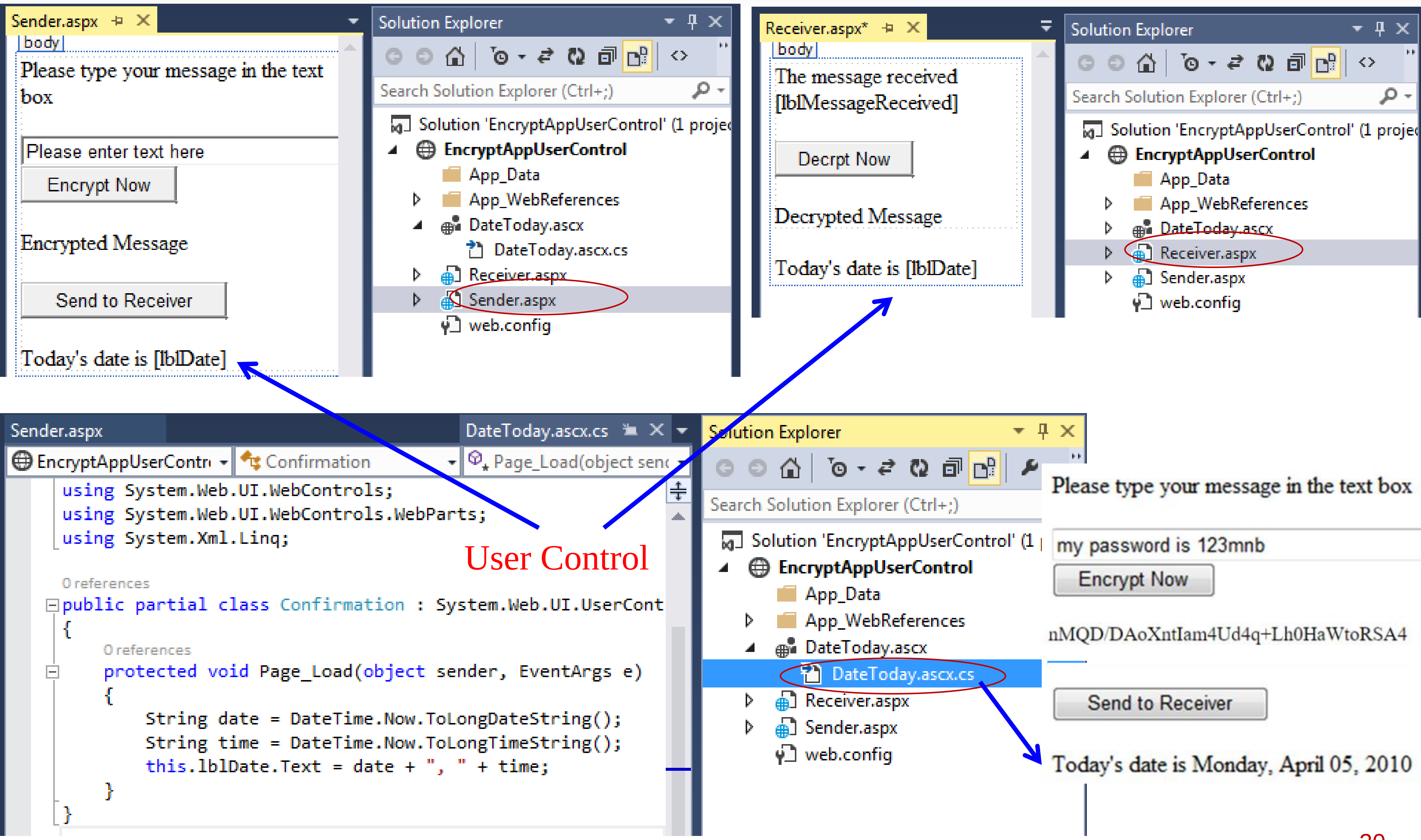
User Control Demonstration

User Name:

Password:

Log In

Sharing a User Control in Multiple Pages



Please type your message in the text box

Please enter text here

Encrypt Now

Encrypted Message

Send to Receiver

Today's date is [lblDate]

Solution 'EncryptAppUserControl' (1 projec
- EncryptAppUserControl
  - App_Data
  - App_WebReferences
  - DateToday.ascx
    - DateToday.ascx.cs
  - Receiver.aspx
  - Sender.aspx
  - web.config

The message received [lblMessageReceived]

Decrpt Now

Decrypted Message

Today's date is [lblDate]

```
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

0 references
public partial class Confirmation : System.Web.UI.UserCont
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        String date = DateTime.Now.ToLongDateString();
        String time = DateTime.Now.ToLongTimeString();
        this.lblDate.Text = date + ", " + time;
    }
}
```

Please type your message in the text box

my password is 123mnb

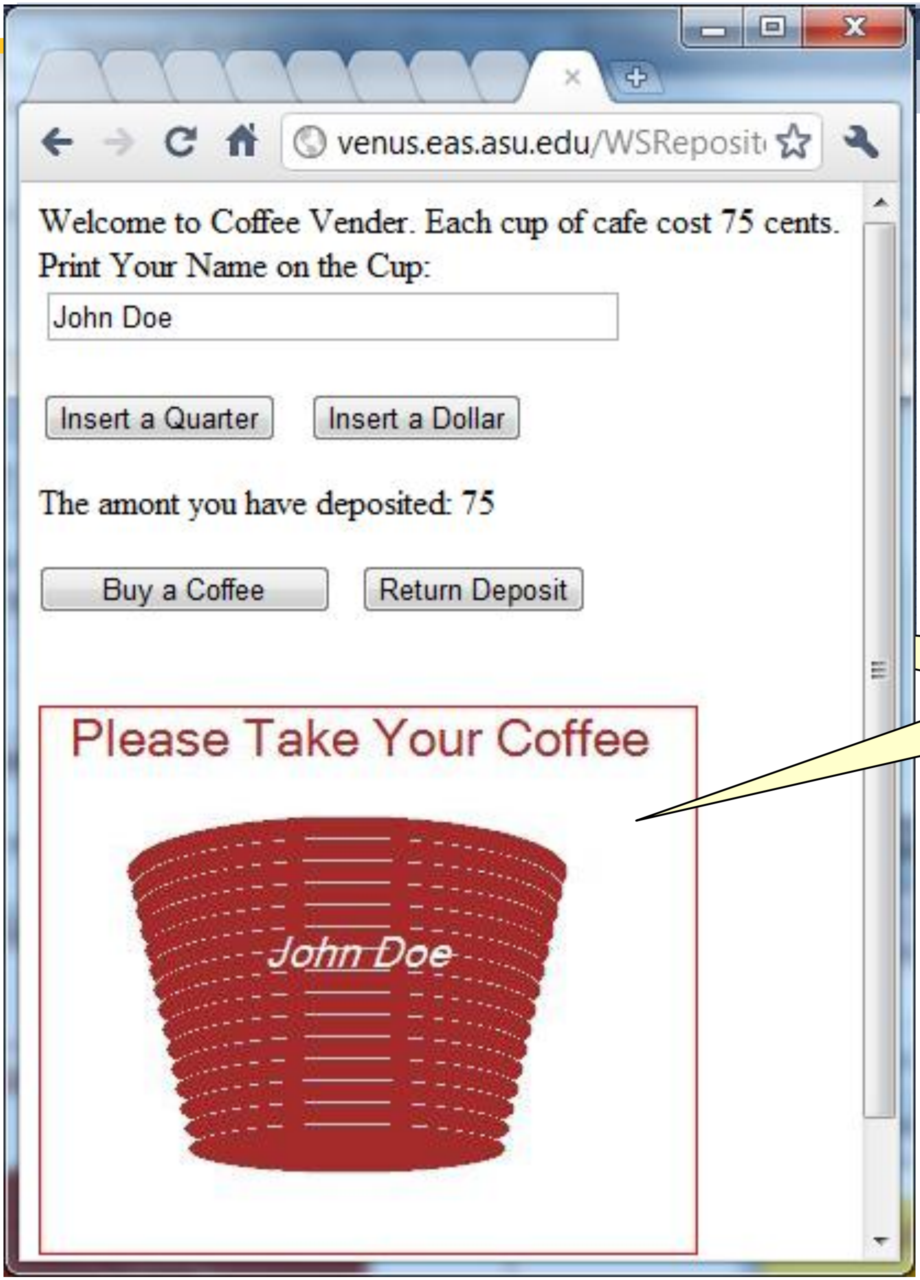
Encrypt Now

nMQD/DAoXntIam4Ud4q+Lh0HaWtoRSA4

Send to Receiver

Today's date is Monday, April 05, 2010

More complex user control example later

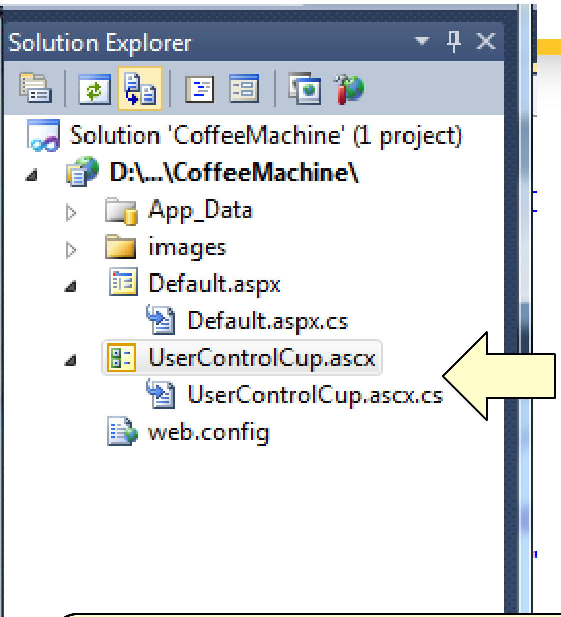

venus.eas.asu.edu/WSReposit

Welcome to Coffee Vender. Each cup of cafe cost 75 cents.

Print Your Name on the Cup:

John Doe

The amont you have deposited: 75



Display image generated by the user control here

http://venus.sod.asu.edu/WSRepository/CoffeeMachine/

Ira A. Fulton Schools of
Engineering
Arizona State University