**How information about the user's locale can be accessed?**

The information regarding a user's locale can be accessed by using the *System.Web.UI.Page.Culture* property.

**Can you set which type of comparison you want to perform by the *CompareValidator* control?**

Yes, by setting the *Operator* property of the *CompareValidator* control.

**Which two new properties are added in ASP.NET 4.0 Page class?**

The two new properties added in the Page class are *MetaKeyword* and *MetaDescription*.

**How does a content page differ from a master page?**

A content page does not have complete HTML source code; whereas a master page has complete HTML source code inside its source file.

**What is the default timeout for a Cookie?**

The default time duration for a Cookie is 30 minutes.

**Explain file-based dependency and key-based dependency.**

In file-based dependency, you have to depend on a file that is saved in a disk. In key-based dependency, you have to depend on another cached item.

**Explain login controls.**

Login controls are built-in controls in ASP.Net for providing a login solution to ASP.NET application. The login controls use the membership system to authenticate a user credentials for a Web site.  
  
There are many controls in login controls.

* *ChangePassword* control - Allows users to change their password.
* *CreateUserWizard* control - Provides an interface to the user to register for that Web site.
* *Login* control - Provides an interface for user authentication. It consists of a set of controls, such as *TextBox*, *Label*, *Button*, *CheckBox*, *HyperLink*.
* *LoginView* control - Displays appropriate information to different users according to the user's status.
* *LoginStatus* control - Shows a login link to users, who are not authenticated and logout link, who are authenticated
* *LoginName* control - Displays a user name, if the user logs in.
* *PasswordRecovery* control - Allows users to get back the password through an e-mail, if they forget.

**What setting must be added in the configuration file to deny a particular user from accessing the secured resources?**

To deny a particular user form accessing the secured resources, the *web.config* file must contain the following code:   
  
*<authorization >  
<deny users="username" />  
</authorization>*

**Describe the complete lifecycle of a Web page.**

When we execute a Web page, it passes from the following stages, which are collectively known as Web page lifecycle:

* **Page request** - During this stage, ASP.NET makes sure the page either parsed or compiled and a cached version of the page can be sent in response
* **Start** - During this stage sets the Request and Response page properties and the page check the page request is either a postback or a new request
* **Page Initialization** - During this stage, the page initialize and the control's Unique Id property are set
* **Load** - During this stage, if the request is postback, the control properties are loaded without loading the view state and control state otherwise loads the view state
* **Validation** - During this stage, the controls are validated
* **Postback event handling** - During this stage, if the request is a postback, handles the event
* **Rendering** - During this stage, the page invokes the Render method to each control for return the output
* **Unload** - During this stage, when the page is completely rendered and sent to the client, the page is unloaded.

**How can you enable impersonation in the *web.config* file?**

To enable impersonation in the *web.confing* file, you need to include the *<identity>* element in the *web.config* file and set the impersonate attribute to true as shown in the following code snippet:  
*<identity impersonate = "true" />*

**In which database is the information, such as membership, role management, profile, and Web parts personalization, stored?**

The *aspnetdb* database stores all information.

**What is State Management? How many ways are there to maintain a state in .NET?**

State management is used to store information requests. The state management is used to trace the information or data that affect the state of the applications.  
  
There are two ways to maintain a state in .NET, Client-Based state management and Server-Based state management.  
  
The following techniques can be used to implement the Client-Based state management:

* View State
* Hidden Fields
* Cookies
* Query Strings
* Control State

The following techniques can be used to implement Server-Based state management:

* Application State
* Session State
* Profile Properties

**How can you ensure that no one has tampered with *ViewState* in a Web page?**

To ensure that no one has tampered with *ViewState* in a Web page, set the *EnableViewStateMac* property to *True*.

**What are the major built-in objects in ASP.NET?**

The major built-in objects in ASP.NET are as follows:

* *Application*
* *Request*
* *Response*
* *Server*
* *Session*
* *Context*
* *Trace*

**What is the function of the *CustomValidator* control?**

It provides the customize validation code to perform both client-side and server-side validation.

**Why a *SiteMapPath* control is referred to as breadcrumb or eyebrow navigation control?**

The *SiteMapPath* control displays a hierarchical path to the root Web page of the Web site. Therefore, it is known as the breadcrumb or eyebrow navigation control.

**Explain the validation controls. How many validation controls in ASP.NET 4.0?**

Validation controls are responsible to validate the data of an input control. Whenever you provide any input to an application, it performs the validation and displays an error message to user, in case the validation fails.  
  
ASP.NET 4.0 contains the following six types of validation controls:

* *CompareValidator* - Performs a comparison between the values contained in two controls.
* *CustomValidator* - Writes your own method to perform extra validation.
* *RangeValidator*- Checks value according to the range of value.
* *RegularExpressionValidator* - Ensures that input is according to the specified pattern or not.
* *RequiredFieldValidator* - Checks either a control is empty or not.
* *ValidationSummary* - Displays a summary of all validation error in a central location.

**How many types of Cookies are available in ASP.NET?**

There are two types of Cookies available in ASP.NET:

* **Session Cookie** - Resides on the client machine for a single session until the user does not log out.
* **Persistent Cookie** - Resides on a user's machine for a period specified for its expiry, such as 10 days, one month, and never.

The user can set this period manually.

**What are the Culture and UICulture values?**

The Culture value determines the functions, such as Date and Currency, which are used to format data and numbers in a Web page. The UICulture value determines the resources, such as strings or images, which are loaded for a Web page in a Web application.

**What is the use of the *<sessionState>* tag in the *web.config* file?**

The *<sessionState>* tag is used to configure the session state features. To change the default timeout, which is 20 minutes, you have to add the following code snippet to the web.config file of an application: *<sessionState timeout="40"/>*

**What events are fired when a page loads?**

The following events fire when a page loads:

* *Init()* - Fires when the page is initializing.
* *LoadViewState()* - Fires when the view state is loading.
* *LoadPostData()* - Fires when the postback data is processing.
* *Load()* - Fires when the page is loading.
* *PreRender()* - Fires at the brief moment before the page is displayed to the user as HTML.
* *Unload()* - Fires when the page is destroying the instances of server controls.

**Write three common properties of all validation controls.**

Three common properties of validation controls are as follows:

* *ControlToValidate* - Provides a control to validate
* *ErrorMessage* - Displays an error message
* *IsValid* - Specifies if the control's validation has succeeded or not
* *Text* - Displays a text for validation control before validation

What are navigation controls? How many navigation controls are there in ASP.NET 4.0?

Navigation controls help you to navigate in a Web application easily. These controls store all the links in a hierarchical or drop-down structure; thereby facilitating easy navigation in a Web application.  
  
**There are three navigation controls in ASP.Net 4.0.**

* *SiteMapPath*
* *Menu*
* *TreeView*

**How can we provide the WebParts control functionality to a server control?**

We can provide the *WebParts* controls functionality to a server control by setting the *CreateWebPart* property of *WebPartManger*.

**How do you prevent a validation control from validating data at the client end?**

You can prohibit a validation control to validate data at the client side by setting the *EnableClientScript* property to *False*.

**What is cross-page posting in ASP.NET?**

The *Server.Transfer()* method is used to post data from one page to another. In this case, the URL remains the same. However, in cross page posting, data is collected from different Web pages and is displayed on a single page. To do so, you need to set the *PostBackUrl* property of the control, which specifies the target page. In the target page, you can access the *PreviousPage* property. For this, you need to use the *@PreviousPageType* directive. You can access the controls of previous page by using the *FindControl()* method.

|  |
| --- |
| How do you turn off cookies for one page in your site? |
| Use Cookie.Discard property, Gets or sets the discard flag set by the server. When true, this property instructs the client application not to save the Cookie on the user's hard disk when a session ends. | |

|  |
| --- |
| Which two properties are on every validation control? |
| We have two common properties for every validation controls:   * Control to Validate * Error Message | |

|  |
| --- |
| What are different types of Assemblies? |
| * Single file and multi file assembly. * Assemblies can be static or dynamic. * Private assemblies and shared assemblies. | |

|  |  |
| --- | --- |
| Which namespaces are used for data access? | |
| * System.Data * System.Data.OleDB * System.Data.SQLClient |

|  |  |
| --- | --- |
| What is the base class of .net? | |
| System.object |

|  |
| --- |
| What is a HashTable? |
| The Hashtable object contains items in key/value pairs. The keys are used as indexes, and very quick searches can be made for values by searching through their keys. | |

|  |
| --- |
| What is CAS or Code Access Security? |
| **Code Access Security** - CAS is the part of the .NET security model that determines whether or not a piece of code is allowed to run, and what resources it can use when it is running. | |

|  |
| --- |
| What is Side-by-Side Execution? |
| The CLR allows any versions of the same-shared DLL (shared assembly) to execute at the same time, on the same system, and even in the same process. This concept is known as side-by-side execution. | |

|  |
| --- |
| What are the layouts of ASP.NET Pages? |
| * GridLayout * FlowLayout   GridLayout positions the form object on absolute x and y co-ordinates of the screen.  FlowLayout positions the form objects relative to each other. | |

|  |
| --- |
| What methods are fired during the page load? |
| * Init() - when the page is instantiated. * Load() - when the page is loaded into server memory. * PreRender() - the brief moment before the page is displayed to the user as HTML. * Unload() - when page finishes loading. | |

|  |
| --- |
| What is the difference between HTTP-Post and HTTP-Get? |
| The GET method creates a query string and appends it to the script's URL on the server that handles the request.  The POST method creates a name/value pairs that are passed in the body of the HTTP request message. | |

|  |
| --- |
| What is Marshalling? |
| Marshaling is a process of making an object in one process (the server) available to another process (the client). There are two ways to achieve the marshalling.   * Marshal by value * Marshal by reference. | |

**In which event of page cycle is the ViewState available?**

   After the Init() and before the Page\_Load().

**How you can add an event handler?**

Usingthe Attributes property of server side control.

e.g.

[csharp]

btnSubmit.Attributes.Add(“onMouseOver”,”JavascriptCode();”)

[/csharp]

**Which type if caching will be used if we want to cache the portion of a page instead of whole page?**

**Fragment Caching:** It caches the portion of the page generated by the request. For that, we can create user controls with the below code:

[xml]

<%@ OutputCache Duration=”120″ VaryByParam=”CategoryID;SelectedID”%>

[/xml]

**List the events in page life cycle.**

1) Page\_PreInit  
2) Page\_Init  
3) Page\_InitComplete  
4) Page\_PreLoad  
5) Page\_Load  
6) Page\_LoadComplete  
7) Page\_PreRender  
8)Render

**Can we add code files of different languages in App\_Code folder?**

 No. The code files must be in same language to be kept in App\_code folder.

**How can we apply Themes to an asp.net application?**

We can specify the theme in web.config file. Below is the code example to apply theme:

[xml]

<configuration>

<system.web>

<pages theme=”Windows7″ />

</system.web>

</configuration>

[/xml]

**What is the difference between ref & out parameters?**

An argument passed as ref must be initialized before passing to the method whereas out parameter needs not to be initialized before passing to a method.

**Describe the accessibility modifier “protected internal”.**

Protected Internal variables/methods are accessible within the same assembly and also from the classes that are derived from this parent class.

**What is the difference between Finalize() and Dispose() methods?**

Dispose() is called when we want for an object to release any unmanaged resources with them. On the other hand Finalize() is used for the same purpose but it doesn’t assure the garbage collection of an object.

**What are the different ways a method can be overloaded?**

Methods can be overloaded using different data types for parameter, different order of parameters, and different number of parameters.

**What is difference between is and as operators in c#?**

“is” operator is used to check the compatibility of an object with a given type and it returns the result as Boolean.

“as” operator is used for casting of object to a type or a class.

**What’s a multicast delegate?**

A delegate having multiple handlers assigned to it is called multicast delegate. Each handler is assigned to a method.

**What is difference between the “throw” and “throw ex” in .NET?**

“Throw” statement preserves original error stack whereas “throw ex” have the stack trace from their throw point. It is always advised to use “throw” because it provides more accurate error information.

**Explain how to access Viewstate values of this page in the next page?**

-PreviousPage property is set to the page property of the nest page to get the value of viewstate for the page in the next page.  
  
Page employee = this.PreviousPage;  
  
-Then a control can be found from the previous page and its state can be read.  
  
Label employeeLabel = poster.findControl("empLabel");  
string lbl = employeeLabel.Text;