

Advanced Web Programming

Mumbai University Examination Paper Solution: Apr-19

Q.P. Code: 65491

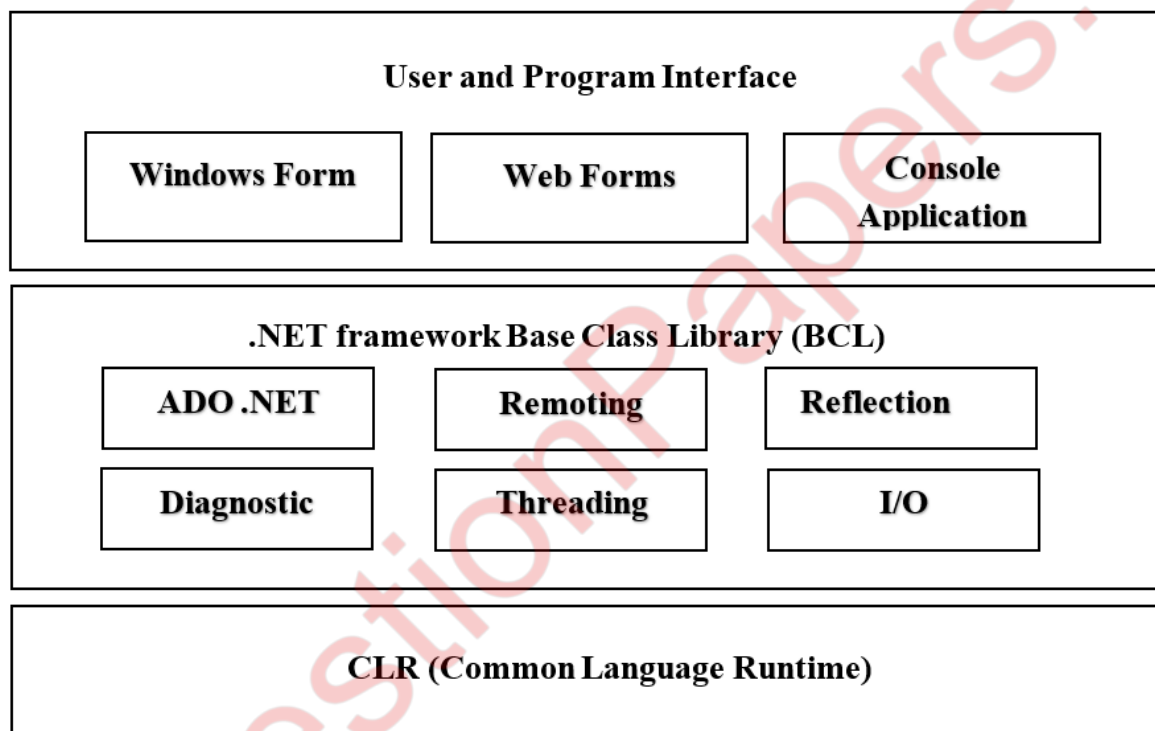
Q1. Attempt any three of the following.

[15]

Q1(a) Draw and Explain .NET framework architecture.

(5)

Ans.



Following are the components of .Net Framework:

A. User and program Interface:

- It specifies the type of application that can generated using .Net.
- We can create following:
 - i. Console application:
 - These are the application where input and output are through console command.
 - There are no forms, no internet connectivity required.
 - ii. Windows application:
 - Here input and output are through windows form.

— No internet connectivity required.

iii. Web application:

— Here input and output are through web forms.

— Int needs remote[server] processing, needs internet connectivity.

— Net primarily used to create web application.

B. Base Class Libraries:

- .NET frame supports rich base class libraries to provide access to system functionalities.
- BCL is the foundation on which .NET framework application its components and controls are build.
- Using BCL we can create types of application such as windows application, console application, web application, etc.
- Developers just need to import the BCL in their language code and uses its predefine methods to create application.

C. CLR [Common Language Runtime]:

- It is virtual machine component of .NET framework.
- It manages creation, compilation, memory allotment, execution, garbage collection of .NET application.
- It ensures conversion of sources code to MSIL [Microsoft Intermediate Language] by compiler.
- It also ensures the JIT conversion of your MSIL code to machine code.
- CLR provides additional service like memory management, execution handling garbage collection, security and thread management.
- All programs written in .NET framework irrespective of language are managed by CLR.

Q1(b) Elaborate Array memory representation with an example. (5)

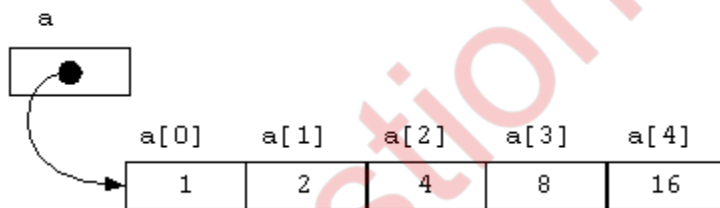
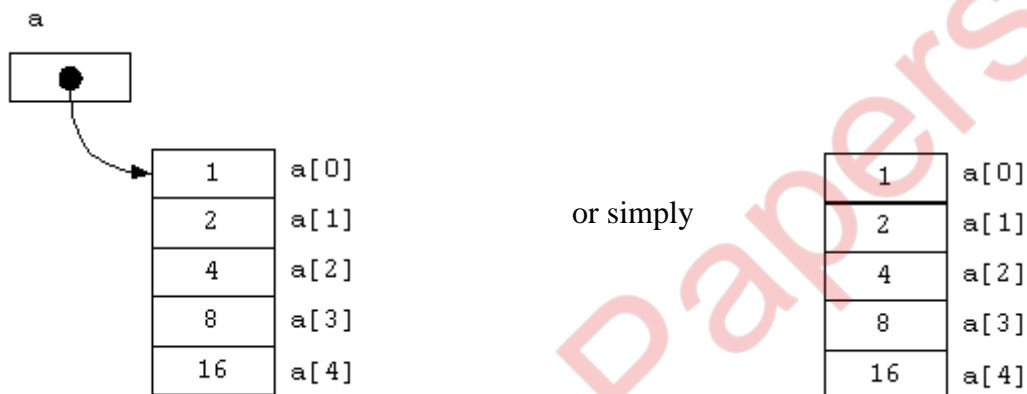
Ans. An array is a collection of homogeneous (same type) data items stored in contiguous memory locations. For E.g. if an array is of type “int”, it can only store integer elements and cannot allow the elements of other types such as double, float, char etc.

Array Memory representation,

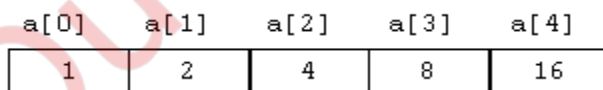
Arrays are often represented with diagrams that represent their memory use. The diagram below is one typical way to represent the memory used by an array.

Each box represents the amount of memory needed to hold one array element. For int's this is usually 4 bytes. We can write the value of an element inside the box. Pointers hold the memory address of other data and are represented by a black disk with an arrow pointing to the data it references.

The actual array variable, a in this example, is a pointer to the memory for all of its elements. A common shortcut is to omit writing the array variable, and write only the elements that hold the data because these are usually the focus of attention. Sometimes the cells are written horizontally, especially when showing C-strings (arrays of characters).



or just



Here is an array declaration and code to initialize it.

```
Int a[5]; // Allocates memory for 5 imts.
```

.....

```
a[0] = 1;
```

```
for (int i=1; i<5; i++)
```

```
{
```

```
    a[i] = a[i-1] *2;
```

```
}
```

Q1(c) Explain any 5 properties/methods of Math class.

(5)

Ans.

Methods	Description
<u>Math.abs()</u>	It will return the Absolute value of the given value.
<u>Math.max()</u>	It returns the Largest of two values.
<u>Math.min()</u>	It is used to return the Smallest of two values.
<u>Math.round()</u>	It is used to round of the decimal numbers to the nearest value.
<u>Math.sqrt()</u>	It is used to return the square root of a number.
<u>Math.cbrt()</u>	It is used to return the cube root of a number.
<u>Math.pow()</u>	It returns the value of first argument raised to the power to second argument.
<u>Math.signum()</u>	It is used to find the sign of a given value.
<u>Math.ceil()</u>	It is used to find the smallest integer value that is greater than or equal to the argument or mathematical integer.
<u>Math.copySign()</u>	It is used to find the Absolute value of first argument along with sign specified in second argument.

Q1(d) Give details about Type Conversion.

(5)

Ans. It is a process of converting value of one type into another type. It is also known as type casting.

Type conversion are of two types:

I. Implicit type conversion:

- It is automatically done by C# in type safe manner.
- Conversion of lower type to higher type is done implicitly.
- No casting required
- No loss of precision.

E.g. `int x = 10;`
`float y = x; // implicit`

II. Explicit type conversion:

- Here the casting operator/ conversion function is explicitly required to perform casting.
- Conversion of higher type to lower type is done explicitly.
- There is loss of precision.

E.g. `double p= 12.6;`
`int x =(int)p;`

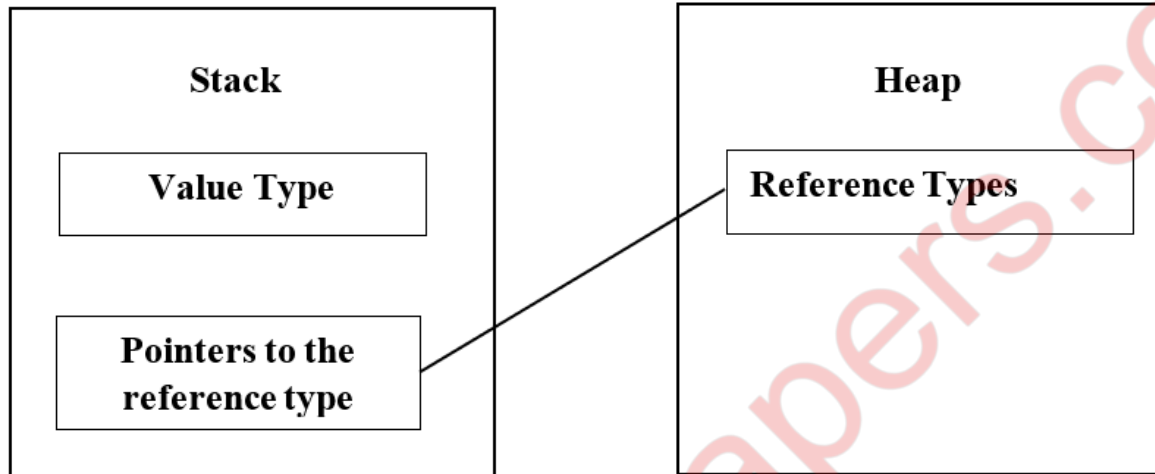
C# provides many conversion function. They are:

- `ToBoolean`= Converts a type to a Boolean where, possible.
- `ToByte`= Converts type to a byte.
- `ToChar`= Converts a type to a single Unicode character, where possible.
- `ToDateTime`= converts a type (integer to string) to date-time structure.
- `ToDecimal`= Converts floating point or integer type to decimal.
- `ToDouble`= Converts a type to a double type.
- `ToInt16`= Converts a type to 16-bit integer.
- `ToInt32`= Converts a type to 32-bit integer.
- `ToInt64`= Converts a type to 64-bit integer.
- `ToString`= converts a type to string.
- `ToSingle`= Converts a type to small floating point number.

Q1(e) Write short note on Value and Reference type.

(5)

Ans. The Types in .NET Framework are either treated by Value Type or by Reference Type. A Value Type hold the data within its own memory allocation and a reference type contains a pointer to another memory location that holds the real data. Reference type variables are stored in the heap while value type variables are stored in the stack.



Value Type:

- A value type stores its contents in memory allocated on stack.
- When you create a value type, a single in memory is allocated to store the value and that variable directly holds a value.
- If you assign it to another variable, the value is copied directly and both variable works independently.
- Predefined data types, structure, Enums are also value types, and work in the same way.
- Value types can be created at compile time and stored in stack memory, because of this, Garbage Collector can't access the stack.

E.g. `int x =10;`

Here the value 10 is stored in an area of memory called the stack.

Reference Type:

- Reference types are used by a reference which holds an address of the object but not the object itself.
- Because reference types represent the address of the variable rather than the data itself, assigning a reference variable to another doesn't copy the data.

- Instead it creates a second copy of the reference, which refers to the same location of the heap the original value.
- Reference type variables are stored in different area of memory called the heap.
- This means that when a reference type variable is no longer used, it can be marked for garbage collection,
- Example of reference types are classes, objects, arrays, indexes, interfaces, etc.

E.g. `int[] a= new int[20];`

In above code the space required for the 20 integers that make up the array is allocated on the heap.

Q1(f) Explain static members and partial class.

(5)

Ans. Static members:

- Static members declared in a class aren't associated to a particular instance, rather, they are associated at the class level.
- The static members can be used in a program using reference to a class rather than instances.

E.g.

```
class demo StaticUse{
    static int age=10;
}
```

- You needn't create an instance to access static members. If the field needs to be referenced by another class, it will reference as:

StaticUse.age

- All instances of a class share the same static member. In other words, static members are not unique to each instance.
- Static members are automatically initialized to their default value if they aren't explicitly initialized.
- Not only fields and methods, the whole class can be declared static. If done so, all the fields and methods need to be declared static.
- A non-static class contains static as well as non-static members. A non-static class cannot access static members through instance of the class.

- A static method can call another static method but cannot call any instance method of the class.
- A static method can access static field but cannot access non static field.
- Non static methods can make use of static methods and static fields.

Partial Class:

- In C#, you can split the implementation of a class, a struct, a method, or an interface in multiple .cs files using the partial keyword.
- The compiler will combine all the implementation from multiple .cs files when the program is compiled.

Example: Partial Class

```
public class Employee
{
    public int EmpId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public int Age { get; set; }

    public Employee(int Id, string name)
    {
        this.EmpId = Id;
        this.Name = name;
    }

    public void DisplayEmployeeInfo()
    {
        Console.WriteLine(this.EmpId + " " this.FirstName + " " + this.LastName);
    }

    public void Save(int id, string firstName, string lastName)
    {
        Console.WriteLine("Saved!");
    }
}
```


— **Rules for Partial Classes**

- All the partial class definitions must be in the same assembly and namespace.
- All the parts must have the same accessibility like public or private, etc.
- If any part is declared abstract, sealed or base type then the whole class is declared of the same type.
- Different parts can have different base types and so the final class will inherit all the base types.
- The Partial modifier can only appear immediately before the keywords class, struct, or interface.
- Nested partial types are allowed.

Q2. Attempt any three of the following.

[15]

Q2(a) List and explain any 5 templates to create ASP.NET application.

(5)

Ans. Following are the list of templates to create ASP.NET application.

- ASP.NET Web Forms Application
- ASP.NET Empty Web Application
- ASP.NET MVC 3 Web Application
- ASP.NET Dynamic Data Entities Web Application
- ASP.NET Server Control

1. ASP.NET Web Forms Application

Use this project template to create a web application that is based on ASP.NET Web Forms pages and that includes the following functionality. You can choose not to use any of these features when they are not required for your application.

- A master pages.
- A cascading style sheets.
- Login security that uses the ASP.NET membership system.
- Ajax scripting that uses jQuery.
- Navigation that uses a menu control.

2. ASP.NET Empty Web Application

This template creates an ASP.NET web application that includes a Web. config file, but no other files. Use this project template when you do not require the functionality built into the standard template.

3. ASP.NET MVC 3 Web Application

Use this project template to create web applications that use a model-view-controller pattern, using the ASP.NET MVC 3 release. The MVC pattern helps separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements. In addition, this project template promotes test-driven development (TDD).

4. ASP.NET Dynamic Data Entities Web Application

Use this project template to create a Dynamic Data web application that uses the ADO.NET Entity Framework. This type of Dynamic Data web application can target any relational database. Dynamic Data automatically generates a functional web application for viewing and editing data, based on the schema of the data.

5. ASP.NET Server Control

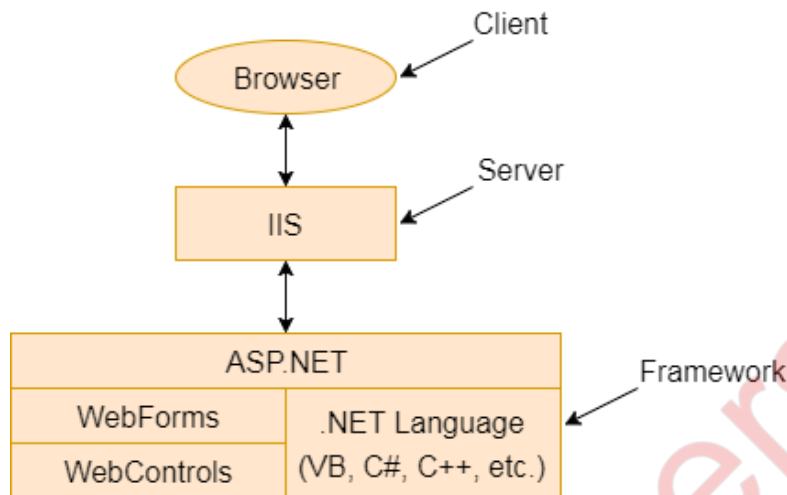
Use this project template to create a custom ASP.NET web server control that can be used in ASP.NET Web Forms pages.

Q2(b) Explain Anatomy of WebForm. (5)

Ans. Web Forms are web pages built on the ASP.NET Technology. It executes on the server and generates output to the browser. It is compatible to any browser to any language supported by .NET common language runtime. It is flexible and allows us to create and add custom controls.

We can use Visual Studio to create ASP.NET Web Forms. It is an IDE (Integrated Development Environment) that allows us to drag and drop server controls to the web forms. It also allows us to set properties, events and methods for the controls. To write business logic, we can choose any .NET language like: Visual Basic or Visual C#.

Web Forms are made up of two components: the visual portion (the ASPX file), and the code behind the form, which resides in a separate class file.



The main purpose of Web Forms is to overcome the limitations of ASP and separate view from the application logic. ASP.NET provides various controls like: server controls and HTML controls for the Web Forms. We have tables all these controls below.

Server Controls

The following table contains the server-side controls for the Web Forms.

Control Name	Applicable Events	Description
Label	None	It is used to display text on the HTML page.
TextBox	TextChanged	It is used to create a text input in the form.
Button	Click, Command	It is used to create a button.
LinkButton	Click, Command	It is used to create a button that looks similar to the hyperlink.
ImageButton	Click	It is used to create an imagesButton. Here, an image works as a Button.

HTML Controls

These controls render by the browser. We can also make HTML controls as server control. we will discuss about this in further our tutorial.

Controls Name	Description
Button	It is used to create HTML button.
Reset Button	Resets all other HTML form elements on a form to a default value
Submit Button	Automatically POSTs the form data to the specified page listed in the Action attribute in the FORM tag
Text Field	Gives the user an input area on an HTML form
Text Area	Used for multi-line input on an HTML form

Q2(c) Write a note on page class.

(5)

Ans. Every web page is custom class that inherits from System.Web.UI.Page. By inheriting from this class, your web page class acquires a number of properties and methods that your code can use. These includes properties for enabling Caching, validation, and tracing.

Property	Description
IsPostBack	This Boolean property indicates whether this is first time the page is being run(false) or whether the page is being resubmitted in response to a control event, typically with stored view state information(true)
EnableViewState	When set of false, this overrides the EnableViewState property of the contained controls, thereby ensuring that no controls will maintain state information.
Application	This collection holds information that's shared between all users in your website.
Session	This collection holds information for single user, so it can be used in different pages.
Cache	This collection allows you to store objects that are time-consuming to create so they can be reused in other pages or for other clients. This technique, when implemented property, can improve performance of your web page.
Request	This refers to an HttpRequest object that contains information about the current web request. You can use the HttpRequest object to get information about the user's browser, although you will probably prefer to leave these details to ASP.NET.

Response	This refers to an HttpResponse object that represents the response ASP.NET will send to the user's browser.
Server	This refers to an HttpServerUtility object that allows you to perform a few miscellaneous tasks.
User	If the users have been authenticated, this property will be initialized with user information.

Q2(d) Explain any 5 properties of ListBox and Drop Down List controls. (5)

Ans. LisBox:

Property	Description
AllowSelection	Gets a value indicating whether the ListBox currently enables selection of list items.
BorderStyle	Gets or sets the type of border drawn around the list box.
ColumnWidth	Gets or sets the width of columns in a multicolumn list box.
HorizontalExtent	Gets or sets the horizontal scrolling area of a list box.
HorizontalScrollBar	Gets or sets the value indicating whether a horizontal scrollbar is displayed in the list box.
ItemHeight	Gets or sets the height of an item in the list box.
Items	Gets the items of the list box.
MultiColumn	Gets or sets a value indicating whether the list box supports multiple columns.
ScrollAlwaysVisible	Gets or sets a value indicating whether the vertical scroll bar is shown at all times.
SelectedIndex	Gets or sets the zero-based index of the currently selected item in a list box.

Drop Down List:

Property	Description
Items.Count	Return the total count of items in the dropdownlist.
Items.Add	Add the new item in to the dropdownlist control.
Items.Insert	Add the new item at specific position in dropdownlist control.

Items.Remove	Remove the item from the dropdownlist control.
Items.RemoveAt	Remove the item from the dropdownlist at desired position index.
Items.Clear	Clear all the items from dropdownlist control.
SelectedItem.Text	Returns the Text value of selected item of the dropdownlist.
SelectedValue	Returns the Value property of the selected item of the dropdownlist.
SelectedIndex	Returns the Index of selected item of dropdownlist. (Index always start from Zero).

Q2(e) Write note on AdRotator control. (5)

Ans. The AdRotator is one of the rich web server control of asp.net. AdRotator control is used to display a sequence of advertisement images as per given priority of image.

Adrotator control display the sequence of images, which is specified in the external XML file. In a xml file we indicate the images to display with some other attributes, like image impressions, NavigateUrl, ImageUrl, AlternateText.

In a Adrotator control images will be changed each time while refreshing the web page.

The Advertisement File:

Syntax:

```
<Advertisements>
<Ad>
<!-- First ad here. -->
</Ad>
<Ad>
<!-- Second ad here. -->
</Ad>
</Advertisement>
```

Advertisement File elements:

Properties	Description
ImageUrl	It is describe the path of image which will be displayed in adrotator control.
NavigateUrl	Describe the navigation webpage link when user click on ad Image.
AlternateText	The text will be displayed when image can not be displayed.

Impressions	Describe the display rate of image or priority of image
Keyword	It is a category of Ad.

The AdRotator Class:

The actual AdRotator class provides a limited set of properties you specify both the appropriate advertisement file in the AdvertisementFile property and the type of window that the link should follow (the Target window).

Target	Description
_blank	The link opens a new unframed window.
_parent	The link opens in the parent of the current frame
_self	The link opens in the current frame.
_top	The link opens in the topmost frame of the current window.

Q2(f) Brief about Graphics class and its any 5 methods. (5)

Ans. One of the features of the .NET Framework GDI+, a set of classes designed for drawing images. You can use GDI+ in windows or an ASP.NET application to create dynamic graphics. In a windows application, the graphics you draw would be copied to a window for display. ASP.NET, your code can render the graphics you want and send them directly to the client browser. In general, using GDI+ code to draw a graphics is slower than using a ready-made image file. One should import the system.Drawing namespace, which has the classes that GDI+ uses: using System.Drawing;

You need to follow four basic steps when using GDI+:

1. First you have to create an in-memory bitmap. This is the drawing space where you will create your masterpiece.

```
Bitmap image = new Bitmap(300,50);
```

2. The next step is to create a GDI+ graphics context for the image, which is represented by a System.Drawing.Graphics object. This object provides the methods that allow you to render content to the in-memory bitmap.

```
Graphics g= Graphics.FromImage(image);
```

3. Using the methods of the graphics class, you can draw text, shapes, and images on the bitmap.

4. Drawing methods of the graphic class:

Methods	Description
DrawArc()	Draws an arc representing a portion of an ellipse specified by a pair of coordinates, a width, and a height.
DrawClosedCurve()	Draws a curve and then closed it off by connecting the end points.
DrawCurve()	Draws a curve (technically, a cardinal spline).
DrawEllipse()	Draws an ellipse defined by a bounding rectangle specified by a pair of coordinates, a height, and a width.
DrawLine() and DrawLines()	Draws one or more lines. Each line connects the two points specified by a coordinate pair.
drawPolygon()	Draws a multisided polygon defined by an array of points.
DrawString()	Draws a string of text in given font.
FillEllipse()	Fills the interior of an ellipse.
FillPolygon()	Fills the interior of a polygon.
FillPath()	Fills the interior of a complex shape that's defined by the path object.
DrawIcon() and DrawIconUnstretched()	Draws the icon represented by an icon object and stretches it to fit a given rectangle.

Q3. Attempt any three of the following.

[15]

Q3(a) Explain Exception handling mechanism in C# with its key features.

(5)

Ans. Exception are problems encountered in program at run time. C# supports exception handling. When an exception occurs in code notification is sent, and exception handling code are specified to handle exception in rational manner.

General structure of Exception Handling Mechanism:

```

try
{ // Risk code goes here. }
catch
{ // An error has been detected. }
finally
{ // Time to clean up, regardless of whether or not there was an error. }

```


To handle exception C# uses following keywords.

- Try:
 - It contains block of code that needs to be monitored for occurrence of exception.
 - If exception occurs within try block there is a chance of being handled.
 - try block can be followed by either catch or finally or both.
- catch:
 - it contains exception handling code.
- throw:
 - used to throw exception manually.
 - Mainly used for throwing user defined exemption.
- finally:
 - contains code that needs to be executed irrespective. Exception is generated or not. Generated exception is handled or not.

Exception handling provides several key features:

- Exceptions are object-based:
 - Each exception provides a significant amount of diagnostic information.
 - These exception objects also support an InnerException property that allows you to wrap a generic error over the more specific error that caused it.
- Exception are caught based on their type:
 - This allows you to streamline error-handling code without needing to shift through error codes.
- Exception handlers use as modern block structure:
 - This makes it easy to activate and deactivate different error handlers for different sections of code and handle their error individually.
- Exception handlers are multilayered:
 - You can easily layer exception handlers on top of other exception handlers, some of which may check for only a specialized set of error.
- Exception are a generic part of the .NET Framework:
 - Exception are completely cross-language compatible.
 - Thus, a .NET component written in C# can throw an exception that you can catch in a web page written in VB.

Q3(b) What are State management techniques in ASP .NET?

(5)

Ans. ASP.NET provides us with 2 ways to manage the state of an application. It is basically divided into the 2 categories:

1. Client Side State Management.
2. Server Side State Management.

Client Side State Management –

It is a way in which the information which is being added by the user or the information about the interaction happened between the user and the server is stored on the client's machine or in the page itself. The server resources is not at all utilized during the process.

This management technique basically makes use of the following:

a. View State:

View State can be used to maintain the State at a page level. The term "Page Level" means that the information is being stored for a specific page and until that specific page is active (i.e. the page which is being currently viewed by the user). Once the user is re-directed or goes to some other page, the information stored in the View State gets lost.

b. Hidden Fields:

ASP.NET provides a server control called "Hidden Field" which can be used to store a value at a page level, which is similar to a View State. The value of the Hidden Field is sent along in the HTTP Form Collection along with the value of other controls.

c. Query String:

A Query String is a string which is appended to the end of the Page URL. It is very simple to use and can be used to send data across pages. It stores information in a key - value pair. A "?" signature is used to append the key and value to the page URL.

d. Cookies:

ASP.Net provides another way of state management, which is by using Cookies. Cookies are one of the best ways of storing information. It is nothing but a text file

which is stored on the client's machine. When the user sends a request to the server, the server creates a cookie and attaches a header and sends it back to the user along with the response. The browser accepts the cookie and stores it at a specific location on the client's machine. Even large sites like Gmail, Facebook, Yahoo use cookies.

Server Side State Management-

It is another way which ASP.NET provides to store the user's specific information or the state of the application on the server machine. It completely makes use of server resources (the server's memory) to store information.

This management technique basically makes use of the following,

a. Application State:

If the information that we want to be accessed or stored globally throughout the application, even if multiple users access the site or application at the same time, then we can use an Application Object for such purposes. It stores information as a Dictionary Collection in key - value pairs. This value is accessible across the pages of the application / website.

b. Session State:

Session is one of the most common way which is being used by developers to maintain the state of the application. The Session basically stores the values as a dictionary collection in key/value pairs. It completely utilizes server resources to store the data. It is a secure way of storing data, since the data will never be passed to the client. For each and every user, a separate Session is created, and each and every Session has its Unique ID. This ID is being stored in the client's machine using cookies. If there are multiple users who are accessing a web application, then for each user a separate Session is created. If a single user logs in and logs out the Session is killed, then if the same user again logs into the application, then a new Session ID is being created for the same user.

Q3(c) Elaborate cookie with suitable code snippet.

(5)

Ans. Cookies provide another way to store state information for later use. Cookies are small files that are created in the web browser's memory or on the client's hard drive. One advantage of cookies is that they work transparently, without the user being aware that information needs to be stored. Cookies are easily accessible and readable if the user finds and opens the corresponding file. Some users disable cookies on their browser, which will cause problems for web applications that require them.

Step to use cookies:

- 1) One should import the System.Net namespace so you can easily work with the appropriate types:

```
Using System.Net;
```

- 2) To set a cookie, just create a new HttpCookie object. It can be then filled with string information and can be attached to the current web response:

```
// Create the cookie object.  
HttpCookie cookie = new HttpCookie("priority");  
// Set a value in it.  
Cookie["SubjectPriority"] = "ASP";  
// Add another value.  
cookie ["Country"] = "INDIA";  
// Add it to the current web response.  
Response.Cookie.Add(cookie);
```

- 3) A cookie added in this way will persist until the user closes the browser and will be sent with every request. To create a longer-lived cookie, you can set an expiration date:

```
// This cookie lives for one year.  
cookie.Expires = DateTime.Now.AddYears(1);
```

- 4) You retrieve cookies by cookie name, using the Request.Cookies collection:

```
HttpCookie cookie = new HttpCookie("Priority");
```

- 5) The only way to remove a cookie is by replacing it with a cookie that has an expiration date that has already passed. This code demonstrates the technique:

```
HttpCookie cookie = new HttpCookie("Priority");  
Cookie.Expires = DateTime.Now.AddYears(-1);  
Response.Cookie.Add(cookie);
```

Cookie's common property

1. Domain => Which is used to associate cookies to domain.
2. Secure => We can enable secure cookie to set true(HTTPs).
3. Value => We can manipulate individual cookie.
4. Values => We can manipulate cookies with key/value pair.
5. Expires => Which is used to set expire date for the cookies.

Advantages of Cookie

1. Its clear text so user can able to read it.
2. We can store user preference information on the client machine.
3. Its easy way to maintain.
4. Fast accessing.

Disadvantages of Cookie

1. If user clear cookie information we can't get it back.
2. No security.
3. Each request will have cookie information with page.

Code for cookies:

```
protected void btncreate_Click(object sender, EventArgs e)
{
    Response.Cookies["name"].Value = txtcreatecookie.Text;
    Response.Cookies["name"].Expires = DateTime.Now.AddMinutes(1);
    Label1.Text = "Cookie Created";
    txtcreatecookie.Text = "";
}
```

```
protected void btnretrieve_Click(object sender, EventArgs e)
{
    if (Request.Cookies["name"] == null)
    {
```

```

txtretrieve.Text = "No cookie found";
}
else
{
txtretrieve.Text = Request.Cookies["name"].Value;
}
}

```

Q3(d) what is cross-page posting? Explain with an example. (5)

Ans. A cross-page postback is a technique that extends the postback mechanism so that one page can send the user to another page, complete with all the information for that page. However, if not careful, it can lead you to create pages that are tightly coupled to others and difficult to enhance and debug.

The infrastructure that supports cross-page postback is property named PostBackUrl. To use cross-posting, you simply set PostBackUrl to the name of another web forms. When the user clicks the button, the page will be posted to that new URL with the values from all the input controls on the current page.

Code on Default.aspx

```

<form id="form1" runat="server">
    <asp:TextBox ID="txtName" name="data" runat="server" />
    <asp:Button Text="ClickPostBack" runat="server" PostBackUrl="~/Display.aspx" />
</form>

```

Code on Display.aspx

```

<form id="form1" runat="server">
    <asp:Label Text="text" name="lblValue" ID="lblValue" runat="server" />
</form>

```

Display.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
    if (PreviousPage != null && PreviousPage.IsCrossPagePostBack)
```

```
    {
```

```
        TextBox txtName = (TextBox) PreviousPage.FindControl("txtName");    lblValue.Text =  
"Welcome " + txtName.Text;
```

```
    }
```

```
}
```

Q3(e) Explain Master Page with its uses and working. (5)

Ans. Masterpages are used to implement a consistent layout across all the pages for an application. User can create one masterpage that will hold uniform layout and common controls of the whole application. On this masterpage we can build webpages also known as content page. While creating a web application the master page is built first and then content page is build. The content pages contain contents specific to that pages. The content pages are binded to the master page.

1. The extension of MasterPage is '.master'.
2. MasterPage cannot be directly accessed from the client because it just acts as a template for the other Content Pages.
3. In a MasterPage we can have content either inside ContentPlaceHolder or outside it. Only content inside the ContentPlaceHolder can be customized in the Content Page.
4. We can have multiple masters in one web application.
5. A MasterPage can have another MasterPage as Master to it.
6. The content page content can be placed only inside the content tag.
7. Controls of MasterPage can be programmed in the MasterPage and content page but a content page control will never be programmed in MasterPage.
8. A master page of one web application cannot be used in another web application.

9. The MasterPageFile property of a webform can be set dynamically and it should be done either in or before the Page_PreInit event of the WebForm. Page.MasterPageFile = "MasterPage.master". The dynamically set Master Page must have the ContentPlaceHolder whose content has been customized in the WebForm.
10. The order in which events are raised: Load (Page) a Load (Master) a LoadComplete (Page) i.e. if we want to overwrite something already done in Load event handler of Master then it should be coded in the LoadComplete event of the page.
11. Page_Load is the name of method for event handler for Load event of Master. (it's not Master_Load).

Q3(f) What is Theme? Explain Global theme.

(5)

Ans. All themes are application specific. To use a theme in a web application, you need to create a folder that defines it. This folder needs to be placed in the App_Themes folder, which must be placed inside the top-level directory for your web application. An application can contain definitions for multiple themes as long as each theme is in a separate folder. Only one theme can be active on a given page at a time. To actually make your theme you need to create at least one skin file in the theme folder.

A skin file is a text file with the .skin extension. ASP.NET never servers skin files directly. A skin file is essentially a list of controls tags. The controls tags in skin file don't need to completely define the controls; they need only set the properties that you want to standardize.

Global theme:

A Global theme is a theme that is applied to all the web sites on a web server and includes property settings, and graphics. This theme allows us to maintain all the websites on the same web server and define the same style for all the web pages of the web sites.

Q4. Attempt any three of the following.

[15]

Q4(a) Explain the SQL DATA Provider model.

(5)

Ans. A .NET Framework data provider is used for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, placed in a DataSet in order to be exposed to the user as needed, combined with data from multiple sources, or remoted between tiers. .NET Framework data providers are lightweight, creating a minimal layer between the data source and code, increasing performance without sacrificing functionality.

The following table lists the data providers that are included in the .NET Framework.

.NET Framework data provider	Description
.NET Framework Data Provider for SQL Server	Provides data access for Microsoft SQL Server. Uses the <u>System.Data.SqlClient</u> namespace.
.NET Framework Data Provider for OLE DB	For data sources exposed by using OLE DB. Uses the <u>System.Data.OleDb</u> namespace.
.NET Framework Data Provider for ODBC	For data sources exposed by using ODBC. Uses the <u>System.Data.Odbc</u> namespace.
.NET Framework Data Provider for Oracle	For Oracle data sources. The .NET Framework Data Provider for Oracle supports Oracle client software version 8.1.7 and later, and uses the <u>System.Data.OracleClient</u> namespace.
EntityClient Provider	Provides data access for Entity Data Model (EDM) applications. Uses the <u>System.Data.EntityClient</u> namespace.
.NET Framework Data Provider for SQL Server Compact 4.0.	Provides data access for Microsoft SQL Server Compact 4.0. Uses the <u>System.Data.SqlServerCe</u> namespace.

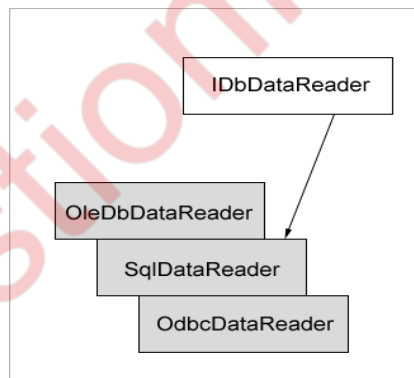
Q4(b) Explain details about DataReader with example.

(5)

Ans. A data reader provides an easy way for the programmer to read data from a database as if it were coming from a stream. The DataReader is the solution for forward streaming data through ADO.NET. The data reader is also called a firehose cursor or forward read-only cursor because it moves forward through the data. The data reader not only allows you to move forward through each record of database, but it also enables you to parse the data from each column. The DataReader class represents a data reader in ADO.NET.

Similar to other ADO.NET objects, each data provider has a data reader class for example; OleDbDataReader is the data reader class for OleDb data providers. Similarly, SqlDataReader and ODBC DataReader are data reader classes for SQL and ODBC data providers, respectively.

The IDataReader interface defines the functionality of a data reader and works as the base class for all data provider-specific data reader classes such as OleDbDataReader, SqlDataReader, and OdbcDataReader. Figure 5-36 shows some of the classes that implement IDataReader.



The DataReader properties

Property	Description
Depth	Indicates the depth of nesting for row
FieldCount	Returns number of columns in a row
IsClosed	Indicates whether a data reader is closed
Item	Gets the value of a column in native format
RecordsAffected	Number of row affected after a transaction

The DataReader methods

METHOD	DESCRIPTION
Close	Closes a DataRaeder object.
Read	Reads next record in the data reader.
NextResult	Advances the data reader to the next result during batch transactions.
Getxxx	There are dozens of Getxxx methods. These methods read a specific data type value from a column. For example. GetChar will return a column value as a character and GetString as a string.

Code for DataReader:

```
SqlCommand myCMD = new SqlCommand("SELECT CategoryID, CategoryName FROM
Categories;" + "SELECT EmployeeID, LastName FROM Employees", nwindConn);
nwindConn.Open();
SqlDataReader myReader = myCMD.ExecuteReader();
do {
    Console.WriteLine("\t{0}\t{1}", myReader.GetName(0), myReader.GetName(1));
    while (myReader.Read()) Console.WriteLine("\t{0}\t{1}", myReader.GetInt32(0), myReader.GetString(1));
}
while (myReader.NextResult());
myReader.Close();
nwindConn.Close();
```

Q4(c) What is Data Binding? Explain its types. (5)

Ans. ASP.NET adds a feature that allows you to skip the repetitive code process to retrieve row of information and pop data directly into HTML elements and fully formatted controls. It's called data binding. Data binding, in the context of .NET, is the method by which controls on a user interface (UI) of a client application are configured to fetch from, or update data into, a data source, such as a database or XML document.

Prior to .NET, access to data binding models was limited to databases. Thus, many database management systems (DBM) could indirectly access the data source through their

application programming interface (API) without any flexibility in controlling the data binding process. This problem is addressed in .NET by providing fine control of how the data is bound and the behavior of UI with Windows Forms and ADO.NET classes in the framework. The development of Web applications is simplified by providing data binding capability to Web pages using .NET server-side Web controls.

Two types of ASP.NET data binding exist,

- Single-Value, or “simple” Data Binding:
 - You can use single-value data binding to add information anywhere on an Asp.NET page.
 - Single-value data binding doesn't necessary have anything to do with ADO.NET.
 - Single-value data binding allows you to take a variable, a property, or an expression and insert it dynamically into a page.
 - Single-value binding also helps you create templates for the rich data controls.
- Repeated-Value, or “List” Binding:
 - Repeated-value data binding allows you to display an entire table.
 - Unlike single-value data binding, this type of data binding requires a special control that supports it.
 - Typically, this is a list control such as CheckBoxList or ListBox, but it can also be a much more sophisticated control such as the GridView.

Q4(d) Write short note on Data Source Controls.

(5)

Ans. Following are the various data source controls in Ado.Net:

- **SqlDataSource:** This data source allows you to connect to any data source that has an ADO.NET data provider. This includes SQL Server, Oracle, and OLE DB or ODBC data sources. When using this data source, you don't need to write the data access code.
- **AccessDataSource:** This data source allows you to read and write the data in an access database file(.mdb). However, its uses are discouraged, because access doesn't scale well to large numbers of users.

- **ObjectDataSource:** This data source allows you to connect to a custom data access class. This is the preferred approach for large-scale professional web applications, but it forces you to write much more code.
- **XmlDataSource:** This data source allows you to connect to an XML file.
- **SiteMapDataSource:** This data source allow you to connect to a .sitemap file that describes the navigational structure of your website
- **EntityDataSource:** This data source allows you to query a database by using the LINQ to Entities feature.
- **LinqDataSource:** This data source allows you to query a database by using the LINQ to SQL feature, which is a similar (but somewhat less powerful) predecessor to LINQ to Entities.

Q4(e) Explain the ways of formatting the GridView Data for display. (5)

Ans. The GridView server control displays data provided by a data source in a tabular format. It renders its data as an HTML table. The GridView control supports automatic paging, sorting, deleting, and selecting. To enable row selection `AutoGenerateSelectButton` property of GridView can be set to true. The GridView data can be sorted based on any or all of its columns. To sort GridView, following are the steps:

- Set `AllowSorting` attribute of GridView to true.
- Also set `SortExpression` property of columns to respective field from database to sort.

Paging refers to the ability of GridView control to display bound data one page at a time. Users can arbitrarily select pages from a GridView. To enable paging feature of a GridView:

- Set `AllowPaging` property of GridView control to true.
- Set `PageSize` property to no of records we want in page.

Example:

```
<asp:gridview AllowSorting = "true" AllowPaging = "true" PageSize= "5"
ID = "GrideView1" runat= "server" dataKeynames= "pid" DataSourceID= "SqlIDS">
<Columns>
<asp:BoundField DataField= "pname" HeaderText= "Product Name" SortExpression=
"pname">
```

</asp:BoundField>

</Columns>

</asp:gridview>

Basic attributes of the GridView Control:

Attribute	Description
ID	The ID of the control.
Runat	Must Specify "server".
DataSourceID	The ID of the data source to bind to.
DataKeyNames	The names of the primary key fields separated by commas.
AutoGenerateRows	If true, a row is automatically generated for each field in the data source. If false, you must define the rows in the fields elements.
DefaultMode	Sets the initial mode of the DetailsView control. Valid options are Edit, Insert, or ReadOnly.
AllowPaging	Set true to allow paging.

Q4(f) Write short note on selecting a GridView Row.

(5)

Ans. The GridView control is used to display the values of a data source in a table. One of the key advantages of the GridView control over other data-bound controls is its ability to automatically take advantage of data source capabilities. In some situations we need to select an entire row's values for different purposes. The following lesson will explain how you can select an entire row from a GridView and retrieve the data from the GridView row.

Selecting the row event is fired when you make a click on the select link. If you need any particular item in that row you can easily select it using the cells property. In the GridView, double-click on the SelectedIndexChanged event and write the following code:

```
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    TextBoxUserID.Text = GridView1.SelectedRow.Cells[1].Text;
    TextBoxUserName.Text = GridView1.SelectedRow.Cells[2].Text; }
}
```

Q5. Attempt any three of the following.

[15]

Q5(a) Write short note on the XML Text Reader class.

(5)

Ans. provides forward-only, read-only access to a stream of XML data. The current node refers to the node on which the reader is positioned. The reader is advanced using any of the read methods and properties reflect the value of the current node. The syntax to declare an object of this class is as follows:

```
XmlTextReader reader = new XmlTextReader ("XML1.xml");
```

XmlTextReader provides the following functionality:

- Enforces the rules of well-formed XML.
- XmlTextReader does not provide data validation.
- Checks that DocumentType nodes are well-formed. XmlTextReader checks the DTD for well-formedness, but does not validate using the DTD.
- For nodes where NodeType is XmlNodeType.EntityReference, a single empty EntityReference node is returned (that is, the Value property is String.Empty).
- Does not expand default attributes.

Because the XmlTextReader does not perform the extra checks required for data validation, it provides a fast-well-formedness parser. To perform data validation, use a validating XmlReader. To read XML data from an XmlDocument, use XmlNodeReader.

XmlTextReader throws an XmlException on XML parse errors. After an exception is thrown the state of the reader is not predictable. For example, the reported node type may be different than the actual node type of the current node. Use the ReadState property to check whether a reader is in error state.

Q5(b) Explain the reading process from an XML Document with example.

(5)

Ans. XML information is read into memory from different formats. It can be read from a string, stream, URL, text reader, or a class derived from the XmlReader.

The Load method brings the document into memory and has overloaded methods available to take data from each of the different formats. There is also a LoadXml method that reads XML from a string.

Different Load methods affect which nodes are created when the XML Document Object Model (DOM) is loaded. The following table lists the differences between some of the Load methods and topics that address them.

Subject	Topic
Creation of white space nodes	The object used to load the DOM has an affect on the white space and significant white space nodes generated in the DOM. For more information, see White Space and Significant White Space Handling when Loading the DOM.
Loading XML starting from a specific node or loading the entire XML document	Using the XmlDocument.Load method data can be loaded from a specific node into the DOM. For more information, see Load Data from a Reader.
Validating the XML as it is loaded	The XML data loaded into the DOM can be validated as it is loaded. This is accomplished using a validating XmlReader. For more information about validating XML as it is loaded, see Validating an XML Document in the DOM.

```
using System;
```

```
using System.IO;
```

```
using System.Xml;
```

```
public class Sample
```

```
{
```

```
    public static void Main()
```

```
    {
```



```
// Create the XmlDocument.  
XmlDocument doc = new XmlDocument();  
doc.LoadXml("<book genre='novel' ISBN='1-861001-57-5'>" +  
    "<title>Pride And Prejudice</title>" +  
    "</book>");  
  
// Save the document to a file.  
doc.Save("data.xml");  
}  
  
}
```

Q5(c) Define Authentication and Authorization. Give types of Authentication. (5)

Ans. Authentication:

- In ASP.NET there are many different types of authentication procedures for web applications. If you want to specify your own authentication methods then that also is possible.
- The different authentication modes are accepted through settings that can be applied to the applications web.config file.
- The web.config file is XML-based file which allows changing of ASP.NET behavior easily.
- ASP.NET provides three different authentication provides:
 - Windows: It allows to authenticate user based on their windows accounts. This provider uses IIS to perform the authentication and then asses the authenticated identity to your code. This is the default provides for ASP.NET.
 - Forms: It uses custom HTML forms to collect authentication information and lets you use your own logic to authenticate users. The users identification are stored in a cookie for use during the session.
 - Passport: it uses Microsoft passport service to authenticate users.

Authorization:

- Authentication and Authorization are two interconnected security concepts.
- Authentication is a process of identifying a user and authorization is the process of checking whether authenticated user has access to the resources they requested.
- There are two forms of authorization available in ASP.NET:
 - FileAuthorization: it is performed by the FileAuthorization Module. It uses the access control list of the .aspx file to resolve whether a user should have access to the file. ACL permissions are confirmed for the users windows identity.
 - UrlAuthorization: In the Web.config file you can specify the authorization rules for various directories or files using the <authorization> element.

Q5(d) What is Windows Authentication? Give details.

(5)

Ans. Form Authentication is a wonderful approach, if you are implementing your own authentication process using a back-end database and a custom page. But if you are creating a web application for a limited number of users who are already part of a network domain then Windows Authentication is beneficial and the preferred choice for authentication. Windows-based authentication is manipulated between the Windows server and the client machine.

The ASP.NET applications resides in Internet Information Server (IIS). Any user's web request goes directly to the IIS server and it provides the authentication process in a Windows-based authentication model. This type of authentication is quite useful in an intranet environment in which users are asked to log into a network.

In this scenario, you can utilize the credentials that are already in place for the authentication and authorization process. This authentication is performed by IIS. It first accepts user's credentials from the domain login "Domain\UserName and Password". If this process fails then IIS displays an error and asks to re-enter the login information.

To configure Asp.NET to use windows authentication uses the following code:

```
<system.web>
```

<authentication mode = "windows"/>

<authorization>

<allow users = "">

</authentication>

</system.web>

<allow users = ""/>

Advantage of Windows Authentication:

- It relies and allows the user to use existing Windows Accounts.
- Establishes the foundation for Uniform Authentication model for multiple types of applications.
- From developer's point of view, easy to implement.

Disadvantage of Windows Authentication:

- Applicable to Microsoft platforms only.
- No custom control over this platform provided authentication process.

Q5(e) Explain AJAX with its advantages and Disadvantages. (5)

Ans. Asynchronous JavaScript and XML can be defined as a set of users in building websites and web applications. Identifying specific purpose in web development applications is the best way to understand Ajax. The main work of Ajax is to update the content asynchronously means in user's web page whole content need not be reloaded and only the required field is reloaded. XML is a markup language means these are coded languages to annotate parts of a web document which gives web browsers instructions about understanding and displaying the user content.

It combines various programming tools like JavaScript, HTML, DHTML, XML (extensible markup language), CSS (cascading style sheets), DOM (document object model) and Microsoft object.

Ajax was developed by Microsoft outlook web applications in 1999 but the complete usage of Ajax came into the picture after 6 years. The name Ajax was named by Jesse James Garrett in February 2005. Before the naming of Ajax, it is called an XML HttpRequest scripting object which runs as MSXML library.

The acronym of Ajax is Asynchronous JavaScript and XML. It can be defined as a path in which the user uses JavaScript to communicate with a web server to get a query generated response and to provide it on the webserver without the user leaving the current page.

Advantages:

1. Reduces the server traffic and increases the speed.
2. Ajax is responsive and time taken is also less.
3. Form validation.
4. Bandwidth usage can be reduced.
5. Asynchronous calls can be made this reduces the time for data arrival.

Disadvantages:

1. Open-source.
2. Active x request is created only in internet explorer and newly created web browser.
3. For security reason, you can only access information from the web host that serve pages. Fetching information from other servers is not possible with Ajax.

Q5(f) Give brief information on Accordion control with appropriate properties. (5)

Ans. The main Accordion features include:

- Unlimited number of hierarchy levels
(`AccordionControl.Elements`, `AccordionControlElement.Elements`).
- Items and groups can display images and text in their headers
(`AccordionControlElementBase.Image`, `AccordionControlElementBase.ImageIndex`, `AccordionControlElementBase.Text`)
- Items can display custom controls in an expandable area below their headers
(`AccordionControlElementBase.ContentContainer`).
- Load items' content dynamically
(`AccordionControl.HasContentContainer`, `AccordionControl.GetContentContainer`).
- One or multiple elements can be expanded simultaneously
(`AccordionControl.ExpandElementMode`)
- Context buttons in item and group headers
(`AccordionControl.ItemContextButtons`, `AccordionControl.GroupContextButtons` and `AccordionControl.ContextButtonCustomize`).
- Custom controls in item headers (`AccordionControlElementBase.HeaderControl`).
- Item selection feature (`AccordionControl.AllowItemSelection`) - Allows you to highlight a clicked item.
- Item and group expanding/collapsing animation (`AccordionControl.AnimationType`)
- The built-in search box (`AccordionControl.ShowFilterControl`) - Allows you to filter items based on the search text.
- Support for an external search box (`SearchControl`).

Gets or sets the default action description of the control for use by accessibility client applications. Inherited from `Control`.

Name	Description
AccessibilityObject	Gets the AccessibleObject assigned to the control. Inherited from Control.
AccessibleDefaultActionDescription	Gets or sets the default action description of the control for use by accessibility client applications. Inherited from Control.
AccessibleDescription	Gets or sets the description of the control used by accessibility client applications. Inherited from Control.
AccessibleName	Gets or sets the name of the control used by accessibility client applications. Inherited from Control.
AccessibleRole	Gets or sets the accessible role of the control. Inherited from Control.
ActiveGroup	Gets or sets the currently selected element.
AllowDrop	Gets or sets a value indicating whether the control can accept data that the user drags onto it. Inherited from Control.
AllowElementDragging	Gets or sets whether or not end-users should be able to drag this AccordionControl's elements at runtime.
AllowGlyphSkinning	Gets or sets whether or not the Glyph Skinning feature is enabled for this AccordionControl.
AllowHorizontalResizing	Gets or sets whether or not end-users can drag AccordionControl borders to resize this control horizontally.