


(2½ hours)

[Total Marks: 75]

Solution Set

1.	Attempt <i>any two</i> of the following:	10
a.	<p>Explain .NET framework 4.0 with diagram in detail. Ans:</p>  <p>The diagram illustrates the .NET Framework architecture. It is divided into two main parts: the .NET Class Library and the Common Language Runtime (CLR).</p> <p>The .NET Class Library: This layer contains various components used for application development, including:</p> <ul style="list-style-type: none"> ADO.NET Data Access Web Forms Windows Forms XML File I/O (And So On) Core System Classes (Threading, Serialization, Reflection, Collections, And So On) <p>The Common Language Runtime: This layer provides the runtime environment for managed code, including:</p> <ul style="list-style-type: none"> Compiler and Loader Code Verification and Optimization Memory Management and Garbage Collection Code Access Security (Other Managed Code Services) 	
b.	<p>What is CLR? Explain the role of CLR in .NET framework.</p> <p>Ans: CLR (Common Language Runtime) is a heart of Dot Net Framework. It is a core runtime environment of .NET Framework for executing applications. The main function of Common Language Runtime (CLR) is to convert the Managed Code into native code and then execute the Program. It acts as a layer between Operating Systems and the applications written in .Net languages. CLR handles the execution of code and provides useful services for the implementation of the program. In addition to executing code, CLR provides services such as memory management, thread</p>	

	management, security management, code verification, compilation, and other system services.	
c.	<p>Explain the features of static member and static function.</p> <p>Ans: Static Members</p> <p>Types of class members:</p> <ul style="list-style-type: none"> • Non-static members: This is the default type for all the members. If you do not use the "static" keyword for the declaration of a field / property or a method, then it can be called a "Non-static member". The main feature of a non-static member is it will be bound with the object only. <p>Non-static Fields / Properties: The memory is allocated when the object is created.</p> <p>Non-static Methods: These methods can implement operations on non-static fields and properties.</p> <ul style="list-style-type: none"> • Static Members: If you use the "static" keyword for the declaration of a field / property or a method, then it is called a "Static member". The main feature of a non-static member is that it will not be bound with any object. It is individually accessible with the class name. In other words, the static members are accessible directly, without even creating one object also. <p>Static Fields / Properties: The memory will be allocated individually, without any relation with the object.</p> <p>Static Methods: These methods can implement operations on static fields and properties only; and can't access the non-static members.</p>	
d.	<p>Explain with suitable example about sealed class and method.</p> <p>Ans: Sealed Class</p> <ul style="list-style-type: none"> • • Sealed class is used to define the inheritance level of a class. • • The sealed modifier is used to prevent derivation from a class. An error occurs if a sealed class is specified as the base class of another class. <ol style="list-style-type: none"> 1. A class, which restricts inheritance for security reason is declared, sealed class. 2. Sealed class is the last class in the hierarchy. 3. Sealed class can be a derived class but can't be a base class. 4. A sealed class cannot also be an abstract class. Because abstract class has to provide functionality and here we are restricting it to inherit. <p>Example:</p> <pre>using System;</pre>	

	<pre>using System.Collections.Generic; using System.Linq; using System.Text; namespace sealed_class { class Program { public sealed class BaseClass { public void Display() { Console.WriteLine("This is a sealed class which can;t be further inherited"); } } } } </pre> <p>Sealed Methods</p> <p>Sealed method is used to define the overriding level of a virtual method.</p> <p>Sealed keyword is always used with override keyword.</p>	
2.	Attempt <i>any two</i> of the following:	10
a.	<p>Explain the concept of private and shared assembly.</p> <p>Ans: A private assembly is normally used by a single application, and is stored in the application's directory, or a sub-directory beneath. and A shared assembly is normally stored in the global assembly cache, which is a repository of assemblies maintained by the .NET runtime. Shared assemblies are usually libraries of code, which many applications will find useful.</p>	
b.	<p>What is namespace? Explain System namespace.</p> <p>Ans: A namespace is designed for providing a way to keep one set of names separate from another. The class names declared in one namespace does not conflict with the same class names declared in another.</p> <p>Syntax:</p> <pre>namespace namespace_name { // code declarations }</pre> <p>The System namespace contains fundamental classes and base classes that define commonly used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions.</p>	
c.	<p>Explain ASP.NET page life cycle.</p> <p>Ans: The page life cycle phases are:</p> <ul style="list-style-type: none"> • Initialization • Instantiation of the controls on the page • Restoration and maintenance of the state • Execution of the event handler codes • Page rendering • Page request - When ASP.NET gets a page request, it decides whether to parse and compile the page, or there would be a cached version of the page; accordingly the response is sent. 	

	<ul style="list-style-type: none"> • Starting of page life cycle - At this stage, the Request and Response objects are set. If the request is an old request or post back, the IsPostBack property of the page is set to true. The UICulture property of the page is also set. • Page initialization - At this stage, the controls on the page are assigned unique ID by setting the UniqueID property and the themes are applied. For a new request, postback data is loaded and the control properties are restored to the view-state values. • Page load - At this stage, control properties are set using the view state and control state values. • Validation - Validate method of the validation control is called and on its successful execution, the IsValid property of the page is set to true. • PostBack event handling - If the request is a postback (old request), the related event handler is invoked. • Page rendering - At this stage, view state for the page and all controls are saved. The page calls the Render method for each control and the output of rendering is written to the OutputStream class of the Response property of page. • Unload - The rendered page is sent to the client and page properties, such as Response and Request, are unloaded and all cleanup done. 	
d.	<p>What is Garabage collector? Explain how it is implemented.</p> <p>Ans: arbage collection is the process of recycling the dynamically allocated memory. It is performed by a garbage collector whose job is to recycle the memory which will never be used again.</p> <p>using System;</p> <pre> class MyClass : IDisposable { public MyClass() //default ctor { this.iNumber = 0; System.Console.WriteLine("ctor:MyClass {0}", iNumber); } public MyClass(Int32 iNumber) // specialized ctor { this.iNumber = iNumber; System.Console.WriteLine("ctor:MyClass {0}", iNumber); } ~MyClass() // dtor or finalize { System.Console.WriteLine("dtor:~MyClass {0}", iNumber); } public void Dispose() // helper finalize function { // here you can free the resources you allocated explicitly System.GC.SuppressFinalize(this); } private int iNumber; } class main { static void Main() </pre>	

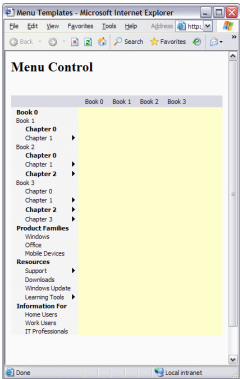
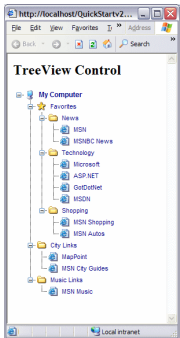
	<pre> { MyClass myClass1 = new MyClass(); MyClass myClass2 = new MyClass(19); myClass1.Dispose(); // myClass1 is explicitly exposed. System.GC.Collect(); System.GC.WaitForPendingFinalizers(); // myClass2 is implicitly exposed by GC. Console.ReadLine(); } } </pre>	
3.	Attempt <u>any two</u> of the following:	10
a.	<p>Explain the usage of button control with suitable example.</p> <p>Ans: The Button control is used to display a push button. The push button may be a submit button or a command button. By default, this control is a submit button.</p> <p>A submit button does not have a command name and it posts the page back to the server when it is clicked. It is possible to write an event handler to control the actions performed when the submit button is clicked.</p> <p>A command button has a command name and allows you to create multiple Button controls on a page. It is possible to write an event handler to control the actions performed when the command button is clicked.</p> <pre> <script runat="server"> Sub submit(sender As Object, e As EventArgs) lbl1.Text="Your name is " & txt1.Text End Sub </script> <!DOCTYPE html> <html> <body> <form runat="server"> Enter your name: <asp:TextBox id="txt1" runat="server" /> <asp:Button OnClick="submit" Text="Submit" runat="server" /> <p><asp:Label id="lbl1" runat="server" /></p> </form> </pre>	
b.	<p>Explain the use of web.config and global.asx files in ASP.NET application.</p> <p>Ans: ASP.NET uses the global.asax to establish any global objects that your Web application uses. The .asax extension denotes an application file rather than .aspx for a page file. Each ASP.NET application can contain at most one global.asax file. The file is compiled on the first page hit to your Web application. ASP.NET is also configured so that any attempts to browse to the global.asax page directly are rejected. However, you can specify application-wide settings in the web.config file. The web.config is an XML-formatted text file that resides in the Web site's root directory. Through Web.config you can specify settings like custom 404 error pages, authentication and authorization settings for the Web site, compilation options for the ASP.NET Web pages, if tracing should be enabled, etc.</p>	
c.	What is the difference between Label and Link Label?	

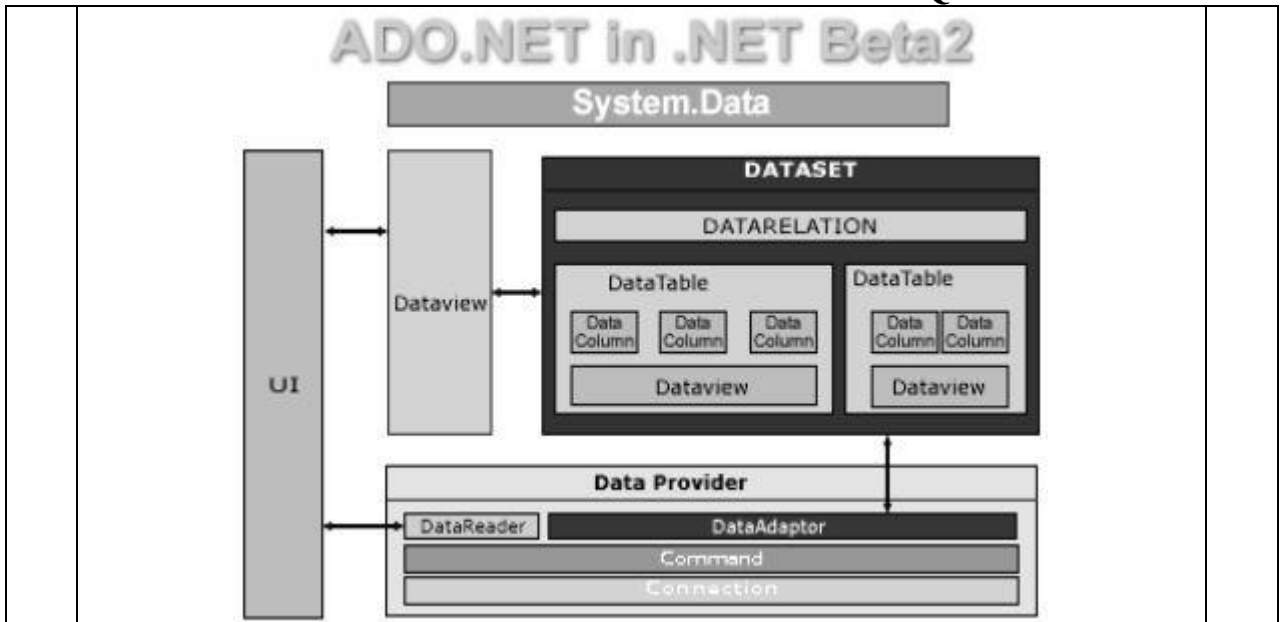
	<p>Ans: A link label is like a hyperlink that you'd see on a webpage. It's a label that's blue and has an underline and, when clicked, can point to a URL. A label is just a label.</p> <p>0 down vote accepted down vote accepted</p> <p>A LinkLabel control is a label control that can display a <i>hyperlink</i>. A LinkLabel control is inherited from the <i>Label</i> class so it has all the functionality provided by the Windows Forms Label control.</p> <p>LinkLabel control does not participate in <i>user input or capture mouse or keyboard events</i>.</p>	
d.	<p>Explain any two ASP.NET server control.</p> <p>Ans:</p> <ul style="list-style-type: none"> • HTML controls • HTML Server controls • ASP.NET Server controls • ASP.NET Ajax Server controls • User controls and custom controls <p>ASP.NET HTML Server Controls</p> <p>ASP.NET provides a way to work with HTML Server controls on the server side; programming with a set of controls collectively is called HTML Controls.</p> <ul style="list-style-type: none"> • These controls are grouped together in the Visual Studio Toolbox in the the HTML Control tab. The markup of the controls are similar to the HTML control. • These controls are basically the original HTML controls but enhanced to enable server side processing. • HTML elements in ASP. NET files are, by default, treated as text. To make these elements programmable, add a <code>runat="server"</code> attribute to the HTML element. This attribute indicates that the element should be treated as a server control. 	

Control Name	HTML tag
HtmlHead	<head>element
HtmlInputButton	<input type=button submit reset>
HtmlInputCheckbox	<input type=checkbox>
HtmlInputFile	<input type = file>
HtmlInputHidden	<input type = hidden>
HtmlInputImage	<input type = image>
HtmlInputPassword	<input type = password>
HtmlInputRadioButton	<input type = radio>
HtmlInputReset	<input type = reset>
HtmlText	<input type = text password>
HtmlImage	 element
HtmlLink	<link> element
HtmlAnchor	<a> element
HtmlButton	<button> element
HtmlButton	<button> element
HtmlForm	<form> element
HtmlTable	<table> element
HtmlTableCell	<td> and <th>
HtmlTableRow	<tr> element
HtmlTitle	<title> element
HtmlSelect	<select> element

4. Attempt <u>any two</u> of the following:	10
<p>a. Explain Validation controls available in ASP.NET.</p> <p>Ans: Validation is important part of any web application. User's input must always be validated before sending across different layers of the application. Validation controls are used to:</p> <ul style="list-style-type: none"> • Implement presentation logic. • To validate user input data. • Data format, data type and data range is used for validation. <p>Validation is of two types:</p> <ol style="list-style-type: none"> 1. Client Side 2. Server Side <p>Client side validation is good but we have to be dependent on browser and scripting language support.</p> <p>Client side validation is considered convenient for users as they get instant feedback. The main advantage is that it prevents a page from being postback to the server until the client validation is executed successfully.</p> <p>There are six types of validation controls in ASP.NET</p> <ol style="list-style-type: none"> 1. RequiredFieldValidation Control 2. CompareValidator Control 3. RangeValidator Control 	

	<p>4. RegularExpressionValidator Control 5. CustomValidator Control 6. ValidationSummary</p> <table border="1" data-bbox="280 327 1382 707"> <thead> <tr> <th>Validation Control</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>RequiredFieldValidation</td> <td>Makes an input control a required field</td> </tr> <tr> <td>CompareValidator</td> <td>Compares the value of one input control to the value of another input control or to a fixed value</td> </tr> <tr> <td>RangeValidator</td> <td>Checks that the user enters a value that falls between two values</td> </tr> <tr> <td>RegularExpressionValidator</td> <td>Ensures that the value of an input control matches a specified pattern</td> </tr> <tr> <td>CustomValidator</td> <td>Allows you to write a method to handle the validation of the value entered</td> </tr> <tr> <td>ValidationSummary</td> <td>Displays a report of all validation errors occurred in a Web page</td> </tr> </tbody> </table>	Validation Control	Description	RequiredFieldValidation	Makes an input control a required field	CompareValidator	Compares the value of one input control to the value of another input control or to a fixed value	RangeValidator	Checks that the user enters a value that falls between two values	RegularExpressionValidator	Ensures that the value of an input control matches a specified pattern	CustomValidator	Allows you to write a method to handle the validation of the value entered	ValidationSummary	Displays a report of all validation errors occurred in a Web page	
Validation Control	Description															
RequiredFieldValidation	Makes an input control a required field															
CompareValidator	Compares the value of one input control to the value of another input control or to a fixed value															
RangeValidator	Checks that the user enters a value that falls between two values															
RegularExpressionValidator	Ensures that the value of an input control matches a specified pattern															
CustomValidator	Allows you to write a method to handle the validation of the value entered															
ValidationSummary	Displays a report of all validation errors occurred in a Web page															
b.	<p>Write a note on state management in ASP .NET. Ans: State Management Types</p> <p>In ASP.NET there are the following 2 State Management methodologies:</p> <div data-bbox="647 909 1026 1111" style="text-align: center;"> <p>The diagram consists of two overlapping circles. The left circle is red and labeled 'Client Side State Management'. The right circle is green and labeled 'Server Side State Management'. The overlapping area in the center is a darker shade of red/green.</p> </div> <p>Client-Side State Management</p> <p>Whenever we use Client-Side State Management, the state related information will directly get stored on the client-side. That specific information will travel back and communicate with every request generated by the user then afterwards provides responses after server-side communication.</p> <p>Server-Side State Management</p> <p>Server-Side State Management is different from Client-Side State Management but the operations and working is somewhat the same in functionality. In Server-Side State Management all the information is stored in the user memory. Due to this functionality there is more secure domains at the server side in comparison to Client-Side State Management.</p>															
c.	<p>Explain the work of session state in ASP.NET. Ans: Session state essentially means all the settings that you have made for your web application for maintaining the session. Session State itself is a big thing. It says all about your session configuration, either in the <i>web.config</i> or from the code-behind. In the <i>web.config</i>, <SessionState> elements are used for setting the configuration of the session. Some of them are Mode, Timeout, StateConnectionString, CustomProvider, etc. I have discussed about each and every section of the connection string. Before I discuss Session Mode, take a brief overview of session events.</p>															

	<p>Session Events</p> <p>There are 2 type of events available in ASP.NET. We can handle both sessions in a global.asax file.</p> <ol style="list-style-type: none"> 1. Session_Start(): When the new session is initialized then the session_start event is raised. <p>Session_end(): When the session is Expires then the Session_End event raised.</p>	
<p>d.</p>	<p>Explain any two Site Navigation Controls in ASP.NET.</p> <p>Ans: Menu Control—This provides a traditional navigational interface, typically down the side, or across the top, of a web site. It can render an arbitrary number of nested submenus, and optional "pops-out" submenus when a user hovers over an item.</p>  <p>TreeView Control—This provides a vertical tree-like user interface that can be expanded and collapsed by selecting the individual nodes. It also provides check box functionality that allows items to be selected.</p> 	
<p>5.</p>	<p>Attempt <i>any two</i> of the following:</p>	<p>10</p>
<p>a.</p>	<p>Explain ADO .NET object model with help of suitable diagram.</p> <p>Ans:</p>	



The ADO.NET object model is rich, but at its heart it is a fairly straightforward set of classes. The most important of these is the DataSet. The DataSet represents a subset of the entire database, cached on your machine without a continuous connection to the database.

Periodically, you'll reconnect the DataSet to its parent database, update the database with changes you've made to the DataSet, and update the DataSet with changes in the database made by other processes.

This is highly efficient, but to be effective the DataSet must be a robust subset of the database, capturing not just a few rows from a single table, but also a set of tables with all the metadata necessary to represent the relationships and constraints of the original database. This is, not surprisingly, what ADO.NET provides.

The DataSet is composed of DataTable objects as well as DataRelation objects. These are accessed as properties of the DataSet object. The Tables property returns a DataTableCollection, which in turn contains all the DataTable objects.

b. Explain in brief the Data Bound Controls.

Ans: **DataList** **Control**

The DataList control was introduced with ASP.NET 1.0. DataList is the next step up from a Repeater; except you have very little control over the HTML that the control renders. DataList allows you to repeat columns horizontally or vertically. The DataList control renders data as a table and enables you to display data records in various layouts, such as ordering them in columns or rows. You can configure the DataList control to enable users to edit or delete a record in the table. We can use a DataList control where we need a single-column list. The DataList control works like the Repeater control, used to display the data in a repeating structure, such as a table. It displays data in a format that you can define using a template and styles. However, it arranges the data defined in the template within various HTML structures. This includes options for horizontal or vertical layout and it also allows you to set how the

	<p>data should be repeated, as flow or table layout. The DataList control does not automatically use a data source control to edit data.</p> <p>GridView Control</p> <p>ASP.NET provides a number of tools for showing tabular data in a grid, including the GridView control. It was introduced with ASP.NET 2.0. The GridView control is used to display the values of a data source in a table. Each column represents a field where each row represents a record. It can also display empty data. The GridView control provides many built-in capabilities that allow the user to sort, update, delete, select and page through items in the control. The GridView control can be bound to a data source control, in order to bind a data source control, set the DataSourceID property of the GridView control to the ID value of the data source control. It's considered a replacement for the DataGrid control from .NET 1.1. Therefore, it is also known as a super DataGrid. The GridView control offers improvements such as the ability to define multiple primary key fields, improved user interface customization using bound fields and templates and a new model for handling or canceling events. Performance is slow compared to DataGrid and ListView.</p>	
c.	<p>Explain in brief about an authentication and authorization.</p> <p>Ans: Authentication</p> <p>Authentication is the process of verifying the identity of a user by obtaining some sort of credentials and using those credentials to verify the user's identity. If the credentials are valid, the authorization process starts. Authentication process always proceeds to Authorization process.</p> <p>ASP.Net Authentication</p> <p>The ASP.NET authentication scheme that is used to identify users who view an ASP.NET application. An ASP.net application has two separate authentication levels because all requests coming through IIS before it handled by ASP.NET. After IIS authentication schemes ASP.NET implements additional authentication schemes. They are :</p> <p>Windows Authentication Forms Authentication</p> <p>Authorization</p> <p>Authorization is the process of allowing an authenticated users to access the resources by checking whether the user has access rights to the system. Authorization helps you to control access rights by granting or denying specific permissions to an authenticated user.</p> <p>ASP.Net Authorization</p> <p>ASP.NET allows two ways to authorize access to a given resources, they are URL authorization and File authorization</p>	

	<p>URL authorization</p> <p>URL authorization maps users and roles to URLs in ASP.NET applications.</p>	
d.	<p>Differentiate between DataSet and DataReader.</p> <p>Ans: DataSet: DataSet is a disconnected orient architecture that means there is no need of active connections during work with datasets and it is a collection of DataTables and relations between tables. It is used to hold multiple tables with data. You can select data form tables, create views based on table and ask child rows over relations. Also DataSet provides you with rich features like saving data as XML and loading XML data.</p> <p>DataReader: DataReader is used to read the data from database and it is a read and forward only connection oriented architecture during fetch the data from database. DataReader will fetch the data very fast when compared with dataset. Generally we will use ExecuteReader object to bind data to datareader.</p>	
6.	Attempt <i>any two</i> of the following:	10
a.	<p>What is LINQ? Give its advantages.</p> <p>Ans: It is a technique for querying data that is integrated into .NET languages such as C#. It has a single unitive syntax for querying multiple data sources such as relational data and XML data.</p> <p>Benefits of LINQ</p> <ul style="list-style-type: none"> • Because LINQ is integrated into the C# language, it provides syntax highlighting and IntelliSense. These features make it easy to write accurate queries and to discover mistakes at design time. • Because LINQ queries are integrated into the C# language, it is possible for you to write code much faster than if you were writing old style queries. In some cases, developers have seen their development time cut in half. • The integration of queries into the C# language also makes it easy for you to step through your queries with the integrated debugger. • The hierarchical feature of LINQ allows you to easily see the relationship between tables, thereby making it easy to quickly compose queries that join multiple tables. • The unitive foundation of LINQ allows you to use a single LINQ syntax when querying multiple data sources. This allows you to get up to speed on new technologies much more quickly. If you know how to use LINQ to Objects, it is not hard to learn how to use LINQ to SQL, and it is relatively easy to master LINQ to XML. • Because LINQ is extensible, you can use your knowledge of LINQ to make new types of data sources querable. • After creating or discovering a new LINQ provider, you can leverage your knowledge of LINQ to quickly understand how to write queries against these new data sources. 	
b.	Explain the working of AJAX.	

	<p>Ans: AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.</p> <p>Steps of AJAX Operation</p> <ul style="list-style-type: none"> • A client event occurs. • An XMLHttpRequest object is created. • The XMLHttpRequest object is configured. • The XMLHttpRequest object makes an asynchronous request to the Webserver. • The Webserver returns the result containing XML document. • The XMLHttpRequest object calls the callback() function and processes the result. • The HTML DOM is updated. 	
c.	<p>Explain the syntax of jQuery with the example.</p> <p>Ans: jQuery Syntax:</p> <p>The jQuery syntax is tailor-made for selecting HTML elements and performing some action on the element(s).</p> <p>Basic syntax is: \$(selector).action()</p> <ul style="list-style-type: none"> • A \$ sign to define/access jQuery • A (selector) to "query (or find)" HTML elements • A jQuery action() to be performed on the element(s) <p>Examples:</p> <p>\$(this).hide() - hides the current element.</p> <p>\$("p").hide() - hides all <p> elements.</p>	
d.	<p>List and explain jQuery selectors.</p> <p>Ans: jQuery selectors allow you to select and manipulate HTML element(s).</p> <p>jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing <u>CSS Selectors</u>, and in addition, it has some own custom selectors.</p> <p>All selectors in jQuery start with the dollar sign and parentheses: \$().</p> <p>The element Selector</p> <p>The jQuery element selector selects elements based on the element name.</p> <p>You can select all <p> elements on a page like this:</p>	

	<p><code>\$("#p")</code> The #id Selector</p> <p>The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.</p> <p>An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.</p> <p>To find an element with a specific id, write a hash character, followed by the id of the HTML element:</p> <p><code>\$("#test")</code> The .class Selector</p> <p>The jQuery class selector finds elements with a specific class.</p> <p>To find elements with a specific class, write a period character, followed by the name of the class:</p> <p><code>\$(".test")</code></p>	
7.	Attempt <u>any three</u> of the following:	15
a.	<p>Explain the concept of boxing and unboxing. Ans: Boxing: Boxing is a process in which object instances are created and copy values in to that instance. It permits any value type to be implicitly converted to type object or to any interface type implemented by value type. Unboxing: Unboxing is vice versa of boxing operation where the value is copied from the instance in to appropriate storage location. Here is sample code of boxing and unboxing where integer data type is converted in to object and then vice versa.</p> <pre>int x = 0; object y = null; x = 10; // boxing process y = x; // unboxing process x = y;</pre>	
b.	<p>Explain the concept of Events and Delegates with suitable example. Ans: Delegate types</p> <p>A Delegate is an abstraction of one or more function pointers (as existed in C++; the explanation about this is out of the scope of this article). The .NET has implemented the concept of function pointers in the form of delegates. With delegates, you can treat a function as data. Delegates allow functions to be passed as parameters,</p>	

returned from a function as a value and stored in an array. Delegates have the following characteristics:

- Delegates are derived from the System.MulticastDelegate class.
- They have a signature and a return type. A function that is added to delegates must be compatible with this signature.
- Delegates can point to either static or instance methods.
- Once a delegate object has been created, it may dynamically invoke the methods it points to at runtime.
- Delegates can call methods synchronously and asynchronously.

```
namespace DelegateArticle
{
    public delegate string FirstDelegate (int x);

    public class Sample
    {
        public delegate void SecondDelegate (char a, char b);
    }
}
```

Events aren't delegate instances.

Events are pairs of methods, appropriately decorated in IL to tie them together and let languages know that the methods represent events. The methods correspond to *add* and *remove* operations, each of which take a delegate instance parameter of the same type (the type of the event). The applications and windows communicate via predefined messages. These messages contain various pieces of information to determine both window and application actions. The .NET considers these messages as an event. If you need to react to a specific incoming message then you would handle the corresponding event. For instance, when a button is clicked on a form, Windows sends a WM_MOUSECLICK message to the button message handler.

```
using System;

class Test
{
    public event EventHandler MyEvent
    {
        add
        {
            Console.WriteLine ("add operation");
        }

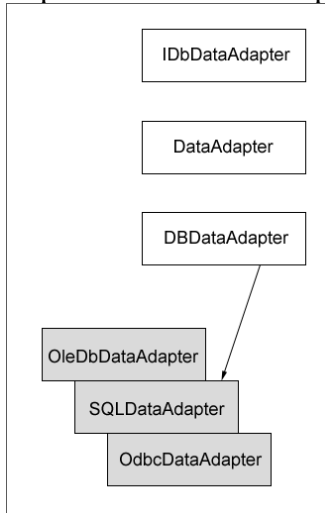
        remove
        {
            Console.WriteLine ("remove operation");
        }
    }

    static void Main()
```

	<pre> { Test t = new Test(); t.MyEvent += new EventHandler (t.DoNothing); t.MyEvent -= null; } void DoNothing (object sender, EventArgs e) { } } </pre>	
c.	<p>What is CSS? Explain internal and external CSS. Ans: CSS is used to design webpage and its content with css we make beautiful and attractive sites that can help easily interact with user.</p> <p>Internal CSS Stylesheet</p> <p>When creating a stylesheet internally in the web page, you will need to use the HTML tags in the Head section of your webpage. All the code for the Internal CSS stylesheet is contained between the section of your websites code. Below is an example of what an Internal stylesheet looks like.</p> <pre> <head> <style type="text/css"> h1 {color:blue;} h2 {color:red;} p {color:green;} </style> </head> </pre> <p>External CSS Stylesheet</p> <p>When using an external stylesheet you must reference the stylesheet in the HTML page that is using it. You would add the code below to your HTML document to reference a stylesheet in the same location as the HTML page called "<i>style.css</i>". You can upload the "<i>style.css</i>" page can be located anywhere in your files. You can name your stylesheet whatever you like and link to as many as you like. You can simply link to it in your head section and every edit your make to the "<i>style.css</i>" sheet will be globally changed through out the site. Below is what the code looks like.</p> <pre> <head>
 <link rel="stylesheet" type="text/css" href="/support/style.css" />
 </head> </pre>	
d.	<p>What is master page? Explain the advantages of using master page. Ans: A master page is an ASP.Net file with extension .master, such as MasterPage.Master, with a predefined layout that can include static text, HTML elements, and server control.</p> <p>Advantages: They allow you to centralize the common functionality of your passage so that you can make updates. They provide an object model that allows to customize the master page from individual content pages.</p>	
e.	Explain DataAdapter object in ADO.NET.	

Ans: The DataAdapter class for all data providers comes from the DbDataAdapter class, which in turn comes from the DataAdapter class.

An application doesn't create an instance of the DbDataAdapter interface directly, but instead it creates an instance of a class that inherits IDbDataAdapter and DbDataAdapter. As you can see from Figure, many data provider "specific" classes implement IDbDataAdapter.



The DataAdapter enables you to connect to a dataset and specify SQL strings for retrieving data from or writing data to a DataSet. As you've seen in the beginning of this article, a dataset represents in-memory cached data. An in-memory object frees you from the confines of the specifics of database and allows you to deal with the data in memory. The DataAdapter serves as an intermediary between the database and the DataSet.

f. Explain following Ajax controls with example and properties.
i. ScriptManager Control ii. Timer Control

Ans:

ScriptManager Control

ScriptManager control is the parent control that needs to be there on every page wherever we are trying to use ASP.NET AJAX controls. ScriptManager control manages client script for AJAX enabled ASP.NET pages. This control enables client script to use the type system extensions and support features for partial page rendering, webservice calls etc.

There are bunch of properties and method associated with ScriptManager :

AllowCustomErrorsRedirect	Gets or sets a value that determines whether custom errors section of the web.config file is used during error.
AsyncPostBackErrorMessage	Gets or sets the error message that is sent to the client when an unhandled server exception occurs during an asynchronous postback.
AsyncPostBackTimeout	Gets or sets the time in seconds before asynchronous postbacks time out if no response is received.

ClientID	Gets the server control identifier generated by ASP.NET (The id for this control that is rendered on the page)
EnablePageMethods	Gets or sets whether public static page methods in asp.net page can be called from client script.
EnableViewState	Gets or sets a value that indicates whether server control persists its viewstate and the viewstate of its child control if any.
IsInAsyncPostBack	Gets or sets a value that indicates whether the current postback is being executed in partial rendering mode.
<p>Timer Control:</p> <p>Timer control performs postbacks at the defined intervals. This is generally used to update the content of UpdatePanel at a defined interval. Apart from this it can be used to post the whole page at defined interval.</p> <p>here are few properties that are used more frequently, these are</p>	
Interval	Gets or sets the time interval in milliseconds after OnTick event of Timer control will fire.
OnTick	Used to specify the method name that will after specified time interval.