1 **Web Forms and Input Controls**

CST272—ASP.NET

2 **ASP.NET Web Forms (Page 1)**

- All ASP.NET Web server controls must appear within an ASP.NET <form> tag (not just the input controls)
- The <form> tag must contain the runat="server" property and value which indicates that:
  - The form should be processed on the server
  - The enclosed controls can be accessed by server scripts
- Format:

  <form id="*FormID*" runat="server">

  ... HTML + ASP.NET server controls

  </form>

3 **ASP.NET Web Forms (Page 2)**

- When an ASP.NET form is submitted, it *always posts back to itself*:
  - If an action attribute is specified for a different Web page, it is ignored (action always is a postback)
  - If the method attribute is omitted, by default it will be set to method="post"
  - Also if the name and id attributes are not specified, they are automatically assigned by ASP.NET.

4 **Postback and Redirect Forms**

- Postback—the ability of a page to reload itself and process its own posted data
  - The values of the form are posted to the same page—postback is default operation when an <asp:Button> Web control (a submit button) is clicked
- Redirect—when a page calls another page
  - Data from the form is passed to the new page which in ASP.NET requires a Response.Redirect() statement in a Web control *event procedure*

5 **Page_Load and Postback (Page 1)**

- When an initial request for a page (a Web Form) is received by ASP.NET, it locates and loads requested Web Form (and if necessary compiles any C# code)
  - Initial page requests are relatively simple
- The default event handler method Page_Load for a Form is created automatically for a new Web Form in the C# code when the page is created

6 **Page_Load and Postback (Page 2)**

- The real work gets done when a postback request is generated (a page is submitted to itself):
  - The current value of every control on a Web Form, referred to as the Post Data, is contained in the postback request
  - The content of the ViewState which holds the original property values of every

control on a Web Form is also contained in the Post Data
  - If a postback was caused by an <asp:Button> Web control (a submit button) click, the Post Data also is used to identify the button that caused the postback

7 **The asp:TextBox Web Control (Page 1)**

- The asp:TextBox collects text input from users
- When the TextMode property is set to SingleLine, (*default* if not entered) creates a one-line textbox
- Format:

  <asp:TextBox id="*TextBoxID*" runat="server"
  TextMode="SingleLine"></asp:TextBox>
- The HTML rendered by ASP.NET for the one-line text box is:

  <input name="*name*" type="text" value="*initial text*" id="*id*" />

8 **The asp:TextBox Web Control (Page 2)**

- Properties:
  - (ID): its name (rendered by ASP.NET as both the name and id attributes in HTML)—prefix for the asp:TextBox Web control is TextBox
  - Text: the text displayed in the text box (updated as a user keys new or updated text)
- If a submit button is clicked and the page *posts back*, text in the asp:TextBox control is assigned as the value of the returned HTML input control

  <input type="text" value="*initial text*" />

9 **The Columns Property**

- The Columns property is set to shrink or expand the width of any style textbox from its default width
- Usually set to a width appropriate for the amount of input information expected
  - The Columns property is *relative*
- All textbox types also may be resized by dragging on their handles
  - This resets the Width property which is the textbox's *absolute* width measured in pixels (overrides Columns)
- There is no Rows property for either the single-line or password textbox (only one row)

10 **The MaxLength Property**

- Limits the number of characters that may be keyed into any textbox
- Appropriate for SingleLine and Password textboxes
  - If not set there is no limit
  - If the value is set to zero (0) which is the default, it means there is no restriction
- Renders in the HTML code as maxlength, e.g.:

  <input name="TextBoxLogin" type="text" value="What is your login?"
  maxlength="15" size="20" id=" TextBoxLogin " />

11 **The Formatting Properties for the asp:TextBox Web Controls (Page 1)**

- BackColor—color within the "box" that is behind the text
- BorderColor—color of the box (border)
- BorderStyle—dotted, dashed, etc.
- BorderWidth—thickness of the box (border) measure in pixels

12 **The Formatting Properties for the asp:TextBox Web Controls (Page 2)**

- Font-Names which is a comma delimited string of one or more typeface names
- Font-Size which is numeric and measured in points
- Font-Bold which is True or False
- Font-Italic which is True or False
- Font-Underline which is True or False
- ForeColor—color of the text

13 **The Formatting Properties for the asp:TextBox Web Controls (Page 3)**

- The formatting properties render together in the HTML code as a style property, e.g.:
  style="font-family:Comic Sans MS; font-size:12pt; font-weight:bold; background-color:#FFCCFF"

14 **The Multiline TextBox (Page 1)**

- Multiline textboxes exist in Web forms to allow users to enter lengthy narratives or comments
- Setting the TextMode property equal to MultiLine creates a multi-line textbox
- Format:
  <asp:TextBox id="*id*" runat="server" TextMode="MultiLine"></asp:TextBox>

15 **The Multiline TextBox (Page 2)**

- Vertical scrollbar automatically activates for scrolling through text when the user enters more lines than will fit into the multi-line textbox
- The HTML rendered by ASP.NET for the multi-line textbox is:
  <textarea name="*name*" *id*="*id*">*Text inside multiline text box*</textarea>

16 **The Columns and Rows Properties for the MultiLine TextBox**

- The Columns and Rows properties are set to create a wider textbox and allow for multiple rows of input
  – Rendered by ASP.NET as the cols and rows attributes in HTML
- Multi-line textbox may be resized on the Form in Design view by dragging on it handles
  – Only resets the Width and Height properties which represent its *absolute* size in pixels (default over the Columns and Rows properties)

17 **The Password TextBox (Page 1)**

- Hides the text entered by the users replacing each character displayed as an asterisk (*)

- Setting the TextMode property equal to Password, creates a password textbox
- Format:
  <asp:TextBox id="*id*" runat="server" TextMode="Password"> </asp:TextBox>

18 **The Password TextBox (Page 2)**

- When a submit button is clicked and the page is *posted back*, the password textbox is cleared
  – Text property of the asp:TextBox Web control is not rendered for security reasons
- Therefore the HTML rendered by ASP.NET for the password textbox is with no value attribute:
  <input name="*name*" type="password" id="*id*" />

20 **The asp:Button Web Control (Page 1)**

- The asp:Button Web server control is a *submit button* that calls a "click" event hander method (its main purpose) or reposts the Web page
- It functions as an HTML submit button (the form is submitted to the server)
  – A form is most often submitted by clicking on a button

21 **The asp:Button Web Control (Page 2)**

- The asp:Button server control in ASP.NET has the following format:
  – The control is used to display a push button
  – The push button may be a submit button or a command button (by default it is a submit button)
- Prefix for the ID name is Button, e.g.:
  ID="ButtonSubmit"

22 **The asp:Button Web Control (Page 3)**

- Format:
  <asp:Button ID="*ButtonID*" runat="server" Text="*Button Text*" />
  – The Text property is the label displayed on the button
  – A *one-sided* tag—requires tag end character (/) inside it
- Example:
  <asp:Button ID="ButtonClickMe" runat="server" Text="Calculate THR" />

23 **The asp:Button Web Control (Page 4)**

- The OnClick property links an asp:Button to a Click event handler (*method*)
- Format:
  <asp:Button ID="*ButtonID*" runat="server" Text="*Button Text*"
  OnClick="*MethodName*" />
- Example:
  <asp:Button ID="ButtonClickMe" runat="server" Text="Calculate THR"
  OnClick="ButtonClickMe_Click" />

24 **The Properties for an asp:Button Web Server Control**

- The properties for the asp:Button control include:
  - CausesValidation—if true (default) page validation occurs when the button is clicked
  - *CommandName*—the name of the event handler method that is called when the button is clicked, e.g. OnClick
  - PostBackUrl—the Web address (URL) of the page that is loaded and displayed when the button is clicked
  - Text—the text displayed on the button

25 **The asp:LinkButton Web Control (Page 1)**

- The asp:LinkButton Web server control is a "button" that displays text that looks like a hyperlink
- Although is behaves like the other button controls, its primary purpose is to request another Web page be loaded and viewed in the browser

26 **The asp:LinkButton Web Control (Page 2)**

- Format:
  <asp:LinkButton id="*LinkButtonID*" runat="server">*hyperlink text*</asp:LinkButton>
  - The *hyperlink text* is entered between the start and end asp:LinkButton tags
- Example:
  <asp:LinkButton ID="LinkButtonPage2" runat="server"
  PostBackUrl="WebForm2.aspx">Page 2 </asp:LinkButton>

27 **The Properties for an asp:LinkButton Web Server Control**

- The properties for the asp:LinkButton control include:
  - CausesValidation—if true (default) page validation occurs when the button is clicked
  - *CommandName*—the name of the event handler method that is called when the button is clicked, e.g. OnClick
  - PostBackUrl—the Web address (URL) of the page that is loaded and displayed when the button is clicked (the *primary purpose* of a link button)
  - Text—the text displayed on the button

28 **The asp:ImageButton Web Control (Page 1)**

- The asp:ImageButton is a button which displays an image on top of it (instead of text)
- It has all the same features of a Button in that it may call a "click" event hander (its main purpose) or repost the Web page

29 **The asp:ImageButton Web Control (Page 2)**

- The additional ImageURL property names the image (and its location) that is displayed on the button
- Format:
  <asp:ImageButton id="*ImageButtonID*" runat="server" ImageURL="path/filename"
  OnClick="*MethodName*" />

- Example:

```
<asp:ImageButton ID="ImageButtonGo" runat="server" ImageUrl="Images/go.jpg"
        OnClick="ImageButtonGo_Click" />
```

30 **The Properties for an asp:ImageButton Web Server Control**

- The properties for the asp:ImageButton control include:
  - CausesValidation—if true (default) page validation occurs when the button is clicked
  - *CommandName*—the name of the event handler method that is called when the button is clicked, e.g. OnClick
  - PostBackUrl—the Web address (URL) of the page that is loaded and displayed when the button is clicked
  - There is no Text property for an asp:ImageButton as the image replaces the text on the button

32 **The asp:DropDownList Web Control**

- The asp:DropDownList Web control creates a Form field that allows users to select from *lists* of options
- Format:

```
<asp:DropDownList id="id" runat="server"></asp:DropDownList>
```

- The HTML rendered by ASP.NET for the drop-down list is:

```
<select id="id"> ... </select>
```

33 **The Items Property (Collection) (Page 1)**

- A collection that represents the list of Items that appear in list of the asp:DropDownList Web control
- Select the DropDownList web control and then in the Properties window click the Build button [...] for the Items property
- Displays the ListItem Collection Editor
- Click:
  - <Add> button to enter list items on separate lines
  - <Remove> button to remove items from list
  - <↑> and <↓> buttons to modify item order in list

34 **The Items Property (Collection) (Page 2)**

- Properties for the list items:
  - Enabled—Boolean property which is True if the item is present in the list when it is displayed (the default)
  - Selected—Boolean property which is set to True if the item is to be selected by *default* when the Web page *initially* displays
  - Text—the string text displayed for that item in the list
  - Value—a string value stored for the DropDownList Web control whenever this is the selected item when the form is submitted for processing
    - If no value is entered, Text property assigned by default

35 **The Items Property (Collection) (Page 3)**

- For each item entered into the "ListItem Collection Editor", an asp:ListItem Web control is inserted into the asp:DropDownList block, e.g.

```
<asp:DropDownList id="DropDownListDepartments" runat="server">
    <asp:ListItem>Gifts</asp:ListItem>
    <asp:ListItem>Jewelry</asp:ListItem>
</asp:DropDownList>
```

36 **The SelectedItem Property for an asp:DropDownList Web Control (Page 1)**

- In C# the SelectedItem property represents the Item currently selected from asp:DropDownList control with the form is submitted
- Format:
  dropDownListID.SelectedItem

37 **The SelectedItem Property for an asp:DropDownList Web Control (Page 2)**

- A way of referencing properties of currently selected item within the "Code Editor" (C# code), e.g.:
  DropDownListID.SelectedItem.Text
  DropDownListID.SelectedItem.Value
- Example:

```
if (DropDownListDay.SelectedItem.Text = "Monday")
{
    ...
```

38 **The SelectedIndex Property for an asp:DropDownList Web Control**

- In C# the SelectedIndex property is a zero-based index value (integer) representing the currently selected item from the asp:DropDownList control
  – A value of minus one (-1) indicates no item is selected
- Format:
  dropDownListID.SelectedIndex
- Format:

```
if (DropDownListDay.SelectedIndex = 1)
{
    ...
```

  – The second item in the list ("Monday") is selected

39 **The AutoPostBack Property (Page 1)**

- A Boolean property used to indicate whether or not an automatic postback occurs when an event is raised for any ASP.NET Web control
- A postback occurs whenever a Web page submits information from its controls on the server and reloads itself with any changed information

40 **The AutoPostBack Property (Page 2)**

- Values for AutoPostBack:
  - True—an automatic postback will occur
  - False—an automatic postback will not occur (default)
- For example if a SelectedIndexChanged event fire for an asp:DropDownList control and AutoPostBack is set to True, a postback will occur
  - Normally only "Submit" button clicks generate a postback

42 **The asp:RadioButton Web Control (Page 1)**

- The asp:RadioButton Web control allows users to choose from a *mutually exclusive* list of options
  - If one radio button in a group is clicked and is "on", any other radio button in the group automatically turns "off"
- Format:
  <asp:RadioButton ID="*RadioButtonID*" runat="server" />
  - A one-sided tag so it should include a slash (/) meaning the "end" symbol inside the tag itself

43 **The asp:RadioButton Web Control (Page 2)**

- The HTML rendered by ASP.NET for the radio box control is:
  <input name="*name*" type="radio" id="*id*" />

44 **The Text Property for the asp:RadioButton Web Control**

- The Text property of the asp:RadioButton Web control is a string that controls the text which is displayed to the right of each radio button
  - It is not necessary to place a *separate label* next to a radio button
- Example:
  <asp:RadioButton ID="RadioButtonGifts" runat="server" Text="Gifts" />

45 **The GroupName Property for the asp:RadioButton Web Control**

- The GroupName property of the asp:RadioButton Web control is a string that is used for providing *grouping* for sets of radio buttons
  - Provides the functionality so that when one radio button in a group is clicked "on", any other radio button in the group turns "off" automatically
  - All radio buttons in the "group" have the same value assigned to their GroupName property
- Example:
  <asp:RadioButton ID="RadioButtonGifts" runat="server" Text="Gifts"
  GroupName="Department" />

46 **The Checked Property for the asp:RadioButton Web Control**

- The Checked property of the asp:RadioButton Web control is a Boolean property that determines if a radio button is "on" or "off"
- It often is used in the "Code Editor" (C# code) to evaluate which radio button is selected at the time the form was submitted

- Example:
  if (RadioButtonGifts.Checked) {
    …

47 **The asp:Image Web Control**

- Displays an image file with dynamic property functionality in an ASP.NET web page
- Example:
  <asp:Image ID="*ImageName*" runat="server" ImageURL("*path/filename*") />
  - A *single-sided* ASP.NET server controls—requires tag end character (/) inside it
  - Prefix for ID name is Image, e.g. "ImageFeatured"
  - Renders as an <img> tag in the HTML page generated by ASP.NET

48 **The ImageURL Property of the asp:Image Web Control**

- The ImageURL property names the path and filename of the image file to be displayed in the control
- Example:
  <asp:Image ID="ImageFeatured" ImageURL="Images/21.jpg" runat="server" />
  - ASP.NET renders this property as the src attribute in the HTML document
  - If this property is not set, the control is displayed in the "Designer" window as an icon with a red square, green circle, and blue triangle

50 **The asp:RadioButtonList Web Control**

- Follows the RadioButton in the Toolbox
- Used to populate the radio buttons when the options are stored in a database
- (Discussed in a later chapter)

51 **The asp:CheckBox Web Control (Page 1)**

- The asp:CheckBox Web control allows users to choose from one or several items in a list of options
  - If one checkbox is clicked "on", it has no effect on other checkboxes on the Form
- Format:
  <asp:CheckBox ID="*ControlName*" runat="server" />
  - A one-sided tag so it should include a slash (/) meaning the "end" symbol inside the tag itself

52 **The asp:CheckBox Web Control (Page 2)**

- The HTML rendered by ASP.NET for the checkbox control is:
  <input name="*ControlName*" type="checkbox" id="*ControlName*" />

53 **The Text Property for the asp:CheckBox Web Control**

- The Text property of the asp:CheckBox Web control is a string that controls the text which is displayed to the right of each checkbox
  - It is not necessary to place a *separate label* next to a textbox
- Example:

```
<asp:CheckBox ID="CheckBoxGifts" runat="server" Text="Gifts" />
```

54 **The Checked Property for the asp:CheckBox Web Control**

- The Checked property of the asp:CheckBox Web control is a Boolean property that determines if a checkbox "on" or "off"
- It often is used in the "Code Editor" (C# code) to evaluate which checkboxes are selected at the time the form was submitted
- Example:

```
if (CheckBoxGifts.Checked) {
    ...
```

56 **The asp:CheckBoxList Web Control**

- Follows the CheckBox in the Toolbox
- Used to populate the check boxes when the options are stored in a database
- (Discussed in a later chapter)