Q.1 Attempt the following (any TWO) **[10]**

Q.1(a) What is boxing and unboxing? How it is achieved? **[5]**

(A) All the C# types are derived from 'object' type. Thus, a reference of type object can be used to refer to any other type, including value types. When an object reference refers to a value type, a process known as boring occurs.

Boxing : Causes the value of a value type to be stored in an object instance.

Unboxing : is the process of retrieving a value from a boxed object. Unboxing is performed using an explicit cost from the object reference to its corresponding value type.

for example :

```
using system :
class program
{
    static void main(string[]g)
    {
        int n = 10;
        object ob = n;        // Boxing

        int m = (int)ob; //Unboxing
        Console.WriteLine("N=" +n);
        Console.WriteLine("ob=" + ob);
    }
}
```

Q.1(b) What is type conversion? Explain implicit and explicit type conversion with suitable example. **[5]**

(A) **Type conversion :** It is converting one type of data to another type. It is also known as type casting.

Type Conversion

Type conversion takes two forms

Implicit Conversion

Explicit conversion

Implicit conversion

- This conversion is performed without any loss of data.
- For Example : short type can be implicitly converted to int because the short range is a subset of int range.

```
short b = 75;
int a = b;
```

- This conversion occurs automatically.
- An implicit conversion is also known as an Automatic type conversion.
- The process of assigning a smaller type to a larger one is known as implicit Conversion or widening or promotion.

Example :

```
byte a1 = 75;
short a2 = a1;
```

```
int a3 = a2;
long a4 = a3;
float a5 = a4;
decimal a6 = a4;
```

Explicit Type conversion

- There are many conversions that cannot be implicitly made between types.
- If one attempts such conversions the compiler gives an error message.
- The process of assigning a larger type to a smaller one is known as explicit Conversion or Narrowing.
- The following conversions cannot be made implicitly.
  int to short
  int to uint
  uint to int
  float to int
  decimal to any numeric type
  any numeric type to char
- However, such conversions can be carried out explicitly using casting, converting and parsing.

Casting

```
Type variable1 = (type) variable2;
int m = 50;
byte n = (byte) m;
```

casting into a smaller type results in the loss of data.

Converting

- C# provides the methods to convert between numeric values and strings

```
int m = 100;
string s = m.ToString();
string s  = Console.readLine();
int s1 = Convert.ToInt32(s);
```

Parsing

- It returns the numeric representation of string data

```
String s = 100;
int  s1 = int.parse(s);
int m = n + 500;
```

Q.1(c) What is user defined exception? Explain with Syntax. [5]

(A) User defined exception : It is possible to manually throw an exception by using the throw statement. The general form of throw statement is
throw new exceptionObject;
where exceptionObject must be an object of an exception class derived from exception.

for example :

```
using system;
class DivisionByGreaterNumber : Exception
{
    public DivisionByGreaterNumber (String msg); base(msg)
    {    }
}

class throwcheck
```

```
{
    public static void Main()
    {
        try{int n = 10/11;
        if (n==0) throw new DivisionByGreterNumber
        ("Division results in zero");
        } Catch(DivisionByGreaterNumber d)
        {
            Consloe.Write(d);
        }
    }
}
```

Q.1(d) Explain try, catch and finally blocks in exception handling with suitable example. [5]

(A) C# uses a keyword try to preface a block of code that is likely to cause an error condition and throw an exception. A catch block defined by the keyword catch catches the exception thrown by the try block and handles it approximately. The catch block is added immediately after the try block. The try block can have one or more statements that could generate an exception. If anyone statement generates an exception the remaining statements in the block or skipped and execution jumps to the catch block that is placed to the next to the try block.

The catch block can have one or more statements that are necessary to process the exception. Every try statement should be followed by atleast one catch or one finally statement. The catch statement is passed a single parameter which is the reference to the exception object. If the catch parameter matches with the type of exception object, then the exception is caught and statements in the catch block will be executed else the exception is not caught and the default exception handler will cause the execution to terminate.

In C# finally statement can be used to handle an exception that is not caught by any of the previous catch statements. A finally block can be used to handle any exception generated within a try block. It can be added immediately after try block or after the last catch block when a finally block is defined the program is guaranteed to execute regardless of how control leaves the try. The general form for try-catch finally block is

```
try
{
    ;
}
catch (…)
{
    : :
}
finally
{   ;
}
using system
class TryCatchFinally
{
    public static void Main()
    {
        int a = 10, b=0, c;
        try
        {
            c = a/b;
    }
```

```
        catch(Exception e)
        {
            console.WriteLine(e.Message);
        }
        finally
        {
            console.WriteLine("inside Finally");
        }
    }
    }
```

Q.2 Attempt the following (any TWO) [10]

Q.2(a) What is an assembly? Explain different components of assembly. [5]

(A) An assembly is a collection of types and references that form a logical unit of functionality. All types in the NET framework most exist in assemblies, the common language runtime does not support types outside the assemblies. Each time an application is created an assembly is build. Each assembly is stored as an .exe or .dll. The goal of assembly is to eliminate .DLL. The components of assembly are

- Assembly Manifest : A manifest is a section of assembly that contains metadata about the assembly. It contains information like version number, publisher, encryption information etc. The assembly manifest can be stored in either a PE file or in a standalone PE file.
- MSIL source code : When you compile an assembly the compiler translates the source code into MSIL instead of machine specific object code. As MSIL is platform independent, here, MSIL source code can be executed on any platform with the support of .NET framework. It also stores the instructions related to the operators like arithmetic and logic operations etc.
- Type metadata : It describes the format of the contained types and external types referenced by that assembly. After executing the code, the CLR loads this metadata into memory and drives the information about classes, their member etc.
- Resources : The .NET framework consists of a main assembly that contain various resources like application icons, sound clips, images etc

Q.2(b) What is CSS selector? Explain different selectors present in CSS. [5]

(A) In CSS Selectors are patterns area to select the elements you want to style. The declaration blocks contains one or more declarations separated by semicolon. Each declaration include a CSS properly name and a value separated by a colon. A CSS declaration always ends with a semicolon and declaration blocks are surrounded by curly braces. For example P {color; red; text-align; center;}. The different types of selectors are.

- Element selector : The element selector selects elements based on the element name. Example P{text-align: center; color: red;}
- Id Selector : They are individually assigned for the purpose of defining an a per-element basis. An Id sector is assigned by using # indictor. Example : #para1{text- align; center, color; red;}
- Class Selector : It selects elements with a specific class attribute. To select elements with a specific class write a period (.) character followed by the name of the class. Example.
  center {text-align; center; color: red;}
- Grouping selectors: In order to decrease repetitious statement with stylesheets, grouping of selectors and declarations is allowed. Example H1, H2, H3, H4, H5, H6 {Color; red, text. align; center;}

Q.2(c) What is event? What are the differences between delegate and event? [5]

(A) Events enable a class or object to notify other classes or objects when something of in latest access. The class that sends (or raises) the event is called the publisher and the classes that receive (or handle) the event are called subscribers.

	Event	Delegate
(i)	It is a data member of a type (class / structure)	It is a datatype (reference type) that holds references of methods with some signatures also called as function pointer.
(ii)	It is declared inside a type (class/structure)	It may or may not be declared inside a class.
(iii)	It is used to generate notifications which are then passed to methods through delegates	It is used as the return type of an event & used in passing messages from event to methods.

Q.2(d) Explain delegate with example. [5]

(A) DELEGATE

```
class program
{
    delegate int MyDelegate (int x, int y);      // delegate
    static int add (intx, int y)
    {
        return (x + y);
    }
    static int mul(int x, int y)
    {
        return (x * y);
    }
    static void main (string [ ] args)
    {
        MyDelegate md; // Reference of delegate
        md = new MyDelegate (add); // function name variable.
        int P = md(10, 20); // function call
        Console.WriteLine ("Sum" + P);
        MyDeletage md1;
        md1 = new MyDelegate (mul);
        P = md1(20, 40);
        Console.WriteLine ("multiplication " + p);
    }
}
```

Note :

- Delegate is a type that enables you to store references to the function.
- Delegates are much like function but with no function body.
- It is declared using delegate keyword.
- The delegate declaration specifies the return type and parameter list.
- After defining a delegate you can define a variable of delegate type & can initialize the variable as reference to function.
- We can call the function using delegate variable by passing arguments to it.

Q.3 Attempt the following (any TWO) **[10]**

Q.3(a) Differentiate between HTML server controls and web server controls. **[5]**

(A)

	HTML Controls	Web Server Controls
(i)	It runs at client side	It runs at server side.
(ii)	You can run HTML controls at server side by adding attribute runat = "server".	You cannot run ASP.Net controls on client side as these controls have this attribute runat = "Server" by default.
(iii)	HTML controls are client side controls, so it does not provide STATE management.	Server side controls provide STATE management.
(iv)	It does not require rendering	It require rendering
(v)	It runs on client side, execution is fast.	It runs on server side, execution is slow.
(vi)	It do not have any separate class for its controls.	It have separate class for its each control.
(vii)	It does not support object oriented paradigm.	It has full support of object oriented paradigm.
(viii)	It can't be accessed from code behind files.	It can be directly worked & accessed from cod behind files.

Q.3(b) Explain the following properties/events of checkbox control. **[5]**

(i) ID **(ii) AccessKey**

(iii) Text **(iv) Checked**

(v) CheckedChanged()

(A) (i) ID :

Gets or sets the programmatic identifier assigned to the server control.

(ii) AccessKey :

Gets or sets the access key that allows you to quickly navigate to the web server control.

(iii) Text :

Gets or sets the text label associated with the checkbox.

(iv) Checked :

Gets or sets a value indicating whether the checkbox control is checked.

(v) CheckedChanged() :

Occurs when the value of the checked property changes between posts to the server.

Q.3(c) What is the difference between inline code and code behind in asp.net? **[5]**

(A)

	Inline code	Code Behind
(i)	It is a page model where server side code is stored in the same file as markup.	A page model where server side code is stored in a separate code file.
(ii)	In inline code model, the code is in <script> blocks in the same .aspx file that contains the HTML and controls.	In Code Behind, the HTML and controls are in the .aspx file, and the code is in a separate .aspx.vb or .aspx.cs file.
(iii)	In this, the .aspx file derives from the Page class.	In Code Behind, the code for the page is compiled into a separate class from which the .aspx file derives.
(iv)	In this, When the page is deployed, the source code is deployed along with the Web Forms page, because it is physically in the .aspx file. However, one does not see the code, only the results are rendered when the page runs.	In Code Behind, all project class files (without the .aspx file itself) are compiled into a .dll file, which is deployed to the server without any source code. When a request for the page is received, then an instance of the project .dll file is created and executed.

(v)	<html
	xmlns="http://www.w3.org/1999/xhtml">
	<head runat="server">
	<title></title>
	</head>
	<body>
	<form id="form1" runat="server">
	
	<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
	
	</form>
	</body>
	</html>
		public partial class _Default :
		System.Web.UI.Page
		{
		protected void Page_Load(object sender, EventArgs e)
		{
		Label1.Text = "Hello.....";
		}
		}

Q.3(d) What is web.config file? Explain <customErrors> and <connectionStrings> tags in web.config file. [5]

(A) It is used to manage various settings that define a website. The settings are stored in XML files that are separate from your application code. In this way you can configure setting independently from your code. Generally a website contains a single web·config file stored inside the application not directory. However there can be many configuration files that manage settings at various levels within an application. Using web·config file we can

- Create a database connection
- Modify Authentication and Authorization
- Set image path for the uploading images
- Manage the larger size files when uploading to the web server
- For email sending.
- <customerror> - This tag is used to handle custom errors at the webpage
- <connectionstring> - This tag is used to specify database connection.

< connection string>
<Add name = "ABC" connection string= "DataSource = (default);
Initial catalog
= 'DB Name"; IntegratedSecurity = true%>
</connection string>

Q.4 Attempt the following (any TWO) [10]

Q.4(a)What is ViewState? Where is the ViewState is stored after the page postback? [5]

(A) The view state is about the state of the page of all its controls. It is automatically maintained across the posts by the ASP.Net framework.

When a page is sent back to the client for output, the changes in the properties of the page and its controls are determined and stored in the value of a hidden input field named _VIEWSTATE. When the page is again post back the _VIEWSTATE field is sent to the server with the HTTP request.

Q.4(b)Explain the relationship between master page and content page. [5]

(A) ASP.NET master pages enable us to create a page layout that we can use with selected or all pages in our website. Master pages can greatly simplify the task of creating a consistent look for our site. A master page can also serve as a template for one or more webforms. Every .aspx page needs only to define the content unique to it and this content will plug into specified area of the master page layout.

The content pages provides the content that's unique to each page in the application. The content of each page is displayed in the master page's content place holder. You can define the content for the master page's place holder controls by creating individual content pages which are ASP.NET pages that are bound to a specific master page. The binding is established in the content page's a page directive by including a MasterPageFile attribute that points to the master page to be used.

For Example :
<% e Page Language – "C#" MasterPageFile = " n/Master.Pages/
Master/.master "Title = " content page" % >

In content page you can create the content by adding content controls and mapping them to contentPlaceholder controls on the MasterPage.

Layout.Master
<% e Master Language = "C#" AutoEventWireUp = "True"
CodeBehind = "Layout.Master.cs" Inherits"
MasterPageApplication.Layout"%">
<head.runat = "server">
<asp: content PlaceHolder ID = "head" runat "server" >
</asp: ContentPlaceHolder>
<style type = "text/css">
</style>
<body>
<form id = "form1" runat = "server">
<asp:conterPlaceHolder ID="ContentHolder1" runat = "server">
</asp: contentPalaceHolder>
</body>
</html>

Contentpage.aspx
<% e Page Title = "" Language = "C#" MasterPageFile = " n/Layout.Master
"AutoEventWireUp = "true"
Codebehind = "ContentPage.aspx.cs" Inherits = "MasterPageApplication. ContentPage"%>
<asp :content ID="Content" ContentPlace Holder ID ="head" runat ="server" > </asp: content
>
<asp:content ID = "Content2" ContentPlaceHolderID = "contentPlaceHolder1" runat =
"server">
<asp:Label ID = "Label3" runat= "server" Text" welcome to BSCIT"> </asp: Label>
</asp:control>

Q.4(c) Explain the following validation controls with example. **[5]**

(i) CompareValidator (ii) CustomValidator (iii) RangeValidator
(iv) RegularExpressionValidator (v) RequriedFieldValidator

(A) <asp:CompareValidator>

Checks if the value is acceptable compared to a given value or compared to the content of another control. In other words, it checks that the value in the validated control matches the value in another control or a specific value. The data type and comparison operation can be specified. If the validated control is empty, no validation takes place. The most important properties in the CompareValidator are ValueToCompare, ControlToCompare,Operator, and type.

<asp:CompareValidator id="comvR" runat="server" display="static"
controlToValidate="txtR" errorMessage="Data not matching"
type="Double" operator="DataTypeCheck"></asp:CompareValidator>

Additional property of this control
ValueToCompare – the value to compare with.
ControlToCompare - The control to compare with.

RangeValidator:
<asp:RangeValidator>
Checks if the input control's value is within a specified range. In other words, it checks that the value in the validated control is within the specified text or numeric range. If the validated control is empty, no validation takes place. The most important properties in the RangeValidator are MaximumValue, MinimumValue, and type.

```
<asp:RangeValidator id="ranvDependents" runat="server"
display="static" controlToValidate="txtDependents"
errorMessage="Must be from 0 to 10"
type="Integer" minimumValue=0 maximumValue=10>
</asp:RangeValidator>
```

Additional property of this control
MinimumValue
MaximumValue

RegularExpressionValidator:
<asp:RegularExpressionValidator>
Checks the value against a regular expression (pattern).Checks that the value in the control matches a specified regular expression. If the validated control is empty, no validation takes place.

The most important property in theRegularExpressionValidator is ValidationExpression.

```
<asp:RegularExpressionValidator  id="regvH"
runat="server" display="static" controlToValidate="txtH"
errorMessage="Hours must be 1-3 digits only"
validationExpression="\d{1,3}"></asp:RegularExpressionValidator>
```

CustomValidator:
<asp:CustomValidator>

Allows you to develop custom validation.Performs user-defined validation on an input control using a specified function (client-side, server-side, or both). If the validated control is empty, no validation takes place. The most important property in the CustomValidator is ClientValidationFunction.

```
<asp:CustomValidator id="cusvDeptNum" runat="server"
display="static" controlToValidate="txtDeptNum"
ClientValidateFunction="validateLength"
errorMessage="Length must be greater than or equal to 4" >
</asp:CustomValidator>
```

Examples In JavaScript

```
function validateLength(oSrc, args)
{

args.IsValid = (args.Value.length >= 4);
```

RequriedFieldValidator

General form

< asp : RequiredFieldValidator ID = "RequiredFieldValidator"
runnat = "Server" ErrorMessage = "RequiredFieldValidator">
</asp : RequiredFieldValidator >

It validates the content of the control by ensuring that the control should not be empty.

Q.4(d)Explain Response.Redirect() and Server.Transfer() methods in ASP.NET. [5]

(A) (i) Response.Redirect( )

This method redirects a client to a new URL. Specifies a new URL and whether execution of the current page should terminate.

(ii) Syntax :

public void Redirect (string url, bool endResponse)

Here url is a string value specifies the location of the target

endResponse is a Boolean value indicates whether execution of the current pages should terminate.

(iii) e.g.:

Response.Redirect(http://www.google.com, false);

The above code will redirect you to google page & second argument "false" indicates we do not want an exception to be thrown to terminate the page.

(iv) The Response. Redirect method redirects a request to a new URL and specifies the new URL while the server.transfer method for the current request, terminates execution of the current page and starts execution of a new page using the specifies URL path of the page.

(v) You can't use Transfer() to send the user to another website or to a non-ASP.net page (such as an HTML page) The transfer() method only allows you to jump from one ASP.net page to another, in the same web application.

(vi) When you use transfer() the user won't have any idea that another page has taken over, because the browser will still show the original URL.

Q.5 Attempt the following (any TWO) [10]

Q.5(a) Explain the various authentication mechanisms in ASP.NET. [5]

(A) There are three types of authentication in ASP.Net.

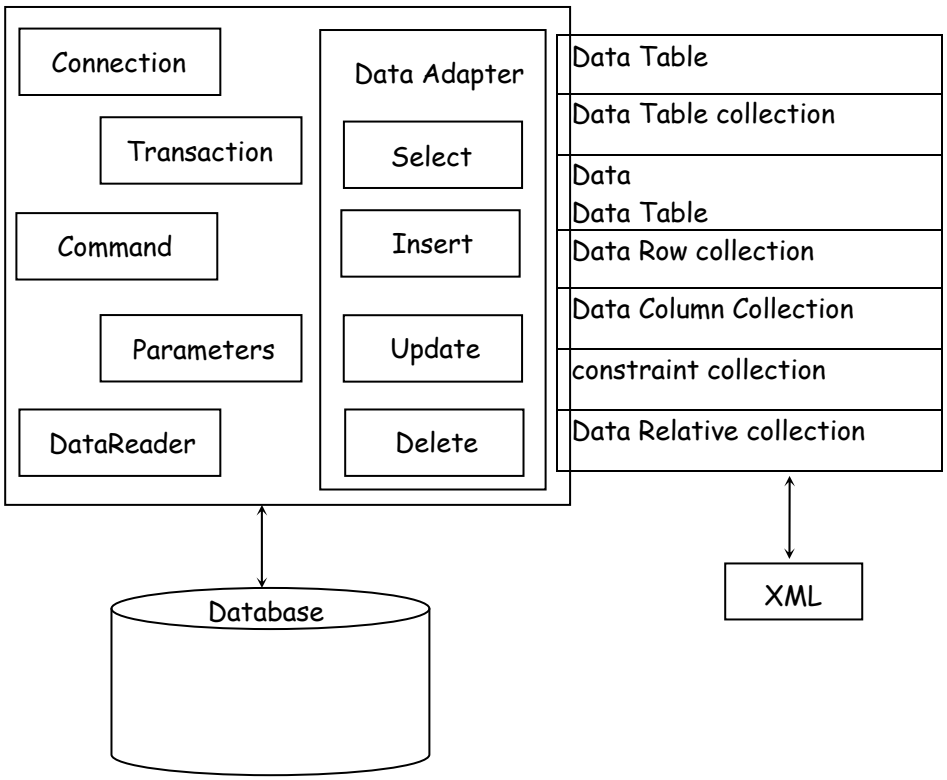
(i) Windows Authentication : In this methodology ASP.Net Web pages will use local window users & groups to authenticate & authorize resources.

(ii) Forms Authentication : This is a cookie based authentication. Where username & password are stored on client machines as cookie files or they are sent through URL for every request. Form based authentication presents the user with an HTML–based Web page that prompts the user for credentials.

(iii) Passport Authentications : Passport authentication is based on the passport website provided by the Microsoft. So when user logins with credentials it will be reached to the passport website where authentication will happen. If Authentication is successful it will return a token to your website.

Q.5(b) Draw and explain the architecture of ADO.NET in brief. [5]

(A)



The database access technologies including components for retrieving data, storing data in memory and binding data to controls are collectively available through ADD.Net. It provides a disconnected architecture where applications connect to the database only once to retrieve data and store it in memory. Then get disconnected from the database and modifies the in memory copy file of the data. If the database needs to be updated with charges made to the in memory copy, a new connection is made and the database is updated. This means that database can support hundreds of users. It provides the classes for developing database applications with .NET languages.

- Data Provider : It is used for providing and maintaining the connection to the database. It is a set of components that are working together to provide data in an efficient manner. Each Data provider consists of the following component classes.
- Connection : It provides a connection to a database.
- Command : It provides an individual SQL statement
- DataReader : It provides read-only, forward only, access to the connected data in a database.
- DataAdapter : It provides a link between the connection, command and a dataset object. It gives a discontented DataSet with data.

Q.5(c) Differentiate between DataSet() and DataReaderQ [5]

(A)

	DataSet	DataReader
(i)	Data set is a collection of in memory tables	It is used to retrieve read only and data from a database. It provides the ability to expose the data from database.
(ii)	It fetches all the rows at a time into its memory area	It fetches one row at a time so very less network cost.
(iii)	It releases the data after loading all the data in memory	It fetches the data from the database and stores it in network buffer and gives whenever requested.
(iv)	There are more system overheads on all the data is fetched at a time	It increases application performance and reduced system overheads.

(v)	Data can be fetched from multiple labels	Data can be fetched from single table
(vi)	We can move back and forth and fetch records randomly as per requirement	We can move forward only hence we cannot fetch random records.
(vii)	Insert, Update, Delete are possible in this	It is read only so no insert, update, delete possible
(viii)	It is disconnected architecture	It is connected architecture.
(ix)	Automatically handles the opening and closing of connection.	Requires connection to be opened and closed manually
(x)	Dataset can be serialized and represented in XML so easily passed around to others.	DataReader cannot be serialized.

Q.5(d) Explain the properties and methods of Command Object [5]

(A) Properties :

- Command Text : Sets or returns a provider command.
- Command Type : Sets or returns the type of a Command object.
- State : Returns a value that describes if the command object is open, closed, connecting, executing or retrieving data.

Methods :

Cancel : Cancels an execution of a method

CreateParameter : Creates a new Parameter object

Execute : Executes the query, SQL statement or procedure in the commandText property.

Q.6 Attempt the following (any TWO) [10]

Q.6(a) What role does the ScriptManager play? Explain the UpdatePanel Control. [5]

(A) Role of ScriptManager :

It manages client script for AJAX–enabled ASP.Net web pages. It registers the script for the Microsoft Ajax Library with the page. This enables client script to use the type system extensions and to support features such as partial page rendering & Web service calls.

Update Panel Control :

It enables you to build rich, client–centric Web applications. By using update Panel controls, we can refresh selected parts of the page instead of refreshing the whole page with a postback.

This is referred to as performing a partial page update. An ASP.Net web page that contains a scriptManager control and one or more updatePanel controls can automatically participate in partial–page updates; without custom client script.

Q.6(b) Explain the jQuery syntax and Document.Ready Event with the example. [5]

(A) The jQuery syntax is tailor–mode for selecting HTML elements and performing some action on the element(s).

Syntax : $(selector).action()

- A $ sign to define/access jQuery
- A (selector) to "query (or find)". HTML element
- A jQuery action() to be performed on the element(s).

e.g.

$(this).hide() : Hides the current element

Document.Ready Event :

```
$(document).ready(function( ) {    });
```

All jQuery methods are inside a document ready event. This is to prevent any jQuery code from scanning before the document is finished loading(is ready).

```
e.g. $(document).ready(function(){
$("button").click(function(){
    $("p").slideToggle();
    });
});
```

The ready event occurs when the DOM has been loaded. Because this event occurs after the document is ready, it is a good place to have all other jQuery events and functions. The ready() method specifies what happens when a ready event occurs.

Q.6(c) Explain DOM manipulation methods in jQuery. [5]

(A) DOM stands for document Object Model & is a mechanism for representing and interacting with your HTML, XHTML or XML documents. It allows you to navigate and manipulate your documents through a programming language, which in the browser will almost always be JavaScript.

jQuery provides various methods to add, edit or delete DOM element(s) in the HTML page.

Following is the list of methods :

text( ) : can read or write the text conside a given HTML element.
Val() : used to get & set the value of form fields.
altr() : Can read & write attributes of HTML elements.
prepend() : It inserts new HTML into the beginning of the selected HTML elements.
before() : It inserts HTML before the selected element (outside the element)
wrap() : Can wrap the selected HTML element in another HTML element.
replaceWith() : It replaces the selected HTML element with a new one.
remove() : It removes the selected HTML element from the DOM.

Q.6(d) Write and explain the basic syntax of a LINQ query in C#. [5]

(A) LINQ Query Syntax

LINQ query syntax is a set of query keywords built into the .NET framework (3.5 and higher) that allow the developer to write SQL style commands in line straight in the code editor, without the use of quotes.

The .NET framework 3.5 introduces us to the following query keywords;

- from / in - Specifies the data source
- where - Conditional boolean expression (e.g. i == 0)
- orderby (ascending/descending) - Sorts the results into ascending or descending order
- select - Adds the result to the return type
- group / by - Groups the results based on a given key

```
from [identifier] in [source collection]
let [expression]
where [boolean expression]
order by [[expression](ascending/descending)], [optionally repeat]
select [expression]
group [expression] by [expression] into [expression]

private readonly List<Character> Characters = new List<Character>
{
//Format: Character Name, Gender, Number of episodes
new Character("Charlie", Gender.Male, 162),
new Character("Alan", Gender.Male, 162),
```

```
new Character("Berta", Gender.Female, 117),
new Character("Jake", Gender.Male, 162),
new Character("Evelyn", Gender.Female, 78),
new Character("Judith", Gender.Female, 69)
};
```

Example :

```
IEnumerable<Character> results = from character in Characters where character.Episodes > 120
select character;
```

- Specify the data source (Characters) and a temporary variable for each element within the data source (in this case a Generic List)
- Create a condition with a boolean result used to query the data source
- If the condition is met, select the temporary variable and add it to the results collection (IEnumerable<Type>)

Q.7 Attempt the following (any THREE) [15]

Q.7(a) What is constructor? Explain parameterized constructor with suitable example. [5]

(A) C# supports a special type of method called a constructor that enables an object to initialize itself when it is created. Constructions have the same name as that of class and they do not specify a return type not even void. Constructors are usually public because they are used to create objects. They can also be private and protected. In some cases the objects of that class cannot be created and also class cannot be used as a baseclass for inheritance.

A constructor with at least one parameter is called as parameterized constructor. In parameterized constructor we can initialize each instance of the class to different values. While creating a object for parameterized constructor one should provide a parameter along the declaration.

Following is the example of parameterized constructor

```
using system
class paramConst.
{
        public void ParamConst()
        {
            Console.WriteLine("Default Constructor");
        }
        public void ParamConst (int a.)
        {
            console.WriteLine("Parameterized constructor");
        }
}
class MainClass
{
        public static void Main()
        {
                ParamConst Pmc1 = new ParamConst();
                ParamConst Pmc2 = new ParamConst(25);
        }
}
```

Q.7(b) Explain following properties : [5]

(i) AutoPostBack (ii) AutoEventWiredUp

(A) (i) AutoPostBack

AutoPostback is a mechanisam by which the page will be posted back to the server automatically based on some events in the web controls. In some of the webcontrols the property called auto postback which if set to true will send the request to the server when an event happens in the control. For normal client contols like list controls with AutoPostBack set to false, when a user choose an item in the list the brower doesnot communicate with the server. There is no network traffic and hence no delay for the user before they see the result of the choise but there is also no oppourtunity to do anything in your server side code like calculate dependent values. If we want to do anything to the server.

(ii) AutoEventWiredUp

The ASP-NEI page framework support an automatic way to associate page events and methods. If the AutoEvent wireup attribute of the page directive is set to true, the page farmework calls page events automatically speifically the page-Init are page-load methods. In that case no explicit Handles clause or delegate is needed

- AutoEventwireUp is an attribute in page directive.
- AutoEventwireUp is a Boolean attibute that indicates wheather the ASP Net pages events are autowire.
- AutoEventwireUP will have a value true or false. By default is true.
  </e page Language = "C#" AutoEventWireUp = "True"
  codeFile = "Default. aspx"cs" Inherits = "Default"
  AutoEventwireup attribute can be delcalred in the <pages> section in the machine.config file or the web.config file.

Q.7(c) What are the event handlers that we can have in global.asax file? [5]

(A) Events in the Global. asax file :

- Application_Init : It is fixed when an application initialize the first time.
- Application_Start : It is fixed the first time when an application starts.
- Session_Start : It is fixed the first time when a user's session is started. This typically contains for session initialization logic code.
- Application_BeginRequest : It is fixed each time a new request comes in.
- Application_End Request : It is fixed when the application terminates.
- Application_Error : It is fixed when an unhandled error occurs written the application.
- Session_End : It is fixed wherever a single user. session ends or times out.
- Application_End : It is the last event of its kind that is fixed when the application ends or times out. It typically contains application cleanup logic.

Q.7(d) What is website navigation? Explain structure of web.sitemap file with suitable example [5]

(A) Website navigation ASP.Net site navigation enables you to store links to all of your pages in a central location, and render those links in lists or navigation menus on each page by including a specific web server control.

Structure of Web sitemap file.

```
<sitemap xmlns="http://schemas.microsoft.com/Asp.Net/SiteMap-file-1.0">
<SiteMapNode url = "~/" title = "home" description = "MUMBAI UNIVERSITY, KALINA">
<SiteMapNode url = "Default.aspx" title = "Home" description = "Go to the homepage"/>
<SiteMapNode url = "IT.aspx" title = "IT Department" description = "BSCIT
INFORMATION"/>
</SiteMapNode>
```

```
<Site MapNode url = "COMMERCE.aspx" title = "COMMERCE DEPARTMENT" description =
"COMMERCE DEPARTMENT INFORMATION">
<SiteMapNode url = "BFM.aspx" title = "BFM" descriptions = "BFM INFORMATION"/>
<SiteMapNode url = "BBI.aspx" title = "BBI" descriptions = "BBI INFORMATION"/>
</SiteMapNode>
</SiteMap>
```

Q.7(e) Explain DataAdapter with its properties and methods. **[5]**

(A) The DateAdapter has properties of type command, which represent the ways it can query, insert, delete & update the database.

	Property	**Description**
(i)	Delete Command	Represents a DELETE statement or stored procedure for deleting records from the data source.
(ii)	Insert Command	Represents an INSERT statement or stored procedure for inserting a new record to the data source.
(iii)	Select Command	Select statement or stored procedure can be used to select records from a data source.
(iv)	Update Command	Update statement or stored procedure for updating recording in a data source.
(v)	Table Mappings	Represents a collection of mappings between actual data source table & a Data Table object.

DataAdapter Methods

	Method	**Description**
(i)	Fill	Fills data records from a DateAdapter to a Dataset Object
(ii)	FillSchema	adds a DataTable to a DataSet.
(iii)	GetFillParameter	Retrives parameters that are used when a SELECT statement is executed.
(iv)	Update	Stores data from a data set to the data source.

Q.7(f) What is AJAX? What are the advantages and disadvantages of AJAX? **[5]**

(A) AJAX stands for Asynchronous JavaScript and XML. This is a cross platform technology which speeds up response time. The AJAX server controls add script to the page which is executed and processed by the browser.

Advantages of Ajax :

- Better interactivity : Ajax allows easier and quicker interaction between user & website as pages are not reloaded for content to be displayed.
- Easier navigation : Ajax applications on websites can be built to allow easier navigation to users in comparison to using the traditional back and forward button on a browser.
- Compact : With AJAX, several multipurpose applications & features can be handled using a single web page, avoiding the need for clutter with several web pages.

Disadvantages :

- The back and refresh button are rendered useless
  With AJAX, as all functions are loaded on a dynamic page without the page being reloaded or more importantly a URL being changed, clicking the back or refresh button would take you to an entirely different web page or to the beginning of what your dynamic web page was processing.
- It is built on javascript.
- Open source
- ActiveX requests are enabled only in I.E. or never latest browser.

❑ ❑ ❑ ❑ ❑