# Que:- [What is the difference between SessionState and ViewState?](http://stackoverflow.com/questions/733482/what-is-the-difference-between-sessionstate-and-viewstate)

# **Session State** contains information that is pertaining to a specific session (by a particular client/browser/machine) with the server. It's a way to track what the user is doing on the site.. **across multiple pages**...amid the statelessness of the Web. e.g. the contents of a particular user's shopping cart is session data. Cookies can be used for session state. **View State** on the other hand is information specific to particular web page. It is stored in a hidden field so that it isn't visible to the user. It is used to maintain the user's illusion that the page remembers what he did on it the last time - dont give him a clean page every time he posts back.

* Session state is saved on the server, ViewState is saved in the page.
* Session state is usually cleared after a period of inactivity from the user (no request happened containing the session id in the request cookies).
* The view state is posted on subsequent post back in a hidden field.
* -session has expired ,view state has not expireddate
* If you're going to store information that you want to access on different web pages, you can use SessionState
* If you want to store information that you want to access from the same page, then you can use Viewstate
* **View State:**
* - View state is maintained in page level only.  
  - View state of one page is not visible in another page.  
  - View state information stored in client only.  
  - View state persist the values of particular page in the client (browser) when post back operation done.  
  - View state used to persist page-instance-specific data. ([Read more](http://www.aspdotnet-suresh.com/2012/11/viewstate-in-aspnet-with-example-in-c.html))

**Session State:**

- Session state is maintained in session level.  
- Session state value is available in all pages within a user session.  
- Session state information stored in server.  
- Session state persist the data of particular user in the server. This data available till user close the browser or session time completes.  
- Session state used to persist the user-specific data on the server side. ([Read more](http://www.aspdotnet-suresh.com/2012/11/aspnet-session-state-example-in-c-vbnet.html))

**Usage**

- If you want to access the information on different web pages, you can use SessionState

- If you want to access from the same page, then you can use Viewstate

**Security**

     Session state provides more security when compared with view state as the data value is stored in server side

Que:- What is the difference between String and StringBuffe

String is used to manipulate character strings that cannot be changed (read-only and immutable).

StringBuffer is used to represent characters that can be modified.

Performance wise, StringBuffer is faster when performing concatenations. This is because when you concatenate a String, you are creating a new object (internally) every time since String is immutable.

String..   
1.Its a class used to handle strings.   
2.Here concatenation is used to combine two strings.   
3.String object is used to concatenate two strings.   
4.The first string is combined to the other string by   
creating a new copy in the memory as a string object, and   
then the old   
string is deleted   
5.we say "Strings are immutable".   
6.When using concatenation consumes more memory.   
  
String Builder..   
1.This is also the class used to handle strings.   
2.Here Append method is used.   
3.Here, Stringbuilder object is used.   
4.Insertion is done on the existing string(no new memory is allocated)   
5.Usage of StringBuilder is more efficient in case large   
amounts of string manipulations have to be performed   
6.Consumes less memory when compared to String, if we add the characters to the string in future.

**String**

String is immutable. Immutable means once we create string object we cannot modify. Any operation like insert, replace or append happened to change string simply it will discard the old value and it will create new instance in memory to hold the new value.

**Example**

|  |
| --- |
| string str = "hi";  // create a new string instance instead of changing the old one  str += "test";  str += "help"; |

**String Builder**

String builder is mutable it means once we create string builder object we can perform any operation like insert, replace or append without creating new instance for every time.

**Example**

|  |
| --- |
| StringBuilder sb = new StringBuilder("");  sb.Append("hi");  sb.Append("test ");  string str = sb.ToString(); |

**Differences**

|  |  |
| --- | --- |
| **String** | **StringBuilder** |
| It’s an immutable | It’s mutable |
| Performance wise string is slow because every time it will create new instance | Performance wise stringbuilder is high because it will use same instance of object to perform any action |
| In string we don’t have append keyword | In StringBuilder we can use append keyword |
| String belongs to **System** namespace | Stringbuilder belongs to **System.Text** namespace |

**What is difference between constants, read-only and, static?**  
Constants: The value can’t be changed

Read-only: The value will be initialized only once from the constructor of the class.

Static: Value can be initialized once.

<http://www.c-sharpcorner.com/UploadFile/c210df/difference-between-const-readonly-and-static-readonly-in-C-Sharp/>

<http://www.dotnettricks.com/learn/csharp/difference-between-constant-and-readonly-and-static>

**What is difference between array and array list?**

| **Array** | **ArrayList** |
| --- | --- |
| Array is strongly typed. This means that an array can store only specific type of items\elements. | ArrayList can store any type of items\elements. |
| Array stores fixed number of elements. Size of an Array must be specified at the time of initialization. | ArrayList grows automatically and you don't need to specify size. |
| No need to cast elements of an array while retriving because it is strongly type and stores specific type of items only. | Items of ArrayList need to be cast to appropriate data type while retriving. |
| Use static helper class Array to perform different tasks on the array. | ArrayList itself includes various utility methods for various tasks. |

<http://www.aspdotnet-suresh.com/2013/09/difference-bw-array-and-arraylist-in-csharp-example.html>

|  |  |
| --- | --- |
| **Arrays** | **ArrayLists** |
| These are strong type collection and allow to store fixed length | Array Lists are not strong type collection and size will increase or decrease dynamically |
| In arrays we can store only one datatype either int, string, char etc… | In arraylist we can store all the datatype values |
| Arrays belong to System.Array namespace | Arraylist belongs to System.Collection namespaces |

**Que:**-[What is the difference between](http://stackoverflow.com/questions/733482/what-is-the-difference-between-sessionstate-and-viewstate) structure and union

|  |  |
| --- | --- |
| Structure | Union |
| 1.The keyword **struct** is used to define a structure | 1. The keyword union is used to define a union. |
| 2. When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is greater than or equal to the sum of sizes of its members. The smaller members may end with unused slack bytes. | 2. When a variable is associated with a union, the compiler allocates the memory by considering the size of the largest memory. So, size of union is equal to the size of largest member. |
| 3. Each member within a structure is assigned unique storage area of location. | 3. Memory allocated is shared by individual members of union. |
| 4. The address of each member will be in ascending order This indicates that memory for each member will start at different offset values. | 4. The address is same for all the members of a union. This indicates that every member begins at the same offset value. |
| 5 Altering the value of a member will not affect other members of the structure. | 5. Altering the value of any of the member will alter other member values. |
| 6. Individual member can be accessed at a time | 6. Only one member can be accessed at a time. |
| 7. Several members of a structure can initialize at once. | 7. Only the first member of a union can be initialized. |

Que:-What is the difference structure and class

In .NET, there are two categories of types, *reference types* and *value types*.

Structs are *value types* and classes are *reference types*.

The general difference is that a *reference type* lives on the heap, and a *value type* lives inline, that is, wherever it is your variable or field is defined.

A variable containing a *value type* contains the entire *value type* value. For a struct, that means that the variable contains the entire struct, with all its fields.

A variable containing a *reference type* contains a pointer, or a *reference* to somewhere else in memory where the actual value resides.

This has one benefit, to begin with:

* \*value type\*s always contains a value
* *reference types* can contain a *null*-reference, meaning that they don't refer to anything at all at the moment

Internally, \*reference type\*s are implemented as pointers, and knowing that, and knowing how variable assignment works, there are other behavioral patterns:

* copying the contents of a *value type* variable into another variable, copies the entire contents into the new variable, making the two distinct. In other words, after the copy, changes to one won't affect the other
* copying the contents of a *reference type* variable into another variable, copies the reference, which means you now have two references to the same *somewhere else* storage of the actual data. In other words, after the copy, changing the data in one reference will appear to affect the other as well, but only because you're really just looking at the same data both places

When you declare variables or fields, here's how the two types differ:

* variable: *value type* lives on the stack, *reference type* lives on the stack as a pointer to somewhere in heap memory where the actual memory lives
* class/struct-field: *value type* lives inside the class, *reference type* lives inside the class as a pointer to somewhere in heap memory where the actual memory lives.

A short summary of each:

**Classes Only:**

* Can support inheritance
* Are reference (pointer) types
* The reference can be null
* Have memory overhead per new instance

**Structs Only:**

* Cannot support inheritance
* Are value types
* Are passed by value (like integers)
* Cannot have a null reference (unless Nullable is used)
* Do not have a memory overhead per new instance - unless 'boxed'

**Both Classes and Structs:**

* Are compound data types typically used to contain a few variables that have some logical relationship
* Can contain methods and events
* Can support interfaces
* **In addition to all the above differences**  
  1. Structs cannot have explicit parameterless constructor where as a class can  
  2. Structs cannot have destructors, where as a class can  
  3. Struct can't inherit from another class where as a class can, Both structs and classes can inherit from an interface.
* If you are after a video explaining all the differences, you can check this link. <http://csharp-video-tutorials.blogspot.com/2012/06/part-29-c-tutorial-difference-between.html>

Besides the basic difference of access specifier, and few mentioned above I would like to add some of the major differences including few of the mentioned above with a code sample with output, which will give a more clear idea of the reference and value

**Structs:**

* Are value types and do not require heap allocation.
* Memory allocation is different and is stored in stack
* Useful for small data structures
* Affect performance, when we pass value to method, we pass the entire data structure and all is passed to the stack.
* Constructor simply returns the struct value itself (typically in a temporary location on the stack), and this value is then copied as necessary
* The variables each have their own copy of the data, and it is not possible for operations on one to affect the other.
* Do not support user-specified inheritance, and they implicitly inherit from type object

**Class:**

* Reference Type value
* Stored in Heap
* Store a reference to a dynamically allocated object
* Constructors are invoked with the new operator, but that does not allocate memory on the heap
* Multiple variables may have a reference to the same object
* It is possible for operations on one variable to affect the object referenced by the other variable

# Que: Difference between Server.Transfer and Response.Redirect

Server.Transfer() : client is shown as it is on the requesting page only, but the all the content is of the requested page. Data can be persist accros the pages using Context.Item collection, which is one of the best way to transfer data from one page to another keeping the page state alive.   
   
  
   
Response.Dedirect() :client know the physical loation (page name and query string as well). Context.Items loses the persisitance when nevigate to destination page. In earlier versions of IIS, if we wanted to send a user to a new Web page, the only option we had was Response.Redirect. While this method does accomplish our goal, it has several important drawbacks. The biggest problem is that this method causes each page to be treated as a separate transaction. Besides making it difficult to maintain your transactional integrity, Response.Redirect introduces some additional headaches. First, it prevents good encapsulation of code. Second, you lose access to all of the properties in the Request object. Sure, there are workarounds, but they’re difficult. Finally, Response.Redirect necessitates a round trip to the client, which, on high-volume sites, causes scalability problems.  
As you might suspect, Server.Transfer fixes all of these problems. It does this by performing the transfer on the server without requiring a roundtrip to the client.

## Response.Redirect

1. Response.Redirect() will send you to a new page, update the address bar and add it to the Browser History. On your browser you can click back.
2. It redirects the request to some plain HTML pages on our server or to some other web server.
3. It causes additional roundtrips to the server on each request.
4. It doesn’t preserve Query String and Form Variables from the original request.
5. It enables to see the new redirected URL where it is redirected in the browser (and be able to bookmark it if it’s necessary).
6. Response. Redirect simply sends a message down to the (HTTP 302) browser.

## Server.Transfer

1. Server.Transfer() does not change the address bar, we cannot hit back.One should use Server.Transfer() when he/she doesn’t want the user to see where he is going. Sometime on a "loading" type page.
2. It transfers current page request to another .aspx page on the same server.
3. It preserves server resources and avoids the unnecessary roundtrips to the server.
4. It preserves Query String and Form Variables (optionally).
5. It doesn’t show the real URL where it redirects the request in the users Web Browser.
6. Server.Transfer happens without the browser knowing anything, the browser request a page, but the server returns the content of another.

http://www.codeproject.com/Tips/724972/Difference-Between-Response-Redirect-and-Server-Tr

http://www.c-sharpcorner.com/UploadFile/3d39b4/difference-between-response-redirect-and-server-transfer/

# Que:-Different between Asp and asp.net

The most important difference between ASP and ASP.Net is that ASP uses interpreted VBScript or JScript, and ASP.net uses any .Net language (including VB.Net, C#, J#, etc.) *compiled*.

ASP 3.0 left all its code in the front of the application. There was no way for a programmer to "hide" the sensitive code which he or she may not want anybody to see. The fact that the code was interpreted also slowed performance. ASP.NET allows the programmer to create dynamic link libraries containing the sensitive code. This may be a disadvantage from an open-source perspective but compiling code into dll's greatly improves performance.

ASP.NET is firmly rooted in XML. Customarily, the dlls that ASP.NET creates start out as namespaces. All of the classes in the namespaces are then compiled into a single dll binary.

1. ASP is mostly written using VB Script and HTML intermixed. Presentation and business logic is intermixed while ASP.NET can be written in several .NET compliant languages such as C# or VB.NET.
2. ASP has maximum of 4 built in classes like Request, Response, Session and Application whereas ASP.NET using .NET framework classes which has more than 2000 in built classes.
3. ASP does not have any server based components whereas ASP.NET offers several server based components like Button, TextBox etc. and event driven processing can be done at server.
4. ASP doesn't support Page level transactions whereas ASP.NET supports Page level transactions.
5. ASP.NET offers web development for mobile devices which alters the content type (wml or chtml etc.) based on the device.
6. ASP.NET allows separation of business and presentation logic because the code need not be included directly in the \*.aspx page.
7. ASP.NET uses languages which are fully object oriented languages like C# and also supports cross language support.
8. ASP.NET offers support for Web Services and rich data structures like DataSet which allows disconnected data processing.

# Better language support   Programmable controls   Event-driven programming   XML-based components   User authentication, with accounts and roles   Higher scalability   Increased performance - Compiled code   Easier configuration and deployment   Not fully ASP compatible

Showing Answers 1 - 8 of 8 Answers

**Nirupama.A**

Answered On : Aug 30th, 2005

ASP .NET has   
   
   
Better language support    
Programmable controls    
Event-driven programming    
XML-based components    
User authentication, with accounts and roles    
Higher scalability    
Increased performance - Compiled code    
Easier configuration and deployment    
Not fully ASP compatible    
 

Answer Question

Yes  1 User has rated as useful.

[Login](http://www.geekinterview.com/login.html) to rate this answer.

**vittal**

Answered On : Sep 8th, 2005

[ASPhttp://images.intellitxt.com/ast/adTypes/icon1.png](http://www.geekinterview.com/question_details/15908).NET is compiled common language runtime code running on the server.While in ASP its interpreted by script engine.

Answer Question

[Login](http://www.geekinterview.com/login.html) to rate this answer.

**kumaraswamy**

Answered On : Oct 13th, 2005

There are a number of striking differences between ASP.NET and ASP. For some of these differences, the benefits will be immediately obvious. For others. Well have to get used to new ways of thinking about dynamic web Pages. Among the changes are:

**ASP.NET pages are complied, not interpreted**. A binary executable is compiled upon the first request to the page. This image is stored in an in-memory cache on the web server, and subsequent request to this page use this executable to service the request.

**The Framework provides a very clean separation of code from content**. With ASP, because the HTML is generated as the page is interpreted, your page logic must be embedded into the page at the location where you want the HTML generated by this logic to be output. With ASP.NET, no HTML is generated until all of the code in your page has finished executing. The entire task of HTML generation is done in the pages rendering step, which uses the properties for all of us who consider script writing and HTML generation a poor substitute to sitting down and writing real code.

**ASP.NET Framework maintains state for you.** Do you ever have to post of the server to apply validation logic to a data entry HTML form? When theres a problem, you must write code to repopulate every input on your HTML form. You must also execute script inline to add validation messages next to the fields that have errors. The resulting code is often a tangled mess. If the business logic changes, thats a tough page to maintain. With the ASP.NET Framework, this state maintenance is done for you. The fields maintain their value without a single line of code written by you. This applies not only to simple text inputs but also to SELECT lists, check boxes, radio buttons, and other input types on your form. Built-in validation controls allow you to enforce your business logic by adding a single tag to your page and simply checking the Page. Is Valid property when it posts to the server.

**ASP.NET runs events on the server**. In ASP, because of the amount of script that must be mixed with the HTML, its common to split a single functional area across several pages. One page may collect data from a user, whereas another accepts the HTTP post and updates your relational data, telling your user the result of the operation. Although its possible to put this functionality into a single ASP page you do so at the risk of needing to maintain a garbled mess of code over the long haul. Breaking these functions into separate pages causes the number of files in your web site to balloon. With ASP.NET, you can set up server side event traps

**ASP.NET provides a consistent event model**. With ASP, script is executed on the page in a top-down manner. Although its possible to put your script within functions that you call from the page body, theres no event model that fires at specific points in the lifecycle of your page. With ASP.NET, this event model has been added. Most importantly, theres an event fired whenever your page. With ASP.NET, this event your page begins to load. This is very much like the from\_Load event in VB. The page load event can be trapped in a script tag or from your code behind the page. This gives you a consistent model for setting up your output.

Answer Question

Yes  2 Users have rated as useful.

[Login](http://www.geekinterview.com/login.html) to rate this answer.

[**ramakrishnag1982**](http://www.geekinterview.com/user-profile/68705)

Answered On : Aug 12th, 2007

[View all questions by ramakrishnag1982](http://www.geekinterview.com/user_questions/68705)   [View all answers by ramakrishnag1982](http://www.geekinterview.com/user_answers/68705)

**ASP:**1) ASP is Interpreted language based on scripting languages like Jscript or   
VBScript.  
  
2) ASP has Mixed HTML and coding logic.  
  
3) Limited development and debugging tools available.  
  
4) Limited OOPS support.  
  
5) Limited session and application state management.  
  
6) Poor Error handling system.  
  
7) No in-built support for XML.  
  
8) No fully distributed data source support.

**ASP.Net:**1) ASP.Net is supported by compiler and has compiled language support.  
  
2) Separate code and design logic possible.  
  
3) Variety of compilers and tools available including the [Visual studiohttp://images.intellitxt.com/ast/adTypes/icon1.png](http://www.geekinterview.com/question_details/15908).Net.  
  
4) Completely Object Oriented.  
  
5) Complete session and application state management.  
  
6) Full proof error handling possible.  
  
7) Full XML Support for easy data exchange.  
  
8) Fully distributed data source support

**Que:-Difference between Abstract and Interface**

**Abstract Class:**  
-Abstract class provides a set of rules to implement next class  
-Rules will be provided through abstract methods  
-Abstract method does not contain any definition  
-While inheriting abstract class all abstract methods must be override  
-If a class contains at least one abstract method then it must be declared as an “Abstract Class”  
-Abstract classes cannot be instantiated (i.e. we cannot create objects), but a reference can be created  
-Reference depends on child class object’s memory  
-Abstract classes are also called as “Partial abstract classes”  
-Partial abstract class may contain functions with body and functions without body  
-If a class contains all functions without body then it is called as “Fully Abstract Class” (Interface)  
  
**Interface:**  
-If a class contains all abstract methods then that class is known as “Interface”  
-Interfaces support like multiple inheritance  
-In interface all methods r public abstract by default  
-Interfaces r implementable  
-Interfaces can be instantiated, but a reference cannot be created

| Feature | Interface | Abstract class |
| --- | --- | --- |
| Multiple inheritance | A class may inherit several interfaces. | A class may inherit only one abstract class. |
| Default implementation | An interface cannot provide any code, just the signature. | An abstract class can provide complete, default code and/or just the details that have to be overridden. |
| Access Modfiers | An interface cannot have access modifiers for the subs, functions, properties etc everything is assumed as public | An abstract class can contain access modifiers for the subs, functions, properties |
| Core VS Peripheral | Interfaces are used to define the peripheral abilities of a class. In other words both Human and Vehicle can inherit from a IMovable interface. | An abstract class defines the core identity of a class and there it is used for objects of the same type. |
| Homogeneity | If various implementations only share method signatures then it is better to use Interfaces. | If various implementations are of the same kind and use common behaviour or status then abstract class is better to use. |
| Speed | Requires more time to find the actual method in the corresponding classes. | Fast |
| Adding functionality (Versioning) | If we add a new method to an Interface then we have to track down all the implementations of the interface and define implementation for the new method. | If we add a new method to an abstract class then we have the option of providing default implementation and therefore all the existing code might work properly. |
| Fields and Constants | No fields can be defined in interfaces | An abstract class can have fields and constrants defined |

# Que :- between String and string in c#?

The main difference is system.string is immutable and system.stringbuilder is a mutable. Append keyword is used in string builder but not in system.string.

Immutable means once we created we cannot modified. Suppose if we want give new value to old value simply it will discarded the old value and it will create new instance in memory to hold the new value.

**Que:-What are differences between Array list and Hash table?**

**Ans:** 1) Hash table store data as name, value pair. While in array only value is store.

2) To access value from hash table, you need to pass name. While in array, to access value, you need to pass index number.

3) you can store different type of data in hash table, say int, string etc. while in array you can store only similar type of data.

**Que:-What are the differences between Application object and session object?**

**Ans:** The session object is used to maintain the session of each user. If one user enter in to the application then they get session id if he leaves from the application then the session id is deleted. If they again enter in to the application they get different session id.  
But for application object the id is maintained for whole application.

**Que:-What are the different types of indexes?**

**Ans:** Two types of indexes are there one is clustered index and non-clustered index

**Que:-Different between Data set and Data Reader?**

**DataReader**

DataReader is used to read the data from database and it is a read and forward only connection oriented architecture during fetch the data from database. DataReader is used to iterate through resultset that came from server and it will read one record at a time because of that memory consumption will be less and it will fetch the data very fast when compared with dataset. Generally we will use [ExecuteReader](http://www.aspdotnet-suresh.com/2012/09/sqlcommand-executereader-example-in.html) object to bind data to datareader.

**DataSet**

DataSet is a disconnected orient architecture that means there is no need of active connections during work with datasets and it is a collection of DataTables and relations between tables. It is used to hold multiple tables with data. You can select data form tables, create views based on table and ask child rows over relations. Also DataSet provides you with rich features like saving data as XML and loading XML data.

**DataAdapter**

DataAdapter will acts as a Bridge between DataSet and database. This dataadapter object is used to read the data from database and bind that data to dataset. Dataadapter is a disconnected oriented architecture. Check below sample code to see how to use DataAdapter in code

|  |  |
| --- | --- |
| DataSet object | DataReader object |
| Read/Write access | Read-only access |
| Supports multiple tables from different databases | Supports a single table based on a single SQL query of one database |
| Disconnected mode | Connected mode |
| Bind to multiple controls | Bind to a single control |
| Forward and backward scanning of data | Forward-only scanning of data |
| Slower access to data | Faster access to data |
| Greater overhead to enable additional features | Lightweight object with very little overhead |
| Supported by Visual Studio .NET tools | Must be manually coded |

### Difference between dataset and datareader.

**Dataset**

DataSet object can contain multiple rowsets from the same data source as well as from the relationships between them

Dataset is a disconnected architecture

Dataset can persist data.

**Datareader**

DataReader provides forward-only and read-only access to data.

Datareader is connected architecture

Datareader can not persist data.

##### Dataset and datareader in ADO.NET - June 06, 2009 at 10:00 AM by Shuchi Gauri

#### Difference between dataset and datareader.

**Dataset**

a. Disconnected  
b. Can traverse data in any order front, back.   
c. Data can be manipulated within the dataset.  
d. More expensive than datareader as it stores multiple rows at the same time.

**Datareader**

a. Connection needs to be maintained all the time  
b. Can traverse only forward.   
c. It is read only therefore, data cannot be manipulated.  
d. It is less costly because it stores one row at a time

### Difference between dataset and datareader.

**Dataset**

DataSet object can contain multiple rowsets from the same data source as well as from the relationships between them

Dataset is a disconnected architecture

Dataset can persist data.

**Datareader**

DataReader provides forward-only and read-only access to data.

Datareader is connected architecture

Datareader can not persist data.

##### Dataset and datareader in ADO.NET - June 06, 2009 at 10:00 AM by Shuchi Gauri

#### Difference between dataset and datareader.

**Dataset**

a. Disconnected  
b. Can traverse data in any order front, back.   
c. Data can be manipulated within the dataset.  
d. More expensive than datareader as it stores multiple rows at the same time.

**Datareader**

a. Connection needs to be maintained all the time  
b. Can traverse only forward.   
c. It is read only therefore, data cannot be manipulated.  
d. It is less costly because it stores one row at a time

### **Que:-[C#](http://www.aspdotnet-suresh.com/2012/10/c-difference-between-aspnet-webservice.html)** [-](http://www.aspdotnet-suresh.com/2012/10/c-difference-between-aspnet-webservice.html) **[Difference between Webservice and WCF in Asp.net](http://www.aspdotnet-suresh.com/2012/10/c-difference-between-aspnet-webservice.html)**

**What is Web Service?**

Web Service is an application that is designed to interact directly with other applications over the internet. In simple sense, Web Services are means for interacting with objects over the Internet. The Web serivce consumers are able to invoke method calls on remote objects by using SOAP and HTTP over the Web. WebService is language independent and Web Services communicate by using standard web protocols and data formats, such as HTTP, XML and SOAP. ([Read More](http://www.aspdotnet-suresh.com/2011/05/aspnet-web-service-or-creating-and.html))

**What is WCF (windows communication foundation) Service?**

Windows Communication Foundation (Code named Indigo) is a programming platform and runtime system for building, configuring and deploying network-distributed services. It is the latest service oriented technology; Interoperability is the fundamental characteristics of WCF. It is unified programming model provided in .Net Framework 3.0. WCF is a combined feature of Web Service, Remoting, MSMQ and COM+. WCF provides a common platform for all .NET communication. ([Read More](http://www.aspdotnet-suresh.com/2011/06/introduction-to-wcf-wcf-tutorial-wcf.html))

**Difference between WCF and Web service**

Web service is a part of WCF. WCF offers much more flexibility and portability to develop a service when comparing to web service. Still we are having more advantages over Web service; following table provides detailed difference between them.

|  |  |  |
| --- | --- | --- |
| **Features** | **Web Service** | **WCF** |
| Hosting | It can be hosted in IIS | It can be hosted in IIS, windows activation service, Self-hosting, Windows service |
| Programming | [WebService] attribute has to be added to the class | [ServiceContract] attribute has to be added to the class |
| Model | [WebMethod] attribute represents the method exposed to client | [OperationContract] attribute represents the method exposed to client |
| Operation | One-way, Request- Response are the different operations supported in web service | One-Way, Request-Response, Duplex are different type of operations supported in WCF |
| XML | System.Xml.serialization name space is used for serialization | System.Runtime.Serialization namespace is used for serialization |
| Encoding | XML 1.0, MTOM(Message Transmission Optimization Mechanism), DIME, Custom | XML 1.0, MTOM, Binary, Custom |
| Transports | Can be accessed through HTTP, TCP, Custom | Can be accessed through HTTP, TCP, Named pipes, MSMQ,P2P, Custom |
| Protocols | Security | Security, Reliable messaging, Transactions |

**Que:-Advantage of .net?**

1. [ASP.NET](http://www.seekdotnet.com/aspnet4hosting.aspx) dramatically reduces the amount of code needed to build large applications.

2. With built-in Windows authentication and application settings, and applications safe and secure.

3. It provides higher performance by using early binding, just-in-time compilation, native optimization, and caching services right out of the box.

4. ASP.NET framework is complemented by a rich and designer tools in Visual Studio integrated development environment. WYSIWYG-editing, drag and drop controls, firewall and automatic deployment are just some of the features of this powerful tool provides.

5. ASP.NET provides a simple and easy to perform common tasks, from simple form submission and client authentication configuration and deployment site.

6. The source code and HTML together, so that ASP.NET pages are easy to maintain and write. In addition, the source code is executed on the server. It provides greater power and flexibility to Web pages.

7. All processes are carefully controlled and managed by ASP.NET, so that if the process is dead, the new process can be created in its place, which helps to keep your application constantly available to handle requests.

8. This is purely server-side technologies, so that the ASP.NET code runs on the server before being sent to the browser.

9. Be independent of the language, it allows you to select the language that best applies to your application or partition applications in many languages.

10. ASP.NET makes for easy deployment. No need to register components because the configuration information is embedded.

11. The Web server continuously monitors the pages, components and applications running on it. If he notices any memory leaks, infinite loops, other illegal activities, which immediately destroys the activity and restarts.

12. It is easy to work with ADO.NET using data binding and formatting of the page. This is an application that works faster and counters large volumes of users without performance problems

**Que:-What is AJAX?**

AJAX stands for Asynchronous JavaScript and XML. Ajax is not a new programming language but its new way to use existing standards. Ajax is used to transfer data between the server and client without reloading whole page. In Ajax only necessary data is transferred back and forth between the client and web server that minimizes the network utilization and processing on the client. Ajax contains a rich set of controls that we can use to build highly responsive and interactive Ajax-enabled ASP.NET Web Forms applications. Gmail is the best example for Ajax enabled site and especially Google have made Ajax very popular.

**Advantages of Ajax**

**Asynchronous calls** — [AJAX](http://www.aspdotnet-suresh.com/search/label/Ajax) allows for the ability to make asynchronous calls to a web server. This allows the client browser to avoid waiting for all data to arrive before allowing the user to act once more.

**Minimal data transfer** — by not performing a full postback and sending all form data to the server, network utilization is minimized and quicker operations occur. In sites and locations with restricted pipes for data transfer, this can greatly improve network performance.

**Limited processing on the server** — Along with the fact that only the necessary data is sent to the server, the server is not required to process all form elements. By sending only the necessary data, there is limited processing on the server. There is no need to process all form elements, process the ViewState, send images back to the client, or send a full page back to the client.

**Responsiveness**— because [AJAX](http://www.aspdotnet-suresh.com/search/label/Ajax) applications are asynchronous on the client, they are perceived to be very responsive.

**Context** — with full postback, users may lose the context of where they are. Users may be at the bottom of a page, hit the Submit button, and be redirected back to the top of the page. With [AJAX](http://www.aspdotnet-suresh.com/search/label/Ajax) there is no full postback. Clicking the Submit button in an application that uses AJAX will allow users to maintain their location. The user state is maintained, and the users are no longer required to scroll down to the location they were at before clicking Submit.

**How Install Ajax in visual studio?**

If we want to use [Ajax](http://www.aspdotnet-suresh.com/search/label/Ajax) features in our application just follow the below steps to download and start using the Ajax Control Toolkit with Visual Studio:

1) First step is to download the latest version of the Ajax Control Toolkit from [CodePlex](http://ajaxcontroltoolkit.codeplex.com/) whichever suits to your visual studio. If you are using Visual Studio 2008 then you should pick the version of the Ajax Control Toolkit for .NET 3.5. If you are using Visual Studio 2010 then you can use either the .NET 4 or .NET 3.5 versions of the Ajax Control Toolkit.

**Que:- Delegates in c#**

[**http://www.c-sharpcorner.com/UploadFile/puranindia/C-Sharp-net-delegates-and-events/**](http://www.c-sharpcorner.com/UploadFile/puranindia/C-Sharp-net-delegates-and-events/)

[**http://www.tutorialsteacher.com/csharp/csharp-delegates**](http://www.tutorialsteacher.com/csharp/csharp-delegates)

**--for More Details**

### [C# - What is Delegates in C# Example | Use of Delegates in C#](http://www.aspdotnet-suresh.com/2013/09/C-Sharp-delegates-example-use-of-delegates-in-C-Sharp.html)

http://www.aspdotnet-suresh.com/2013/09/C-Sharp-delegates-example-use-of-delegates-in-C-Sharp.html

**Description:**

In previous posts I explained [OOPS examples in c#](http://www.aspdotnet-suresh.com/2010/04/introduction-to-object-oriented.html), [polymorphism in c#](http://www.aspdotnet-suresh.com/2013/09/polymorphism-in-c-with-example-types-of-polymorphism.html), [Difference b/w datareader, dataset and dataadapter in c#](http://www.aspdotnet-suresh.com/2012/10/aspnet-difference-between-datareader.html), [Difference b/w appsettings and connection strings in asp.net](http://www.aspdotnet-suresh.com/2012/10/aspnet-difference-between-appsettings.html), [Difference b/w executereader, executenonquery and executescalar in c#](http://www.aspdotnet-suresh.com/2012/09/differences-between-executereader.html) and many articles relating to [interview questions](http://www.aspdotnet-suresh.com/search/label/Interview%20Questions) in [c#](http://www.aspdotnet-suresh.com/search/label/C%23.Net), [asp.net](http://www.aspdotnet-suresh.com/search/label/Asp.net), [sql server](http://www.aspdotnet-suresh.com/search/label/SQL%20Server), [javascript](http://www.aspdotnet-suresh.com/search/label/Javascript), [jquery](http://www.aspdotnet-suresh.com/search/label/JQuery). Now I will explain delegates in [c#.net](http://www.aspdotnet-suresh.com/search/label/C%23.Net) with example.

Whenever we want to create delegate methods we need to declare with **delegate** keyword and delegate methods signature should match exactly with the methods which we are going to hold like same return types and same parameters otherwise delegate functionality won’t work if signature not match with methods.

**Syntax of Delegate & Methods Declaration**

Check below sample code for delegate declaration and methods declaration

|  |
| --- |
| public delegate int Delegatmethod(int a,int b);  public class Sampleclass  {  public int Add(int x, int y)  {  return x + y;  }  public int Sub(int x, int y)  {  return x + y;  }  } |

If you observe above code I declared Delegatmethod method with two parameters which matching with methods declared in Sampleclass class.

**Complete Example**

|  |
| --- |
| public delegate int DelegatSample(int a,int b);  public class Sampleclass  {  public int Add(int x, int y)  {  return x + y;  }  public int Sub(int x, int y)  {  return x - y;  }  }  class Program  {  static void Main(string[] args)  {  Sampleclass sc=new Sampleclass();  DelegatSample delgate1 = sc.Add;  int i = delgate1(10, 20);  Console.WriteLine(i);  DelegatSample delgate2 = sc.Sub;  int j = delgate2(20, 10);  Console.WriteLine(j);  }  } |

**Output**

Whenever we run above code we will get output like as shown below

|  |
| --- |
| **Add Result : 30**  **Sub Result : 10** |

**What is the use of Delegates?**

Suppose if you have multiple methods with same signature (return type & number of parameters) and want to call all the methods with single object then we can go for delegates.

Delegates are two types

      -   Single Cast Delegates

      -  Multi Cast Delegates

**Single Cast Delegates**

Single cast delegate means which hold address of single method like as explained in above example.

**Multicast Delegates**

Multi cast delegate is used to hold address of multiple methods in single delegate. To hold multiple addresses with delegate we will use overloaded += operator and if you want remove addresses from delegate we need to use overloaded operator -=

Multicast delegates will work only for the methods which have return type only void. If we want to create a multicast delegate with return type we will get the return type of last method in the invocation list

Check below sample code for delegate declaration and methods declaration

**Syntax of Multicast Delegate & Method Declaration**

Check below sample code for multicast delegate declaration and methods declaration

|  |
| --- |
| public delegate void MultiDelegate(int a,int b);  public class Sampleclass  {  public static void Add(int x, int y)  {  Console.WriteLine("Addition Value: "+(x + y));  }  public static void Sub(int x, int y)  {  Console.WriteLine("Subtraction Value: " + (x - y));  }  public static void Mul(int x, int y)  {  Console.WriteLine("Multiply Value: " + (x \* y));  }  } |

If you observe above code I declared MultiDelegate method with void return type.

**Complete Example**

|  |
| --- |
| public delegate void MultiDelegate(int a,int b);  public class Sampleclass  {  public static void Add(int x, int y)  {  Console.WriteLine("Addition Value: "+(x + y));  }  public static void Sub(int x, int y)  {  Console.WriteLine("Subtraction Value: " + (x - y));  }  public static void Mul(int x, int y)  {  Console.WriteLine("Multiply Value: " + (x \* y));  }  }  class Program  {  static void Main(string[] args)  {  Sampleclass sc=new Sampleclass();  MultiDelegate del = Sampleclass.Add;  del += Sampleclass.Sub;  del += Sampleclass.Mul;  del(10, 5);  Console.ReadLine();  }  } |

**Output**

Whenever we run above code we will get output like as shown below

|  |
| --- |
| **Addition Value : 15**  **Subtraction Value : 5**  **Multiply Value : 50** |

**Que:-Different between Appsetting and Conncection String**

**AppSettings in web.config**

The AppSettings section in web.config is used to store connection strings, server names, file paths, and other miscellaneous settings needed by an application. ([Read More](http://www.aspdotnet-suresh.com/2011/11/write-connection-strings-in-webconfig.html))

**Connection String in web.config**

The connection string in web.config is a collection of database connection strings only.

Actually in previous versions of ASP.NET all connection string values are storing in appsettings only. In ASP.NET 2.0 + version new features has introduced such as Session, Membership, Personalization, and Role Manager these will depend on connection strings and stored in the connectionStrings element only. ([Read More](http://www.aspdotnet-suresh.com/2011/11/write-connection-strings-in-webconfig.html))

**Difference**

The main difference is in **appsettings** section we can store any data string values including database connection strings also but in **connectionStrings** section only database connection strings can store those are our application connection strings and new features (Membership, Personalization and Role Manager) connection strings only.

Instead of this there is no much difference between appsettings and connectionStrings.

**Que:-Type Of polymorphicm in c#**

**Polymorphism**

Polymorphism means many forms (ability to take more than one form). In Polymorphism poly means “multiple” and morph means “forms” so polymorphism means many forms.

In polymorphism we will declare methods with same name and different parameters in same class or methods with same name and same parameters in different classes. Polymorphism has ability to provide different implementation of methods that are implemented with same name.

In Polymorphism we have 2 different types those are

        -   **Compile Time Polymorphism** (Called as Early Binding or Overloading or static binding)

        -   **Run Time Polymorphism** (Called as Late Binding or Overriding or dynamic binding)

**Compile Time Polymorphism**

Compile time polymorphism means we will declare methods with same name but different signatures because of this we will perform different tasks with same method name. This compile time polymorphism also called as **early binding** or **method overloading**.

Method Overloading or compile time polymorphism means same method names with different signatures (different parameters)

**Example**

|  |
| --- |
| public class Class1  {  public void NumbersAdd(int a, int b)  {  Console.WriteLine(a + b);  }  public void NumbersAdd(int a, int b, int c)  {  Console.WriteLine(a + b + c);  }  } |

In above class we have two methods with same name but having different input parameters this is called method overloading or compile time polymorphism or early binding.

**Run Time Polymorphism**

Run time polymorphism also called as **late binding** or **method overriding** or **dynamic polymorphism**. Run time polymorphism or method overriding means same method names with same signatures.

In this run time polymorphism or method overriding we can override a method in base class by creating similar function in derived class this can be achieved by using inheritance principle and using “**virtual** & **override**” keywords.

In base class if we declare methods with **virtual** keyword then only we can override those methods in derived class using **override** keyword

**Example**

|  |
| --- |
| //Base Class  public class Bclass  {  public virtual void Sample1()  {  Console.WriteLine("Base Class");  }  }  // Derived Class  public class DClass : Bclass  {  public override void Sample1()  {  Console.WriteLine("Derived Class");  }  }  // Using base and derived class  class Program  {  static void Main(string[] args)  {  // calling the overriden method  DClass objDc = new DClass();  objDc.Sample1();  // calling the base class method  Bclass objBc = new DClass();  objBc.Sample1();  }  } |

If we run above code we will get output like as shown below

**Output**

|  |
| --- |
| ----------------------------------  Derived Class  Derived Class |

# Que:- [What is](http://stackoverflow.com/questions/733482/what-is-the-difference-between-sessionstate-and-viewstate) different difference between compile time polymorphism and runtime polymorphism

**Polymorphism**

Polymorphism means many forms (ability to take more than one form). In Polymorphism poly means “multiple” and morph means “forms” so polymorphism means many forms.

In polymorphism we will declare methods with same name and different parameters in same class or methods with same name and same parameters in different classes. Polymorphism has ability to provide different implementation of methods that are implemented with same name.

In Polymorphism we have 2 different types those are

     -  **Compile Time Polymorphism** (Called as Early Binding or Overloading or static binding)

     -  **Run Time Polymorphism** (Called as Late Binding or Overriding or dynamic binding)

**Compile Time Polymorphism**

Compile time polymorphism means we will declare methods with same name but different signatures because of this we will perform different tasks with same method name. This compile time polymorphism also called as **early binding** or **method overloading**.

Method Overloading or compile time polymorphism means same method names with different signatures (different parameters)

For more details check this link [polymorphism in c#](http://www.aspdotnet-suresh.com/2013/09/polymorphism-in-c-with-example-types-of-polymorphism.html)

**Run Time Polymorphism**

Run time polymorphism also called as **late binding** or **method overriding** or **dynamic polymorphism**. Run time polymorphism or method overriding means same method names with same signatures.

In this run time polymorphism or method overriding we can override a method in base class by creating similar function in derived class this can be achieved by using inheritance principle and using “**virtual** & **override**” keywords.

# Que:- [What is](http://stackoverflow.com/questions/733482/what-is-the-difference-between-sessionstate-and-viewstate) different between overloading and overriding

**Polymorphism**

Polymorphism means many forms (ability to take more than one form). In Polymorphism poly means “multiple” and morph means “forms” so polymorphism means many forms.

In polymorphism we will declare methods with same name and different parameters in same class or methods with same name and same parameters in different classes. Polymorphism has ability to provide different implementation of methods that are implemented with same name.

In Polymorphism we have 2 different types those are

       -  **Overloading** (Called as Early Binding or Compile Time Polymorphism or static binding)

       -  **Overriding** (Called as Late Binding or Run Time Polymorphism or dynamic binding)

**Overloading**

Overloading means we will declare methods with same name but different signatures because of this we will perform different tasks with same method name. This overloading also called as **compile time polymorphism** or **early binding**.

Method Overloading or compile time polymorphism means same method names with different signatures (different parameters)

For more details check this link [polymorphism in c#](http://www.aspdotnet-suresh.com/2013/09/polymorphism-in-c-with-example-types-of-polymorphism.html)

**Overriding**

Overriding also called as **run time polymorphism** or **late binding** or **dynamic polymorphism**. Method overriding or run time polymorphism means same method names with same signatures.

In this method overriding or run time polymorphism we can override a method in base class by creating similar function in derived class this can be achieved by using inheritance principle and using “**virtual** & **override**” keywords.

# Que:- [What is](http://stackoverflow.com/questions/733482/what-is-the-difference-between-sessionstate-and-viewstate) different between Ref and Out in c#

**Description:**

In previous posts I explained [difference between wcf and web application](http://www.aspdotnet-suresh.com/2012/10/c-difference-between-aspnet-webservice.html), [interview questions in asp.net, sql server, c#](http://www.aspdotnet-suresh.com/2010/05/interview-questions-in-aspnetcnetsql.html), [Difference between functions and stored procedures in SQL Server](http://www.aspdotnet-suresh.com/2012/10/sql-server-functions-vs-stored.html), [difference between len and datalength in sql server](http://www.aspdotnet-suresh.com/2012/09/difference-between-len-and-datalength.html) and many articles relating to [interview questions](http://www.aspdotnet-suresh.com/search/label/Interview%20Questions). Now I will explain difference between ref and out parameters in [c#.net](http://www.aspdotnet-suresh.com/search/label/C%23.Net) with example.

**Ref Parameter**

If you want to pass a variable as ref parameter you need to initialize it before you pass it as ref parameter to method. Ref keyword will pass parameter as a reference this means when the value of parameter is changed in called method it get reflected in calling method also.

**Declaration of Ref Parameter**

Generally we will use ref parameters like as shown below

|  |
| --- |
| int i=3; // variable need to be initialized  Refsample(ref i); |

If you observe above code first we declared variable and initialized with value 3 before it pass a ref parameter to **Refsample** method

**Example**

|  |
| --- |
| class Program  {  static void Main()  {  int i; // variable need to be initialized  i = 3;  Refsample(ref i);  Console.WriteLine(i);  }  public static void Refsample(ref int val1)  {  val1 += 10;  }  } |

When we run above code we will get like as shown below

**Output**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\*  **13**  \*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\* |

As we discussed if ref parameter value changed in called method that parameter value reflected in calling method also

**Out Parameter**

If you want to pass a variable as out parameter you don’t need to initialize it before you pass it as out parameter to method. Out keyword also will pass parameter as a reference but here out parameter must be initialized in called method before it return value to calling method.

**Declaration of Out Parameter**

Generally we will use out parameters like as shown below

|  |
| --- |
| int i,j; // No need to initialize variable  Outsample(out i, out j); |

If you observe above code first we declared variable and we it pass a out parameter to **Outsample** method without initialize the values to variables

**Example**

|  |
| --- |
| class Program  {  static void Main()  {  int i,j; // No need to initialize variable  Outsample(out i, out j);  Console.WriteLine(i);  Console.WriteLine(j);  }  public static int Outsample(out int val1, out int val2)  {  val1 = 5;  val2 = 10;  return 0;  }  } |

If we observe code we implemented as per our discussion like out parameter values must be initialized in called method before it return values to calling method

When we run above code we will get like as shown below

**Output**

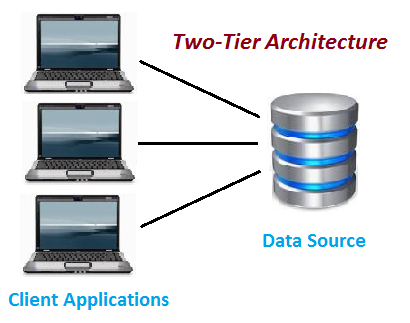
|  |
| --- |
| \*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\*  **5**  **10**  \*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\* |

# Que:- [What is](http://stackoverflow.com/questions/733482/what-is-the-difference-between-sessionstate-and-viewstate) different between 1 2 3 tier?

**Difference between 1-tier/2-tier & 3-tier architecture are as follows:**   
"Tier" can be defined as "one of two or more rows, levels, or ranks arranged one above another".  
   
1-Tier Architecture is the simplest, single tier on single user, and is the equivalent of running an application on a personal computer. All the required component to run the application are located within it. User interface, business logic, and data storage are all located on the same machine. They are the easiest to design, but the least scalable. Because they are not part of a network, they are useless for designing web applications.   
   
2-Tier Architectures supply a basic network between a client and a server. For example, the basic web model is a 2-Tier Architecture. A web browser makes a request from a web server, which then processes the request and returns the desired response, in this case, web pages. This approach improves scalability and divides the user interface from the data layers. However, it does not divide application layers so they can be utilized separately. This makes them difficult to update and not specialized. The entire application must be updated because layers aren’t separated.  
  
3-Tier Architecture is most commonly used to build web applications. In this model, the browser acts like a client, middleware or an application server contains the business logic, and database servers handle data functions. This approach separates business logic from display and data.So the 3 layers commonly known as:Presentation Layer(PL/UI),Business Logic Layer(BLL) & Data Access Layer(DAL).

# Two-Tier Architecture:

The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server. Because of tight coupling a 2 tiered application will run faster.

[](http://www.softwaretestingclass.com/what-is-difference-between-two-tier-and-three-tier-architecture/two-tier-architecture/)

Two-Tier Architecture

The above figure shows the architecture of two-tier. Here the direct communication between client and server, there is no intermediate between client and server.

Let’s take a look of real life example of Railway Reservation two-tier architecture:

Let’s consider that first Person is making Railway Reservation for Mumbai to Delhi by Mumbai Express at Counter No. 1 and at same time second Person is also try to make Railway reservation of Mumbai to Delhi from Counter No. 2

If staff from Counter No. 1 is searching for availability into system & at the same staff from Counter No. 2 is also looking for availability of ticket for same day then in this case there is might be good change of confusion and chaos occurs. There might be chance of lock the Railway reservation that reserves the first.

But reservations can be making anywhere from the India, then how it is handled?

So here if there is difference of micro seconds for making reservation by staff from Counter No. 1 & 2 then second request is added into queue. So in this case the Staff is entering data to Client Application and reservation request is sent to the database. The database sends back the information/data to the client.

In this application the Staff user is an end user who is using Railway reservation application software. He gives inputs to the application software and it sends requests to Server. So here both Database and Server are incorporated with each other, so this technology is called as “***Client-Server Technology***“.

The Two-tier architecture is divided into two parts:

**1) Client Application (Client Tier)  
2) Database (Data Tier)**

On client application side the code is written for saving the data in the SQL server database. Client sends the request to server and it process the request & send back with data. The main problem of two tier architecture is the server cannot respond multiple request same time, as a result it cause a data integrity issue.

**Advantages:**

1. Easy to maintain and modification is bit easy
2. Communication is faster

**Disadvantages**:

1. In two tier architecture application performance will be degrade upon increasing the users.
2. Cost-ineffective

# Three-Tier Architecture:

**Three-tier architecture** typically comprise a presentation tier, a business or data access tier, and a data tier. Three layers in the three tier architecture are as follows:

**1) Client layer**  
**2) Business layer**  
**3) Data layer**

**1) Client layer:**

It is also called as *Presentation layer* which contains UI part of our application. This layer is used for the design purpose where data is presented to the user or input is taken from the user. For example designing registration form which contains text box, label, button etc.

**2) Business layer:**

In this layer all business logic written like validation of data, calculations, data insertion etc. This acts as a interface between Client layer and Data Access Layer. This layer is also called the intermediary layer helps to make communication faster between client and data layer.

**3) Data layer:**

In this layer actual database is comes in the picture. Data Access Layer contains methods to connect with database and to perform insert, update, delete, get data from database based on our input data.

[](http://www.softwaretestingclass.com/what-is-difference-between-two-tier-and-three-tier-architecture/three-tier-architecture/)

Three-tier Architecture

**Advantages**

1. High performance, lightweight persistent objects
2. Scalability – Each tier can scale horizontally
3. Performance – Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers.
4. High degree of flexibility in deployment platform and configuration
5. Better Re-use
6. Improve Data Integrity
7. Improved Security – Client is not direct access to database.
8. Easy to maintain and modification is bit easy, won’t affect other modules
9. In three tier architecture application performance is good.

**Disadvantages**

1. Increase Complexity/Effort

# Que:- [What is](http://stackoverflow.com/questions/733482/what-is-the-difference-between-sessionstate-and-viewstate) ADO.NET? different between ADO and ADO.NET ?

# ActiveX Data Object.NET (ADO.NET)

## Definition - What does ActiveX Data Object.NET (ADO.NET) mean?

ActiveX Data Object.NET (ADO.NET) is a software library in the .NET framework consisting of software components providing data access services. ADO.NET is designed to enable developers to write managed code for obtaining disconnected access to data sources, which can be relational or non-relational (such as XML or application data). This feature of ADO.NET helps to create data-sharing, distributed applications.  
  
ADO.NET provides connected access to a database connection using the .NET-managed providers and disconnected access using datasets, which are applications using the database connection only during retrieval of data or for data update. Dataset is the component helping to store the persistent data in memory to provide disconnected access for using the database resource efficiently and with better scalability.

## Techopedia explains ActiveX Data Object.NET (ADO.NET)

ADO.NET evolved from ADO, which is also a technology similar to ADO.NET with a few basic structural changes. Although there is a provision to work in disconnected mode using ADO, data is transmitted to the database in ADO.NET more efficiently using data adapters. The in-memory representation of data differs between ADO and ADO.NET. ADO.NET can hold the data in a single result table, but ADO holds multiple tables along with their relationship details. Unlike ADO, data transmission between applications using ADO.NET does not use COM (component object model) marshalling but uses dataset, which transmits data as an XML stream.  
  
The architecture of ADO.NET is based on two primary elements: DataSet and .NET framework data provider.   
  
Dataset provides the following components:

1. a complete set of data including related tables, constraints and their relationships
2. functionality-like access to remote data from XML Web service
3. manipulation of data dynamically
4. data processing in a connectionless manner
5. provision for hierarchical XML view of relational data
6. usage of tools like XSLT and XPath Query to operate on the data

The .NET framework data provider includes the following components for data manipulation:

* Connection: This provides connectivity to the data source
* Command: This executes the database statements needed to retrieve data, modify data or execute stored procedures.
* DataReader: This retrieves data in forward only and read-only form.
* DataAdapter: This acts as bridge between dataset and data source to load the dataset and reconcile changes made in dataset back to the source.

ADO.NET entity framework of .NET framework 4.0, the new technology of ADO.NET, abstracts the level of data programming so as to eliminate the impedance mismatch between data models and languages, which application developers would otherwise have to deal with.

**Different**

**Difference between ADO and ADO.net**   
1. ADO used connected data usage, while ADO.net used disconnected data environment.  
2. ADO used OLE DB to access data and is COM-based, while ADO.net uses XML as the format for transmitting data to and from your database and web application.  
3. In ADO, Record set, is like a single table or query result, while in ADO.net Dataset, can contain multiple tables from any data source.  
4. In ADO, it is sometime problematic because firewall prohibits many types of request, while in ADO.net there is no such problem because XML is completely firewall-proof.

**Difference between ADO.net Dataset and ADO Recordset**1 A DataSet can represent an entire relational database in memory, complete with tables, relations, and views.  
2 A DataSet is designed to work without any continuing connection to the original data source.  
3 Data in a DataSet is bulk-loaded, rather than being loaded on demand.  
There's no concept of cursor types in a DataSet.  
4 DataSets have no current record pointer You can use For Each loops to move through the data.  
5 You can store many edits in a DataSet, and write them to the original data source in a single operation.  
6 Though the DataSet is universal, other objects in ADO.NET come in different versions for different data sources.

# Que:- Different between asynorach postback and asynchronous postback?

## AsyncPostBackTrigger

* Converts postbacks into async callbacks
* Typically used to trigger updates when controls outside an UpdatePanel post back
* If ChildrenAsTriggers="false", can be used to specify which controls inside UpdatePanel should call back rather than post back

## PostBackTrigger

* Lets controls inside UpdatePanel post back.
* Typically used to allow certain controls to post back when ChildrenAsTriggers="true"

# Que:- Different between User Control, Custom Control and Component

The main difference between User Control, Custom Control and Component is that they inherit from different levels in the inheritance tree:

MyComponent

|-> Component

MyCustomControl

|-> Control

|-> Component

MyUserControl

|-> ContainerControl

|-> ScrollableControl

|-> Control

|-> Component

So, in short you get a different amount of pre-wired functionality with the different options.

When would you use the different options? (these are thoughts and opinions, not truths)

* Create a **component** if you want to provide **functionality without UI** (such as Timer components, data sources, ...)
* Create a **custom control** if you want to make a component where you have full **control over its visual appearance**, and you don't want any baggage of unnecessary functionality. Typical cases would be simple controls with limited functionality (such as a button)
* Create a **user control** if you are going to **combine existing controls** into reusable building blocks (such as two lists with buttons where you can move items between the lists).

# ****Usercontrol:**** 1) User control can be used for the Reusable purpose only. 2) Once you create User control that can be access in current project. 3) User control extenction is .ascx file. 4) It can be visible in Solution explorer. 5) It is Locally used.If you want used this control in any page just drag and drop from Solution or Register that particular page like <%@ Register TagPrefix="scott" TagName="header" Src="Controls/Header.ascx" %> <% ****Custom Control:**** 1) Custom control can used for Globale purpose like Tollbox controls. 2) custom control can created extenction file is .dll . 3) It can Add to Toolbox like Right click on toolbox add->choose itemes->select path of file. 4) custom control If you want used in any forms just you can drag and drop like normal contro

# Que:- Different between nvarchar and varchar?

Below table lists out the major difference between the VARCHAR and NVARCHAR Data Type in Sql Server:

|  |  |  |
| --- | --- | --- |
|  | Varchar[(n)] | NVarchar[(n)] |
| Basic Definition | Non-Unicode **Var**iable Length **char**acter data type. Example: DECLARE @FirstName AS VARCHAR(50) =‘BASAVARAJ’ SELECT @FirstName | U**N**icode **Var**iable Length **char**acter data type. It can store both non-Unicode and Unicode (i.e. Japanese, Korean etc) characters. Example: DECLARE @FirstName AS NVARCHAR(50)= ‘BASAVARAJ’ SELECT @FirstName |
| No. of Bytes required for each character | It takes 1 byte per character  Example: DECLARE @FirstName AS VARCHAR(50) = ‘BASAVARAJ’ SELECT @FirstName AS FirstName, DATALENGTH(@FirstName) AS  Length  **Result:** FirstName Length BASAVARAJ **9** | It takes 2 bytes per Unicode/Non-Unicode character. Example: DECLARE @FirstName AS NVARCHAR(50)= ‘BASAVARAJ’ SELECT @FirstName AS FirstName, DATALENGTH(@FirstName) AS Length  **Result:** FirstName Length BASAVARAJ **18** |
| Optional Parameter **n** range | Optional Parameter n value can be from 1 to 8000.Can store maximum 8000 Non-Unicode characters. | Optional Parameter n value can be from 1 to 4000.Can store maximum 4000 Unicode/Non-Unicode characters |
| If Optional Parameter **n** is not specified in the variable declaration or column definition | If Optional parameter value ***n*** is not specified in the variable declaration or column definition then it is considered as 1. Example: DECLARE @firstName VARCHAR = ‘BASAVARAJ’ SELECT @firstName FirstName, DATALENGTH(@firstName) Length  **Result:** FirstName Length **B 1** | If Optional parameter value ***n*** is not specified in the variable declaration or column definition then it is considered as 1. Example: DECLARE @firstName NVARCHAR = ‘BASAVARAJ’ SELECT @firstName FirstName, DATALENGTH(@firstName) Length  **Result:** FirstName Length **B 2** |
| If Optional Parameter **n** is not specified in while using CAST/ CONVERT functions | When this optional parameter n is not specified while using the CAST/CONVERT functions, then it is considered as 30.Example: DECLARE @firstName VARCHAR(35) = ‘BASAVARAJ PRABHU BIRADAR INDIA ASIA’  SELECT CAST(@firstName AS VARCHAR) FirstName, DATALENGTH(CAST(@firstName AS VARCHAR)) Length  **Result:** FirstName Length BASAVARAJ PRABHU BIRADAR INDIA 30 | When this optional parameter n is not specified while using the CAST CONVERT functions, then it is considered as 30.Example: DECLARE @firstName NVARCHAR(35) = ‘BASAVARAJ PRABHU BIRADAR INDIA ASIA’  SELECT CAST(@firstName AS NVARCHAR) FirstName, DATALENGTH(CAST(@firstName AS NVARCHAR)) Length  Result: FirstName Length BASAVARAJ PRABHU BIRADAR INDIA 60 |
| Which one to use? | If we know that data to be stored in the column or variable doesn’t have any Unicode characters. | If we know that the data to be stored in the column or variable can have Unicode characters. |
| Storage Size | Takes no. of bytes equal to the no. of Characters entered plus two bytes extra for defining offset. | Takes no. of bytes equal to twice the no. of Characters entered plus two bytes extra for defining offset. |

*As both of these are variable length datatypes, so irrespective of the length (i.e. optional parameter* ***n*** *value) defined in the variable declaration/column definition it will always take the no. of bytes required for the actual charcters stored. The value of* ***n*** *defines maximum no. of characters that can be stored.*

# Que:- Different between KEY in sql

**Difference between Primary Key & Foreign Key**

Primary Key Foreign Key

Primary key uniquely identify a record in the table.

Foreign key is a field in the table that is primary key in another table.

Primary Key can't accept null values.

Foreign key can accept multiple null value.

By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index.

Foreign key do not automatically create an index, clustered or non-clustered. You can manually create an index on foreign key.

We can have only one Primary key in a table.

We can have more than one foreign key in a table.

**Difference between Primary Key & Unique Key**

Primary Key Unique Key

Primary Key can't accept null values.

Unique key can accept only one null value.

By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index.

By default, Unique key is a unique non-clustered index.

We can have only one Primary key in a table.

We can have more than one unique key in a table.

Primary key can be made foreign key into another table.

In SQL Server, Unique key can be made foreign key into another table.

# Difference Between Candidate Keys and Primary Key

**Candidate Key** – A Candidate Key can be any column or a combination of columns that can qualify as unique key in database. There can be multiple Candidate Keys in one table. Each Candidate Key can qualify as Primary Key.

**Primary Key** – A Primary Key is a column or a combination of columns that uniquely identify a record. Only one Candidate Key can be Primary Key.

One needs to be very careful in selecting the Primary Key as an incorrect selection can adversely impact the database architect and future normalization. For a Candidate Key to qualify as a Primary Key, it should be Non-NULL and unique in any domain. I have observed quite often that Primary Keys are seldom changed. I would like to have your feedback on not changing a Primary Key.

# Types Of Key

* Alternate key - An alternate key is any candidate key which is not selected to be the primary key
* Candidate key - A candidate key is a field or combination of fields that can act as a primary key field for that table to uniquely identify each record in that table.
* Compound key - compound key (also called a composite key or concatenated key) is a key that consists of 2 or more attributes.
* Primary key - a primary key is a value that can be used to identify a unique row in a table. Attributes are associated with it. Examples of primary keys are Social Security numbers (associated to a specific person) or ISBNs (associated to a specific book). In the relational model of data, a primary key is a candidate key chosen as the main method of uniquely identifying a tuple in a relation.
* Superkey - A superkey is defined in the relational model as a set of attributes of a relation variable (relvar) for which it holds that in all relations assigned to that variable there are no two distinct tuples (rows) that have the same values for the attributes in this set. Equivalently a superkey can also be defined as a set of attributes of a relvar upon which all attributes of the relvar are functionally dependent.
* Foreign key - a foreign key (FK) is a field or group of fields in a database record that points to a key field or group of fields forming a key of another database record in some (usually different) table. Usually a foreign key in one table refers to the primary key (PK) of another table. This way references can be made to link information together and it is an essential part of database normalization

# Or

## Super Key

Super key is a set of one or more than one keys that can be used to identify a record uniquely in a table.**Example :** Primary key, Unique key, Alternate key are subset of Super Keys.

## Candidate Key

A Candidate Key is a set of one or more fields/columns that can identify a record uniquely in a table. There can be multiple Candidate Keys in one table. Each Candidate Key can work as Primary Key.

**Example:** In below diagram ID, RollNo and EnrollNo are Candidate Keys since all these three fields can be work as Primary Key.

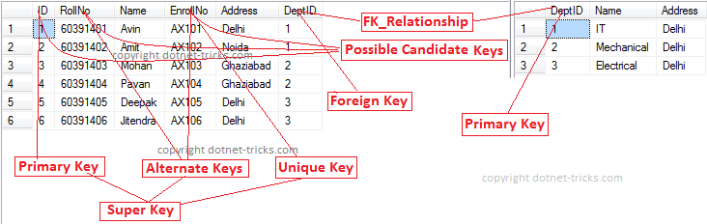
## Primary Key

Primary key is a set of one or more fields/columns of a table that uniquely identify a record in database table. It can not accept null, duplicate values. Only one Candidate Key can be Primary Key.

## Alternate key

A Alternate key is a key that can be work as a primary key. Basically it is a candidate key that currently is not primary key.

**Example:** In below diagram RollNo and EnrollNo becomes Alternate Keys when we define ID as Primary Key.



## Composite/Compound Key

Composite Key is a combination of more than one fields/columns of a table. It can be a Candidate key, Primary key.

## Unique Key

Uniquekey is a set of one or more fields/columns of a table that uniquely identify a record in database table. It is like Primary key but it can accept only one null value and it can not have duplicate values. For more help refer the article [Difference between primary key and unique key](http://www.dotnet-tricks.com/Tutorial/sqlserver/V2bS260912-Difference-between-Primary-Key-and-Unique-Key.html).

## Foreign Key

Foreign Key is a field in database table that is Primary key in another table. It can accept multiple null, duplicate values. For more help refer the article [Difference between primary key and foreign key](http://www.dotnet-tricks.com/Tutorial/sqlserver/TENc260912-Difference-between-Primary-Key-and-Foreign-Key.html).

**Example :** We can have a DeptID column in the Employee table which is pointing to DeptID column in a department table where it a primary key.

### **Que:-Difference between Clustered Index and Non clustered Index**

[Listen to this article. Powered by Odiogo.com](javascript:showOdiogoReadNowFrame%20('501286',%20'difference%20between%20clustered%20index%20%20and%20non%20clustered%20index',%20'0',%20290,%2055);)

**Indexes**-Indexing  is way to sort and search records in the table. It will improve the speed of locating and retrieval of records from the table.It can be compared with the index which we use in the book to search a particular record.

In Sql Server there are two types of Index

1) Clustered Index

2) Non Clustered Index

**Clustered Index:**- Clustered index physically stored the data of the table in the order of the keys values and the data is resorted every time whenever a new value is inserted or a value is updated in the column on which it is defined.

In a table only 1clustered index is possible.

In a clustered Index, the leaf node contains the actual data.

**Non Clustered Index:-** In case of Non clustered index it create a sperate list of key values (or created a table of pointers) which points towards the location of the datain the data pages.

In a table 249 non clustered index is possible.

In a non clustered Index, the leaf node contains the pointer to the data rows of the table.

# Que:- Different between AJAX and Jqury and Javascipt

There are multiple of languages that can be used in making a web page nowadays, some are even not even unique but just a derivative of another language. jQuery is one of these derivatives. It is a lightweight library of [JavasScript](http://www.differencebetween.net/technology/difference-between-java-and-javascript/) that focuses more on interactions with HTML elements. [AJAX](http://www.differencebetween.net/technology/difference-between-ajax-and-silverlight/), on the other hand, is not a specific technology but a combination of varying technologies to provide a new functionality. Whenever you request a new set of data from web site, it clears the whole page and loads the new one. [AJAX](http://www.differencebetween.net/technology/difference-between-ajax-and-javascript/) is used to circumvent this behavior and allow new data to be retrieved without modifying the whole page.

AJAX is a very powerful tool to use but it can’t be utilized with simple HTML since HTML doesn’t allow the page to be changed after it has fully loaded. In order to use AJAX, you would need a client side scripting language that allows you to detect the actions of the user and modify elements on the page accordingly. jQuery does that exactly, that is why both are often used together to present web pages that a user can interact with easily without repetitive loading.

jQuery does all the work on the front end, therefore you would need to have a full understanding of it in order to properly set-up your page. You would not need to learn the exact mechanisms of AJAX in order to utilize it as jQuery gives you an AJAX command to retrieve whichever data you need from the server.

Although the use of jQuery and AJAX makes the browsing experience a lot better for the user, the effect to the server hosting these files are not as desirable. Every time you make another AJAX request, a new connection to the server is made. Too many connections can sometimes be difficult for the server to cope with. Most hosting companies have made steps in order to prevent overloads since jQuery and [AJAX](http://www.differencebetween.net/technology/difference-between-ajax-and-dhtml/) are truly here to stay.

Summary:  
1. JQuery is a lightweight client side scripting library while AJAX is a combination of technologies used to provide asynchronous data transfer  
2. jQuery and AJAX are often used in conjunction with each other  
3. jQuery is primarily used to modify data on the screen dynamically and it uses AJAX to retrieve data that it needs without changing the current state of the displayed page  
4. Heavy usage of AJAX functions often cause server overload due to the greater number of connections made

Read more: [Difference Between jQuery and AJAX | Difference Between | jQuery vs AJAX](http://www.differencebetween.net/technology/difference-between-jquery-and-ajax/#ixzz35xbb0BxO) <http://www.differencebetween.net/technology/difference-between-jquery-and-ajax/#ixzz35xbb0BxO>

or

# ****AJAX****

1. AJAX = Asynchronous JavaScript and XML.
2. AJAX is a technique for creating fast and dynamic web pages.
3. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
4. Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

# JQUERY

1. JQuery is a lightweight, "write less, do more", JavaScript library.
2. The purpose of jQuery is to make it much easier to use JavaScript on your website.
3. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
4. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

So you can code in Jquery very easily wherever you need a AJAX technique

|  |  |
| --- | --- |
| 78 down vote accepted | [AJAX](http://en.wikipedia.org/wiki/Ajax%5F%28programming%29) is a technique to do an [XMLHttpRequest](http://en.wikipedia.org/wiki/XMLHttpRequest) (out of band Http request) from a web page to the server and send/retrieve data to be used on the web page. AJAX stands for Asynchronous Javascript And XML. It uses javascript to construct an XMLHttpRequest, typically using different techniques on various browsers.  [jQuery](http://en.wikipedia.org/wiki/Jquery) ([website](http://jquery.com)) is a javascript framework that makes working with the DOM easier by building lots of high level functionality that can be used to search and interact with the DOM. Part of the functionality of jQuery implements a high-level interface to do AJAX requests. jQuery implements this interface abstractly, shielding the developer from the complexity of multi-browser support in making the request. |
| 7 down vote accepted | AJAX = Asynchronous JavaScript and XML.  AJAX is not a programming language, but a new way to use existing standards.  AJAX is simply the act of sending and receiving data asynchronously using JavaScript and XML.  Often AJAX is used to Update parts of a web page without the need to reload the whole page |

Or

differences between AJAX and JavaScript:

|  |  |
| --- | --- |
| **AJAX** | **JavaScript** |
| AJAX allows the coder send request data asynchronously in order load new data without changing the web page. | JavaScript is a client side scripting language that allows the creation of dynamic web pages by providing a new level of interactivity. |
| AJAX supports the server side scripting Language. | JavaScript provides support to the client side scripting language. |
| AJAX can load the web page after it is been loaded for the first time. | JavaScript cannot load the pages after it is once loaded. |
| AJAX does not install Trojan in the computer. | JavaScript can install Trojan in the computer. |

Or

JavaScript is a programming language   
  
Ajax is shorthand for "Using JavaScript to send or receive information to/from a webserver without leaving the current page"   
  
jQuery is a library of functions (written in JavaScript) that perform some commonly desired things so that author's don't need to reinvent some common wheels.

1. Javascript   
  
A language that comes with your browser. It allows you to show an alert box, the one that you need to press Ok, validate forms, on the client, or without change the page. Usually, it is stopped by an alert, confirm, etc box.   
  
2. jQuery   
  
A library code made with Javascript, that make easier to program Javascript and AJAX. You can download it on jquery.com .   
  
3. AJAX   
  
Asynchronous Javascript And XML. Which means that with a help from a programming language like PHP, ASP.NET and etc. it can work changing data without refresh the whole page, just the part of the data will be refreshed

# Que:- Different between 3.5 and 4

# Read From pdf

# 1) Client Data access:

**ASP.NET 3.5:** There is no direct method to access data from client side. We can go for any of these methods  
  
1) Pagemethods of script manager  
2) ICallbackEventHandler interface  
3) XMLHttphanlder component  
  
**ASP.NET 4.0:** In this framework there is an inbuilt feature for this. Following are the methods to implement them.  
  
1) Client data controls   
2) Client templates   
3) Client data context   
  
i.e we can access the data through client data view & data context objects from client side.

# 2) Setting Meta keyword and Meta description:

Meta keywords and description are really useful for getting listed in search engine.  
  
**ASP.NET 3.5:** It has a feature to add meta as following tag

<meta name="keywords" content="These, are, my, keywords" />   
 <meta name="description" content="This is the description of my page" />

**ASP.NET 4.0:** Here we can add the keywords and description in Page directives itself as shown below.

< %@ Page Language="C#" CodeFile="Default.aspx.cs"   
 Inherits="\_Default"   
 Keywords="Keyword1,Key2,Key3,etc"   
 Description="description" %>

# 3) Enableviewstage property for each control

**ASP.NET 3.5:** this property has two values “True” or “false”  
  
**ASP.NET 4.0:** ViewStateMode property takes an enumeration that has three values: Enabled, Disabled, and Inherit.  
Here inherit is the default value for child controls of a control.

# 4) Setting Client IDs

# Some times ClientID property creates head ach for the programmers. ASP.NET 3.5: We have to use ClientID property to find out the id which is dynamically generated ASP.NET 4.0: The new ClientIDMode property is introduced to minimize the issues of earlier versions of ASP.NET. It has following values. AutoID – Same as ASP.NET 3.5 Static – There won’t be any separate clientid generated at run time Predictable-These are used particularly in datacontrols. Format is like clientIDrowsuffix with the clientid vlaue Inherit- This value specifies that a control's ID generation is the same as its parent.

# Que:- Different between tostring() and covert.tostring() ? The basic difference between them is “Convert.ToString(variable)” handles NULL values even if variable value become null but “variable.ToString()” will not handle NULL values it will throw a NULL reference exception error. So as a good coding practice using “convert” is always safe.

**What is the difference between int.Parse and int.TryParse methods?**

[Page copy protected against web site content infringement by Copyscape](http://www.copyscape.com/)

***int.Parse()*** is a simple method used to convert *string to integer*. It throws exception when null orinvalid input is provided with the Parse() method. Hence it is slow.  
  
**As for Example :**

       string strQty = "a10";  
        //This Line Will Throw Exception "Input string was not in a correct format."  
        int qty = int.Parse(strQty);   
        Response.Write(qty.ToString());

So, when you  are using *Int.Parse()* in your code, you need handle the exception properly.

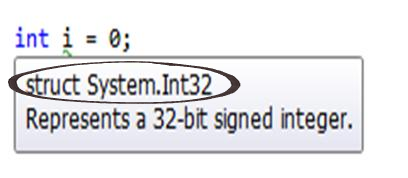
    try  
        {  
            string strQty = "a10";  
            //This Line Will Throw Exception "Input string was not in a correct format."  
            int qty = int.Parse(strQty);  
            Response.Write(qty.ToString());  
        }  
        catch(FormatException fx)  
        {  
          // Handle the exception  
        }

Int.TryParse() does not throw any exception instead it returns *Boolean* value which represent the success *or* faliure.  We must describe second parameter as out  parameter which holds result.   
 **As for Example:**

string strValue = "DotNetFunda";  
        int intval = 0;  
        if (int.TryParse(strValue, out intval))  
        {  
            Response.Write("String With Integer : " + intval);  
        }  
        else  
        {  
           Response.Write("String Without Integer");  
        }

Here The output will be "String Without Integer";

# What is the difference between “int” and “Int32” in C#

"int” and “Int32” are both one and same . If you move your mouse on the integer data type you can see “int” is actually just “Int32”. In other words it will store max value of 2 ^ 32 and minimum value of -2 ^ 32.  
  
Below is a nice video which talks about one more important .NET interview question: - What is the use of param key word ?

# What is the difference between char (10) and nchar (10)?

|  |  |
| --- | --- |
| **char** | **nchar** |
| char are fixed length data-types. | nchar are also fixed length data-types. |
| char does not support Unicode character. | nchar support Unicode character. |
| char reserves 1 byte(8 bits) of memory space. | nchar reserves 2 bytes (16 bits) of memory space. |
| char support Ansi code character, which max upto 256 character. | nchar support Unicode character, which max upto 65536 character. |
| char(n) specifies a length of n bytes by default. | nchar(n) specifies a length of n characters by default. |
| char does not support character of different languages. | nchar support characters of different languages. |

In order to view the exact difference between the char(10) and nvarchar(10), we will use DataLength.

# Difference between Cache and Session?

Session are private to the user.  
Cache are global to the application.  
  
Sessions may change from user to user whereas a single  
Cache will be maintained for the entire application.  
  
The following code shows how you can declare Session and Cache.

Session["MySession"] = TextBox1.Text;// Here Session is declared.  
 Cache.Insert("MyCache", TextBox2.Text);// Here Cache is declared.

# Difference between Cache and Application?

Application and Cache both help to share global data and cache data across the users.but cache object is proactive and you can define dependency.In application object you can't define depenency.

Below is the code for declaring application and cache object.

CacheDependency objCacheDependency = new CacheDependency(Server.MapPath("Banner.txt"));  
Cache.Insert("Banner", strBanner, objCacheDependency);  
Application["Banner"] = "strBanner";

# What is the difference between app.config, web.config and machine.config ?

In this .NET Interview questions interviewer expects two things. First the importance of configuration and second in which scenarios are the above file applicable. So lets answer the question in two parts.

**The importance of config files**==================================================  
App.config, web.config and machine.config are files which store configuration data in XML

format. We need configuration data at application level or at machine/server level.

**Scenarios in which the above config files are used**===================================================  
Machine.config file stores configuration information at system level. It can contain configuration information like timeout in ASP.NET application, requestLimit, memoryLimit, and ClientConnectedCheck etc.

Generally we have two kinds of application web application and windows application. Web.config file stores configuration data for web applications and app.config file store configuration information for windows application.

Application level configuration data can be like connection strings,security etc.

**Some connected questions from this question which the interviewer can ask**================================================================================  
Where do you store connectionstring ?.  
How do you encrypt connection string?  
How do you read configuration data from web.config or app.config file in .NET ?  
Can we have two web.config files for a web application.

**What is the difference between abstraction and encapsulation ?**

This is a very typical [.](http://www.questpond.com)NET  interview question which confuses most of the .NET professionals. Both abstraction and encapsulation look similiar , but they have huge differences between them.

Abstraction is nothing but simplifying objects while encapsulation is hiding complexity.

Encapsulation implements abstraction. Abstraction is a thought process which happens during planning phase.  While encapsulation implements abstraction by using access modifiers ( private,public, protected,internal and protected internal).

### **Difference between throw vs throws in Java**

throw and throws are two Java keyword related to Exception feature of Java programming language. If you are writing Java program and familiar with What is Exception in Java, its good chance that you are aware of [What is throw and throws in Java](http://javarevisited.blogspot.sg/2012/02/difference-between-throw-and-throws-in.html). In this Java tutorial we will compare throw vs throws and see some worth noting difference between throw and throws in Java. Exception handling is an important part of [Java programming](http://javarevisited.blogspot.sg/2011/11/run-java-program-from-command-prompt.html) language which enables you to write robust programs. There are five keywords related to Exception handling in Java e.g. try, catch, finally, throw and throws. Apart from difference between [final](http://javarevisited.blogspot.sg/2011/12/final-variable-method-class-java.html), finally and finalize, throw vs throws is one of the frequently asked [Java interview question](http://java67.blogspot.sg/2012/09/top-10-tough-core-java-interview-questions-answers.html). Difference listed here will not only help you to answer this interview question but also understand concept behind throw and throws better.

1) throw keyword is used to throw Exception from any method or [static block in Java](http://javarevisited.blogspot.sg/2011/11/static-keyword-method-variable-java.html) while throws keyword, used in method declaration, denoted which Exception can possible be thrown by this method.

2) If any method throws [checked Exception](http://javarevisited.blogspot.sg/2011/12/checked-vs-unchecked-exception-in-java.html) as shown in below Example, than caller can either handle this exception by catching it or can re throw it by declaring another throws clause in method declaration.

public void read() throws IOException{

throw new IOException();

}

failure to either catch or declaring throws in [method signature](http://java67.blogspot.sg/2012/08/what-is-method-overloading-in-java-example.html) will result in compile time error.

3) throw keyword can be used in [switch case in Java](http://java67.blogspot.sg/2012/09/how-to-use-java-enum-in-switch-case-example.html) but throws keyword can not be used anywhere except on method declaration line.

4) As per [Rules of overriding in Java](http://java67.blogspot.sg/2012/09/what-is-rules-of-overloading-and-overriding-in-java.html), overriding method can not throw [Checked Exception](http://javarevisited.blogspot.sg/2011/12/checked-vs-unchecked-exception-in-java.html) higher in hierarchy than overridden method . This is rules for throws clause while [overriding method in Java](http://javarevisited.blogspot.sg/2011/12/method-overloading-vs-method-overriding.html).

5) throw transfers control to caller, while throws is suggest for information and compiler checking.

6) Both [Checked and Unchecked Exception](http://javarevisited.blogspot.sg/2011/12/checked-vs-unchecked-exception-in-java.html) can be declared to be thrown using throws clause in Java.

That's all on **difference between throw vs throws in Java** and Exception handling. You must try some example to use throw and throws as well and rather importantly you must know when to use throw and throws keyword in Java. In summary use throw to actually throw exception which will give control back to caller and use throws to declare which Exception can be thrown by a particular method, which allows caller to handle them.

|  |  |
| --- | --- |
| 1)throw is used to explicitly throw an exception. | throws is used to declare an exception. |
| 2)checked exception can not be propagated without throws. | checked exception can be propagated with throws. |
| 3)throw is followed by an instance. | throws is followed by class. |
| 4)throw is used within the method. | throws is used with the method signature. |
| 5)You cannot throw multiple exception | You can declare multiple exception e.g. public void method()throws IOException,SQLException. |

**Que:- types of css**

1)Internal

2)Inline

3)External

* **External style sheet**, which you use when you want to apply the same styles consistently across all the pages in your Web site that are linked to it. Also known as linked style sheet.
* **Embedded style sheet**, which you use when you want to define styles for the current page
* **Inline style sheet**, which you apply to individual elements on a page.

#### Other benefits of CSS

There are a lot of other reasons for using cascading style sheets, including:

* More control over the display of individual elements on a page.
* Faster download times.
* Easier site management.

http://www.expression-web-tutorial.com/Types\_CSS\_Styles.html#.U-ruYmPpws0

**Que:-Difference between HTML and HTML5:**

# -see html and html5.pdf

## **Que:-Advantages of Stored Procedures**

To help you build powerful database applications, stored procedures provide several advantages including better performance, higher productivity, ease of use, and increased scalability.

### Performance

Stored procedures are compiled once and stored in executable form, so procedure calls are quick and efficient. Executable code is automatically cached and shared among users. This lowers memory requirements and invocation overhead.

By grouping SQL statements, a stored procedure allows them to be executed with a single call. This minimizes the use of slow networks, reduces network traffic, and improves round-trip response time. OLTP applications, in particular, benefit because result set processing eliminates network bottlenecks.

Additionally, stored procedures enable you to take advantage of the computing resources of the server. For example, you can move computation-bound procedures from client to server, where they will execute faster. Likewise, stored functions called from SQL statements enhance performance by executing application logic within the server.

### Productivity and Ease of Use

By designing applications around a common set of stored procedures, you can avoid redundant coding and increase your productivity. Moreover, stored procedures let you extend the functionality of the RDBMS. For example, stored functions called from SQL statements enhance the power of SQL.

You can use the Java integrated development environment (IDE) of your choice to create stored procedures. Then, you can deploy them on any tier of the network architecture. Moreover, they can be called by standard Java interfaces such as JDBC, CORBA, and EJB and by programmatic interfaces and development tools such as SQLJ, the OCI, Pro\*C/C++, and JDeveloper.

This broad access to stored procedures lets you share business logic across applications. For example, a stored procedure that implements a business rule can be called from various client-side applications, all of which can share that business rule. In addition, you can leverage the server's Java facilities while continuing to write applications for your favorite programmatic interface.

### Scalability

Stored procedures increase scalability by isolating application processing on the server. In addition, automatic dependency tracking for stored procedures aids the development of scalable applications.

The shared memory facilities of the Multi-Threaded Server (MTS) enable Oracle8i to support more than 10,000 concurrent users on a single node. For more scalability, you can use the Net8 Connection Manager to multiplex Net8 connections.

### Maintainability

Once it is validated, a stored procedure can be used with confidence in any number of applications. If its definition changes, only the procedure is affected, not the applications that call it. This simplifies maintenance and enhancement. Also, maintaining a procedure on the server is easier than maintaining copies on various client machines.

### Interoperability

Within the RDBMS, Java conforms fully to the Java Language Specification and furnishes all the advantages of a general-purpose, object-oriented programming language. Also, like PL/SQL, Java provides full access to Oracle data, so any procedure written in PL/SQL can be written in Java.

PL/SQL stored procedures complement Java stored procedures. Typically, SQL programmers who want procedural extensions favor PL/SQL, and Java programmers who want easy access to Oracle data favor Java.

The RDBMS allows a high degree of interoperability between Java and PL/SQL. Java applications can call PL/SQL stored procedures using an embedded JDBC driver. Conversely, PL/SQL applications can call Java stored procedures directly.

### Security

You can restrict access to Oracle data by allowing users to manipulate the data only through stored procedures that execute with their definer's privileges. For example, you can allow access to a procedure that updates a database table, but deny access to the table itself.

### Replication

With Oracle Advanced Replication, stored procedures can be replicated (copied) from one Oracle8i database to another. This feature makes them ideal for implementing a central set of business rules. Once written, the stored procedures are replicated and distributed to work groups and branch offices throughout the company. In this way, policies can be revised on a central server rather than on individual servers.

Stored procedures have been viewed as the de facto standard for applications to access and manipulate database information through the use of codified methods, or "procedures." This is largely due to what they offer developers: the opportunity to couple the set-based power of SQL with the iterative and conditional processing control of code development. Developers couldn't be happier about this; finally, instead of writing inline SQL and then attempting to manipulate the data from within the code, developers could take advantage of:

* **Familiar Coding Principles**
  + Iterative Loops
  + Conditionals
  + Method Calls (the stored procedure itself is built and similarly called like a method)
* **One-time, One-place Processing**
  + Instead of having inline SQL code spread throughout the application, now sections of SQL code can be encapsulated into chunks of named methods that are easily identifiable and accessible all within one location – the "Stored Procedure" folder of the database.
  + All complex data processing can now be performed on the server, allowing the client processing to focus more on presentation rather than manipulation of data.

Of course, just because something is popular doesn't always mean that it's the best tool in all situations. The efficiency, efficacy and utility of Stored Procedures, just like the implementation of all programming languages and platforms, are all dependent on the needs of the client and the subsequent architecture of the application.

## Advantages of Using Stored Procedures

Stored procedures are so popular and have become so widely used and therefore expected of Relational Database Management Systems (RDBMS) that even MySQL finally caved to developer peer pressure and added the ability to utilize stored procedures to their very popular [open source](http://www.seguetech.com/blog/2013/03/27/open-source-best-for-your-company) database. The list below details why stored procedures have gained such a stalwart following among application developers (and even Database Administrators for that matter):

* **Maintainability** 
  + Because scripts are in one location, updates and tracking of dependencies based on schema changes becomes easier
* **Testing**
  + Can be tested independent of the application
* **Isolation of Business Rules**
  + Having Stored Procedures in one location means that there's no confusion of having business rules spread over potentially disparate code files in the application
* **Speed / Optimization**
  + Stored procedures are cached on the server
  + Execution plans for the process are easily reviewable without having to run the application
* **Utilization of Set-based Processing**
  + The power of SQL is its ability to quickly and efficiently perform set-based processing on large amounts of data; the coding equivalent is usually iterative looping, which is generally much slower
* **Security**
  + Limit direct access to tables via defined roles in the database
  + Provide an "interface" to the underlying data structure so that all implementation and even the data itself is shielded.
  + Securing just the data and the code that accesses it is easier than applying that security within the application code itself

## Drawbacks to Using Stored Procedures

There are certainly drawbacks to Stored Procedures that preclude them from being the one-stop shop solution to application database access. The list below contains some reasons why Stored Procedures might not be right for your application solution. Interestingly, you'll probably recognize some headings that also appear in the "Advantages" section above; this is because what one developer views as affirmative evidence for their use might cause another to see the same evidence to disprove their viability as a solution.

* **Limited Coding Functionality**
  + Stored procedure code is not as robust as app code, particularly in the area of looping (not to mention that iterative constructs, like cursors, are slow and processor intensive)
* **Portability**
  + Complex Stored Procedures that utilize complex, core functionality of the RDBMS used for their creation will not always port to upgraded versions of the same database. This is especially true if moving from one database type (Oracle) to another (MS SQL Server).
* **Testing**
  + Any data errors in handling Stored Procedures are not generated until runtime
* **Location of Business Rules**
  + Since SP's are not as easily grouped/encapsulated together in single files, this also means that business rules are spread throughout different Stored Procedures. App code architecture helps to ensure that business rules are encapsulated in single objects.
  + There is a general opinion that business rules / logic should not be housed in the data tier
* **Utilization of Set-based Processing**
  + Too much overhead is incurred from maintaining Stored Procedures that are not complex enough. As a result, the general consensus is that simple SELECT statements should not be bound to Stored Procedures and instead implemented as inline SQL.
* **Cost**
  + Depending on your corporate structure and separation of concern for development, there is the potential that Stored Procedure development could potentially require a dedicated database developer. Some businesses will not allow developers access to the database at all, requiring instead a separate DBA. This will automatically incur added cost.
  + Some companies believe (and sometimes it's true, but not always) that a DBA is more of a SQL expert than an application developer, and therefore will write better Stored Procedures. In that case, an extra developer in the form of a DBA is required.

# Language Integrated Query (LINQ)

**Definition - What does Language Integrated Query (LINQ) mean?**

## Language integrated query (LINQ) is a Microsoft .NET framework programming model, which adds query capabilities to the .NET programming languages. These extensions provide shorter and expressive syntax to manipulate data. A number of features have been added to C# and Visual Basic to support LINQ. It envelops powerful querying on objects, XML files, and databases. LINQ encapsulates heavy generics. Distinguishing features include extension methods, lambda expressions, an object initializer, query syntax, and anonymous types. These are language**Techopedia explains Language Integrated Query (LINQ)**

Extension methods extend existing values without deriving any new type. Lambda expressions create expression trees and delegates, while the object initializer produces object initialization syntax that generates an equivalent code.   
  
A language is free to use its own query syntax, which needs to be translated into LINQ method calls. Anonymous types permit the compiler to compile classes, which contain data member declarations.   
  
LINQ supports a number of LINQ providers, as follows:

* LINQ to XML: Converts XML documents to a group of XElement objects. These objects are then queried by the local execution engine, which is part of the standard query operator.
* LINQ to SQL: Allows LINQ to be used to query SQL server databases. Because SQL server data resides in a remote database and uses its own query engine, this provider converts a LINQ query to SQL query first and sends it to the SQL server for processing.
* LINQ to DataSets: Enables support for any generic database. This provider uses ADO.net, which handles communication with the database.
* LINQ to Objects: Uses the local query execution engine for in-memory collections. The generated code refrences the standard query operators' implementation.

extensions to enhance syntactic performance to queries.

# Q : UNION and UNION ALL in SQL?

Both UNION and UNION ALL concatenate the result of two different SQLs. They differ in the way they handle duplicates.

* UNION performs a DISTINCT on the result set, eliminating any duplicate rows.
* UNION ALL does not remove duplicates, and it therefore faster than UNION.

# https://stackoverflow.com/questions/49925/what-is-the-difference-between-union-and-union-all