**Explain the Page Life cycle of ASP.NET?**

**PreInit** – This event represents the entry point of the page life cycle. If you need to change the Master page or theme programmatically, then this would be the event to do so. Dynamic controls are created in this event.

**Init** – Each control in the control collection is initialized.

**Init Complete**\* - Page is initialized and the process is completed.

**PreLoad**\* - This event is called before the loading of the page is completed.

**Load** – This event is raised for the Page and then all child controls. The controls properties and view state can be accessed at this stage. This event indicates that the controls have been fully loaded.

**LoadComplete**\* - This event signals indicates that the page has been loaded in the memory. It also marks the beginning of the rendering stage.

**PreRender** – If you need to make any final updates to the contents of the controls or the page, then use this event. It first fires for the page and then for all the controls.

**PreRenderComplete**\* - Is called to explicitly state that the PreRender phase is completed.

**SaveStateComplete**\* - In this event, the current state of the control is completely saved to the ViewState.

**Unload** – This event is typically used for closing files and database connections. At times, it is also used for logging some wrap-up tasks.

**Explain the AutoPostBack feature in ASP.NET?**

AutoPostBack allows a control to automatically postback when an event is fired. For eg: If we have a Button control and want the event to be posted to the server for processing, we can set AutoPostBack = True on the button.

**Explain the ViewState in ASP.NET?**

Http is a stateless protocol. Hence the state of controls is not saved between postbacks. Viewstate is the means of storing the state of server side controls between postbacks. The information is stored in HTML hidden fields. In other words, it is a snapshot of the contents of a page.

**Explain the Validation Controls used in ASP.NET 2.0?**

Validation controls allows you to validate a control against a set of rules. There are 6 different validation controls used in ASP.NET 2.0.

RequiredFieldValidator – Checks if the control is not empty when the form is submitted.

CompareValidator – Compares the value of one control to another using a comparison operator (equal, less than, greater than etc).

RangeValidator – Checks whether a value falls within a given range of number, date or string.

RegularExpressionValidator – Confirms that the value of a control matches a pattern defined by a regular expression. Eg: Email validation.

CustomValidator – Calls your own custom validation logic to perform validations that cannot be handled by the built in validators.

ValidationSummary – Show a summary of errors raised by each control on the page on a specific spot or in a message box.

**What are Master Pages?**

Master pages is a template that is used to create web pages with a consistent layout throughout your application. Master Pages contains content placeholders to hold page specific content. When a page is requested, the contents of a Master page are merged with the content page, thereby giving a consistent layout.

**How is a Master Page different from an ASP.NET page?**

The MasterPage has a @Master top directive and contains ContentPlaceHolder server controls. It is quiet similar to an ASP.NET page.

**How do you attach an existing page to a Master page?**

By using the MasterPageFile attribute in the @Page directive and removing some markup

**What is a User Control?**

User controls are reusable controls, similar to web pages. They cannot be accessed directly.

**Explain briefly the steps in creating a user control?**

* Create a file with .ascx extension and place the @Control directive at top of the page.
* included the user control in a Web Forms page using a @Register directive

**Where do you store your connection string information?**

The connection string can be stored in configuration files (web.config).

**What is the difference between ‘Web.config’ and ‘Machine.config’?**

Web.config files are used to apply configuration settings to a particular web application whereas machine.config file is used to apply configuration settings for all the websites on a web server.

Web.config files are located in the application's root directory or inside a folder situated in a lower hierarchy. The machine.config is located in the Windows directory Microsoft.Net\Framework\*Version*\CONFIG.

There can be multiple web.config files in an application nested at different hierarchies. However there can be only one machine.config file on a web server.

**What is the difference between Server.Transfer and Response.Redirect?**

Response.Redirect involves a roundtrip to the server whereas Server.Transfer conserves server resources by avoiding the roundtrip. It just changes the focus of the webserver to a different page and transfers the page processing to a different page.

Response.Redirect can be used for both .aspx and html pages whereas Server.Transfer can be used only for .aspx pages.

Response.Redirect can be used to redirect a user to an external websites. Server.Transfer can be used only on sites running on the same server. You cannot use Server.Transfer to redirect the user to a page running on a different server.

Response.Redirect changes the url in the browser. So they can be bookmarked. Whereas Server.Transfer retains the original url in the browser. It just replaces the contents of the previous page with the new one.

**What method do you use to explicitly kill a users session?**

Session.Abandon().

**What is a webservice?**

Web Services are applications delivered as a service on the Web. Web services allow for programmatic access of business logic over the Web. SOAP over HTTP is the most commonly used protocol for invoking Web services.

**Explain client side state management system.**

ASP.NET provides several techniques for storing state information on the client. These include the following:

View state: ASP.NET uses view state to track values in controls between page requests. It works within the page only. You cannot use view state value in next page.

Control state: You can persist information about a control that is not part of the view state. If view state is disabled for a control or the page, the control state will still work.

Hidden fields: It stores data without displaying that control and data to the user’s browser. This data is presented back to the server and is available when the form is processed. Hidden fields data is available within the page only (page-scoped data).

Cookies: Cookies are small piece of information that server creates on the browser. Cookies store a value in the user’s browser that the browser sends with every page request to the web server.

Query strings: In query strings, values are stored at the end of the URL. These values are visible to the user through his or her browser’s address bar. Query strings are not secure. You should not send secret information through the query string.

Session State: Session object stores user-specific data between individual requests. This object is same as application object but it stores the data about particular user.

**Explain the ASP.NET page lifecycle.**

Lifecycle of a page in ASP.NET follows following steps:

Page\_Init(Initialization of the page) >> LoadViewState(loading of View State) >> LoadPostData(Postback data processing) >> Page\_Load(Loading of page) >> RaisePostDataChangedEvent(PostBack change notification) >> RaisePostBackEvent (PostBack event handling) >> Page\_PreRender (Page Pre Rendering Phase) >> SaveViewState (View state saving) >> Page\_Render (Page rendering) >> Page\_UnLoad (Page unloading)

**What is Fragment Caching in ASP.NET?**

Fragment caching allows to cache specific portions of the page rather than the whole page. It is done by implementing the page in different parts by creating everything in form of user controls and caching each user control individually.

**What is partial classes in .net?**

When there is a need to keep the business logic separate from the User Interface or when there is some class which is big enough to have multiple number of developers implement the methods in it, the class can be separated and written in different files as partial class.

**What is the difference between URL and URI?**

URI - Uniform Resource Identifier: it’s a string and its responsibility is to identify a resource by meta-information. It gives information about only one resource.

URL - Uniform Resource Locator: identifies the resource on the net and tells it is obtainable using what protocols.

**Explain Singleton architecture of Remoting.**

- This architecture comprises of messages, sinks, proxies and configuration objects.

There are two approaches that are used in Remoting architecture:

1. Single call

2. Singleton

- The singleton remoting architecture is effective for limiting the maximum number of instances of a class to exactly one.

- This architecture is used when all the applications have to use or share same data.