**Difference between Asp.Net MVC and Web Forms**

Asp.Net Web Form follow a traditional event driven development model.

Asp.Net MVC is a lightweight and follow MVC (Model, View, Controller) pattern based development model.

Asp.Net Web Form has server controls.

Asp.Net MVC has html helpers.

Asp.Net Web Form supports view state for state management at client side.

Asp.Net MVC does not support view state.

Asp.Net Web Form has file-based URLs means file name exist in the URLs must have its physically existence.

Asp.Net MVC has route-based URLs means URLs are divided into controllers and actions and moreover it is based on controller not on physical file.

Asp.Net Web Form follows Web Forms Syntax

Asp.Net MVC follow customizable syntax (Razor as default)

In Asp.Net Web Form, Web Forms(ASPX) i.e. views are tightly coupled to Code behind(ASPX.CS) i.e. logic.

In Asp.Net MVC, Views and logic are kept separately.

Asp.Net Web Form has Master Pages for consistent look and feels.

Asp.Net MVC has Layouts for consistent look and feels.

Asp.Net Web Form has User Controls for code re-usability.

Asp.Net MVC has Partial Views for code re-usability.

Asp.Net Web Form has built-in data controls and best for rapid development with powerful data access.

Asp.Net MVC is lightweight, provide full control over markup and support many features that allow fast & agile development. Hence it is best for developing interactive web application with latest web standards.

Asp.Net Web Form is not Open Source.

Asp.Net Web MVC is an Open Source.

What’s an extension method?

**Difference between stored procedure and function**

1) Procedure can return zero or n values whereas function can return one value which is mandatory.  
2) Procedures can have input, output parameters for it whereas functions can have only input parameters.  
3) Procedure allows select as well as DML statement in it whereas function allows only select statement in it.  
4) Functions can be called from procedure whereas procedures cannot be called from function.  
5) Exception can be handled by try-catch block in a procedure whereas try-catch block cannot be used in a function.  
6) We can go for transaction management in procedure whereas we can't go in function.  
7) Procedures cannot be utilized in a select statement whereas function can be embedded in a select statement.

**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, but a reference cannot 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 cannot be instantiated, but a reference can be created

**Index types in SQL Server**

**Clustered Index**

Only 1 allowed per table

physically rearranges the data in the table to confirm to the index constraints for use on columns that are frequently searched for ranges of data for use on columns with low selectivity.

**Non-Clustered Index**

Up to 249 allowed per table creates a separate list of key values with pointers to the location of the data in the data pages For use on columns that are searched for single values 

**Interview questions**

**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.

**What are differences between system.stringbuilder and system.string?**

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.

**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.

**What are the different types of indexes?**

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

**How many types of memories are there in .net?**

**Ans:**Two types of memories are there in .net stack memory and heap memory

**Is it possible to set the session out time manually?**

**Ans:**Yes we can set the session out time manually in web.config.

**What is the difference between primary key and unique key with not null?**

**Primary Key**

* Primary key cannot have a NULL value.
* Each table can have only one primary key.
* By default, Primary key is clustered index, and the data in database table is physically organized in the sequence of clustered index.
* Primary key can be related to another tables as a Foreign Key.

**Unique Key**

* Unique Constraint may have a NULL value.
* Each table can have more than one Unique Constraint.
* By default, Unique key is a unique non-clustered index.
* Unique Constraint can not be related with another table's as a Foreign Key.

**What is boxing and unboxing concepts in .net?**

**Ans:** Boxing is a process of converting value type into reference type

Unboxing is a process of converting reference type to value type.

**What are the differences between value type and reference type?**

**Ans:** Value type contain variable and reference type are notcontaining value directly in its memory.

Memory is allocated in managed heap in reference type and in value type memory allocated in stack. Reference type ex-class value type-struct, enumeration

**Is it possible to host the website from desktop?**

**Ans:**Yes

**Why we go for page rendering in Asp.Net Page life cycle?**

**Ans:**Browser understands an only html control that’s why in page rendering we will convert the aspx controls into html controls.

**Write a sample query for self join?**

**Ans:**Select e1.ename, e2.empid from emp e1, emp e2 where e1.empid=e2.mgrid;

**Can we change the index of primary key on table?**

**Ans:**No

**How to change the name of the table or stored procedure in sql?**

**Ans:**sp\_rename oldtablename newtablename

For changing the column name

Sp\_rename  ‘tablename.[Oldcolumnname]’,’newcolumnname’,’Column’

Ex:sp\_rename ‘tblemp.first’,’namechange’,’Column’

**How to find out which index is defined on table?**

**Ans:**sp\_helpindex tablename

**Can you write the program to find the length of string without using library function?**

**Ans:**for (int i=0; str[i]!=”\n”; i++)

{

Count++;

}

**What is the difference between scope\_identity() and current\_identity()?**

**Ans:**Scope\_identity and current \_identity both are similar and it will return the last identity value generated in the table.

Scope\_Identity will return the identity value in table that is currently in scope

**What are difference between GET and POST Methods?**

Get and Post both are used for sending client information to web server means both methods are used to transfer data from client to server.

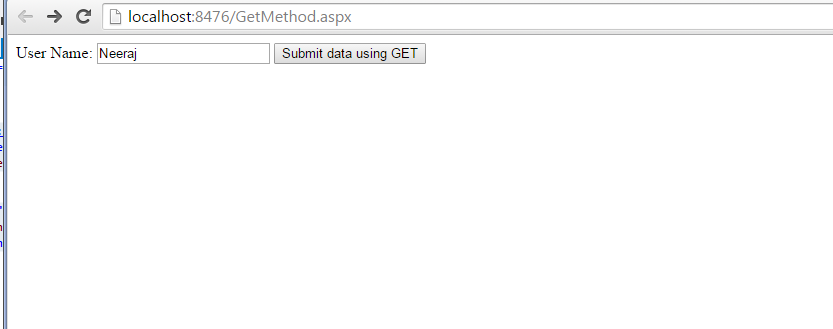
**Get**  
  
Data is passed from client in the form of Url and Url data is visible to every user if you are submitting any form then data which you are passing will be visible in Url ,so this will not safe. also we have some restrictions on that you can pass only 1024 characters in the case of Get.

**Example for this**

I have a application from where i will pass data

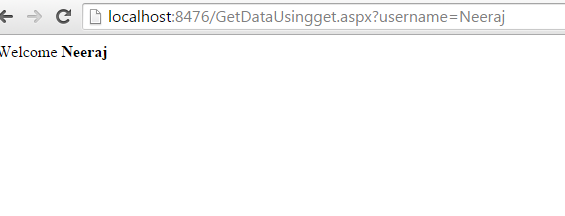
GetMethod.aspx

1. <html
2. xmlns="http://www.w3.org/1999/xhtml">
3. <body>
4. <form method="get" action="GetDataUsingget.aspx">
5. User Name:
6. <input id="txtuserName" type="text" name="username" />
7. <input id="btnSubmit" type="submit" value="Submit data using GET" />
8. </form>
9. </body>
10. </html>



From there i have entered my name and then data will be posted on  "GetDataUsingget.aspx" page.

After clicking on button this will be like as shown in below picture

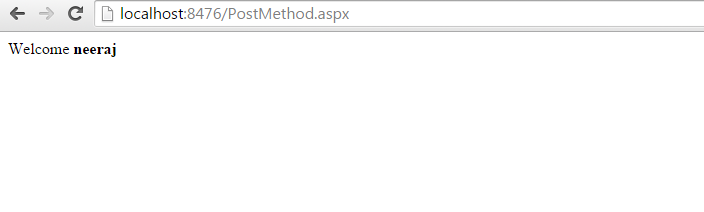


**Post**  
  
On the case of post Method data will be passed through http headers  so using secure http protocol and data will be more secure and also we have no data restriction there we can pass large number of data and binary data we can pass here also.

Prcactical example for this-

1. <html
2. xmlns="http://www.w3.org/1999/xhtml">
3. </head>
4. <body>
5. <form method="post" action="PostMethod.aspx">
6. User Name:
7. <input id="txtuserName" type="text" name="username" />
8. <input id="btnSubmit" type="submit" value="Submit data using GET" />
9. </form>
10. </body>undefined</html>

If you will click on submit button the.



**What are difference between truncate and delete?**

**Ans:** 1) Delete keep the lock over each row where Truncate keeps the lock on table not on all the row.  
2) Counter of the Identity column is reset in Truncate where it is not reset in Delete.   
3) Trigger is not fired in Truncate where as trigger is fired in Delete.

4) In TRUNCATE we cannot rollback.

5) In DELETE we can rollback

**What is the difference Grid View and between Data Grid (Windows)?**

**Ans:**

1) GridView Control Enables you to add sorting, paging and editing capabilities without writing any code.   
2)GridView Control Automatically Supports paging by setting the ‘PagerSetting’ Property.The Page Setting Property supports four Modles

a. Numeric(by default)   
b. Next Previous   
c. NumericFirstLast   
d. Next PreviousLast   
  
3)It is Used in asp.net   
4)GridView Supports RowUpdating and RowUpdated Events.   
5)GidView is Capable of Pre-Operations and Post-Operations.   
6)GridView Has EditTemplates for this control   
7)It has AutoFormat   
  
**DataGrid(Windows)**  
  
1)DataGid Control raises single Event for operations   
2)DataGird Supports the SortCommand Events that occur when a column is Soted.   
3)DataGrid Supports UpdataCommand Event that occurs when the UpdateButton is clicked for an item in the grid.   
4)DataGrid is used in Windows GUI Application.   
5)It doesnot have EditTemplates for this control   
6)It doesnot have AutoFormat

**If I write System.exit (0); at the end of the try block, will the finally block still execute?**

**Ans:**No in this case the finally block will not execute because when you say system.exit(0),the control immediately goes out of the program, and thus finally never executes.

**What are the different levels of State management in ASP.NET?**

**Ans:**

State management is the process by which you maintain state and page information over multiple requests for the same or different pages.

There are 2 types State Management:   
  
1. Client – Side State Management   
This stores information on the client's computer by embedding the information into a Web page, a uniform resource locator (url), or a cookie. The techniques available to store the state information at the client end are listed down below:   
  
a. View State – Asp.Net uses View State to track the values in the Controls. You can add custom values to the view state. It is used by the Asp.net page framework to automatically save the values of the page and of each control just prior to rendering to the page. When the page is posted, one of the first tasks performed by page processing is to restore view state.   
  
b. Control State – If you create a custom control that requires view state to work properly, you should use control state to ensure other developers don’t break your control by disabling view state.   
  
c. Hidden fields – Like view state, hidden fields store data in an HTML form without displaying it in the user's browser. The data is available only when the form is processed.   
  
d. Cookies – Cookies store a value in the user's browser that the browser sends with every page request to the same server. Cookies are the best way to store state data that must be available for multiple Web pages on a web site.   
  
e. Query Strings - Query strings store values in the URL that are visible to the user. Use query strings when you want a user to be able to e-mail or instant message state data with a URL.   
  
2. Server – Side State Management   
a. Application State - Application State information is available to all pages, regardless of which user requests a page.   
  
b. Session State – Session State information is available to all pages opened by a user during a single visit.   
  
Both application state and session state information is lost when the application restarts. To persist user data between application restarts, you can store it using profile properties.

**Below are the few main difference between Abstract Class and Interface**  
  
a.    In abstract class method can have definition as well as declaration also. But Interface should have only declarations.  
b.    All the Methods are Public as default and don’t have any access Modifier Controls in interface, whereas for abstract class we can have access modifier for methods.  
c.    Abstract class can have constructor or destructor, whereas interface not.  
d.    Abstract class can’t be part of multiple inheritance and we can implement multiple interface.

**What do you mean by String objects are immutable?**

String objects are immutable as its state cannot be modified once created. Every time when we perform any operation like add, copy, replace, and case conversion or when we pass a string object as a parameter to a method a new object will be created.  
  
**Example:**

String str = "ABC";

str.Replace("A","X");

Here Replace() method will not change data that "str" contains, instead a new string object is created to hold data "XBC" and the reference to this object is returned by Replace() method.  
  
So in order to point *str* to this object we need to write below line.

str = str.Replace("A","X");

Now the new object is assigned to the variable str. earlier object that was assigned to str will take care by garbage collector as this one is no longer in used.

**What is dll hell problem in .NET and how it will solve?**

**Ans:**Dll hell, is kind of conflict that occurred previously, due to the lack of version supportability of dll for (within) an application

.NET Framework provides operating system with a global assembly cache. This cache is a repository for all the .net components that are shared globally on a particular machine. When a .net component installed onto the machine, the global assembly cache looks at its version, its public key and its language information and creates a strong name for the component. The component is then registered in the repository and indexed by its strong name, so there is no confusion between the different versions of same component, or DLL

**What is a Partial class?**

**Ans:**Instead of defining an entire class, you can split the definition into multiple classes by using partial class keyword. When the application compiled, c# compiler will group all the partial classes together and treat them as a single class. There are a couple of good reasons to use partial classes. Programmers can work on different parts of classes without needing to share same physical file

Ex:

Public partial class employee

{

Public void somefunction()

{

}  
}

Public partial class employee

{

Public void function ()

{

}

}

**When should you use Abstract Class vs Interface while programming?**  
 **Ans:**When we want that sub class must implement all the methods of base class. In such a situation we will implement the **interface.** In the other hand when we want only some method of base class in our sub class then use base class as abstract class.

**What is the difference between application exception and system exception?**

**Ans:**The difference between application exception and system exception is that system exceptions are thrown by CLR and application exceptions are thrown by applications.

**What is the difference between authorization and authentication?**

**Ans:**Authorization is a process of allowing or denying resources to particular user or record

**Declaration of authorization is**

<authorization>

<allow users=”Suresh, Sanjay”/>

<deny users=”Ramana, Rakesh”>

</authorization>

Sometimes authorization allows the unauthorized persons at that time we will use

<deny users=”?”/>

**Authentication means**

Authentication is a process where we identify the credentials of user i.e. username, password and create an identity to mention user as an authenticated.

**What is the serialization?**

**Ans:**Serialization is a process of converting object into a stream of bites.

**What is synchronization?**

**Ans:**The mechanism needed to block one thread access to the data. If the data is being accessed by another thread.

Synchronization can be accessed by using **system.monitor** class

A monitor class methods are enter, exit, pulse for this lock statement is also used

Suppose if we need to synchronize some data at that time we need to place that data in this block

**Lock**

**{**

**}**

Whatever the data has been placed into the lock block that data has been blocked

**What is the difference between .tostring(), Convert.tostring()?**

**Ans:**The basic difference between them is “Convert” function handles NULLS while  
“.ToString()” does not it will throw a NULL reference exception error. So as a good coding  
practice using “convert” is always safe.

**What is Collation?**

**Ans:**Collation refers to a set of rules that determine how the data is sorted and compared.

**How many web.config files are there in 1 project?**

**Ans:** There might be multiple web.config files for a single project depending on the hierarchy of folders inside the root folder of the project, so for each folder we can use one web.config file

**What is the difference between view state and hidden field?**

**Ans:**viewstate is secured hidden field is insecure

Viewstate will store large amount of data but hidden filed will store small amount of data.

**What is the Difference between read only and constant variables?**

**Ans:**Read only can assign the values at runtime only.

Constant will assign the values at compile time only.

We cannot modify the both variable values.

**What is static keyword in .Net?**

**Ans:** Static is same as constant variable but we can change the value of static variable and we can access the variables without creating any instances

### 1.What is the Postback in ASP.NET?

A postback is a request sent from a client to server from the same page, user is already working with.

* + Add a new ASP.NET web form page to a project e.g. WebForm1.aspx.
  + View the page code in HTML Source view.
* Look the form line of code.

<form id=”form1″ runat=”server”>;  
It represents a server-side implementation of form control.

* Now just run the application to see WebForm1.aspx page and view its source code. HTML source of the page will display form element as follows:  
  <code>form id="form1" action="WebForm1.aspx" method="post"&gt;
* You can see that an HTML form element generated with an HTTP method as “POST” and action=”WebForm1.aspx”. So, if a submit button is clicked, the page will postback to itself by default.
* Following figure will also give you more understanding on ASP.NET Postback.

You can read more on postback vs Callback difference [**here**](http://www.csharpstar.com/postback-vs-callback-in-asp-net/).

### 3.Please briefly explain ASP.NET Page life Cycle?ASP.net Page Life Cycle

Initialization: Controls raise their Init event in this stage.Objects and variables are initializes for complete lifecycle of request.  
LoadViewState: is a post back stage and loads the view state for the controls that enabled its view state property.

LoadPostBackData: is also a post back stage and loads the data posted for the controls and update them.   
Load: In this stage page as well as all the controls raise their Load event. Till this stage all the controls are initialized and loaded. In most of the cases, we are coding this event handler.  
RaisePostBackEvent: is again a postback stage. For example, it’s raise against a button click event. We can easily put our code here to perform certain actions.  
SaveViewState: Finally, controls state is saved in this stage before Rendering HTML.  
Render: This is the stage where HTML is generated for the page.  
Dispose: Lastly, all objects associated with the request are cleaned up.

You can see the detailed explanation on ASP.net page life cycle [**here**](http://www.csharpstar.com/asp-net-application-and-page-life-cycle/)

### 4.What is the difference between custom controls and user controls?

Custom controls are basically compiled code i.e. DLLs. These can be easily added to toolbox, so it can be easily used across multiple projects using drag and drop approach. These controls are comparatively hard to create.  
But User Controls (.ascx) are just like pages (.aspx). These are comparatively easy to create but tightly couple with respect to User Interface and code. In order to use across multiple projects, we need to copy and paste to the other project as well.

### 5.Please briefly explain the usage of Global.asax?

Global.asax is basically ASP.NET Application file. It’s a place to write code for Application-level events such as Application start, Application end, Session start and end, Application error etc. raised by ASP.NET or by HTTP Modules.

There is a good list of events that are fired but following are few of the important events in Global.asax:

* Application\_Init occurs in case of application initialization for the very first time.
* Application\_Start fires on application start.
* Session\_Start fires when a new user session starts
* Application\_Error occurs in case of an unhandled exception generated from application.
* Session\_End fires when user session ends.
* Application\_End fires when application ends or time out.

You can read the detailed article [**here**](http://www.csharpstar.com/global-asax-in-asp-net/).

### 6.What is the concept of view state in ASP.NET?

* ViewState of a webform is available only within that webform
* ViewState is stored on the page in a hidden field called viewState. it will be lost if the user will navigate away from the page or if browser is closed
* ViewState is used y all asp.net controls to retain their state across postback

You can read the difference between **view state,session state and Application state** **here**.

### 7.Difference between Response.Redirect and Server.Transfer?

Both Server.Transfer and Response.Redirect are ASP.NET objects and are used for navigation between web-pages. However, there are noticeable differences between these two techniques:

  Using ‘Server. Transfer’ we cannot redirect to external websites or website pages. E.g. if your website is [www.webcodeexpert.com](http://www.webcodeexpert.com/) then you cannot use ‘Server. Transfer’ to move to www.google.com but yes, you can move to internal pages  [www.webcodeexpert.com/asp.net](http://www.webcodeexpert.com/search/label/ASP.NET), i.e. within the websites.

3.     With ‘Response. Redirect’ we can redirect the user to the both type of pages .html or .aspx e.g. Response. Redirect (“OtherPage.html”) OR Response. Redirect (“OtherPage.aspx”) But in case of ‘Server. Transfer’ we can redirect user to .asp or .aspx pages only e.g. Server. Transfer (“OtherPage.asp”) OR Server. Transfer (“OtherPage.aspx”) not to Server. Transfer (“OtherPage.html”).

4.     In ‘Server. Transfer’ URL doesn’t change but in case of ‘Response. Redirect’ URL changes.

 ‘Response. Redirect’ has a round trip but ‘Server.Transfer’ has no round trip. (Roundtrip is the combination of a request being sent to the server and response being sent back to browser.)

7.     ‘Server. Transfer’ is a server process whereas ‘Response. Redirect’ is a client process.

### 8.What are the different types of Validation controls in ASP.NET?

In order to validate user input, ASP.NET provides validation server controls. All validation controls inherits from BaseValidator class which contains the common validation properties and methods like ControlToValidate, Enabled, IsValid, EnableClientScript, ValidationGroup,Validate() etc.

ASP.Net provides a range of validation controls:

* RequiredFieldValidatorvalidates required input.
* RangeValidatorvalidates the range. Validates that input is between the given range values.
* CompareValidatorvalidates or compares the input of a control with another control value or with a fixed value.
* RegularExpressionValidator validates input value against a defined regular expression pattern.
* CustomValidator allows to customize the validation logic with respect to our application logic.
* ValidationSummarydisplays all errors on page collectively.

### 9.What are HttpHandlers and HttpModules in ASP.NET?

HttpHandler: ASP.NET Engine uses HttpHandlers to handle specific requests on the basis of it’s extensions. ASP.NET Page Handler handles all requests coming for (.aspx) pages. We can define our own custom HttpHandler to handle a specific request with a specific extension, say .jpeg, .gif, or .ahmad. But there will always be only one handler for a specific request.

HttpModule: ASP.NET Engine uses HttpModules to inject some specific functionality along with ASP.NET default functionality for all incoming requests regardless of its extensions. There are a number of built-in modules already available in ASP.NET HTTP Pipeline. But we can write our own custom HTTP module to perform some additional functionality (for example, URL rewriting or implementing some security mechanism) for all incoming requests.

We have discussed HttpHandler and HttpModule in detailed [**here**](http://www.csharpstar.com/asynchronous-http-modules-and-handlers-in-asp-net/).

### 10.What are the types of Authentication in ASP.NET?

There are three types of authentication available in ASP.NET:

* **Windows Authentication**: This authentication method uses built-in windows security features to authenticate user.
* **Forms Authentication**: authenticate against a customized list of users or users in a database.
* **Passport Authentication**: validates against Microsoft Passport service which is basically a centralized authentication service.

### 11.What are Session state modes in ASP.NET?

ASP.NET supports different session state storage options:

* In-Process is the default approach. It stores session state locally on same web server memory where the application is running.
* StateServer mode stores session state in a process other than the one where application is running. Naturally, it has added advantages that session state is accessible from multiple web servers in a Web Farm and also session state will remain preserved even web application is restarted.
* SQLServer mode stores session state in SQL Server database. It has the same advantages as that of StateServer.
* Custom modes allows to define our custom storage provider.
* Off mode disables session storage.

### 12.Hyperlink Vs LinkButton in ASP.NET?

A Hyperlink redirects to a given URL identified by “NavigateURL” property but a LinkButton which actually displays a Hyperlink style button causes a postback to the same page but it doesn’t redirect to a given URL.

### 13.Globalization Vs Localization

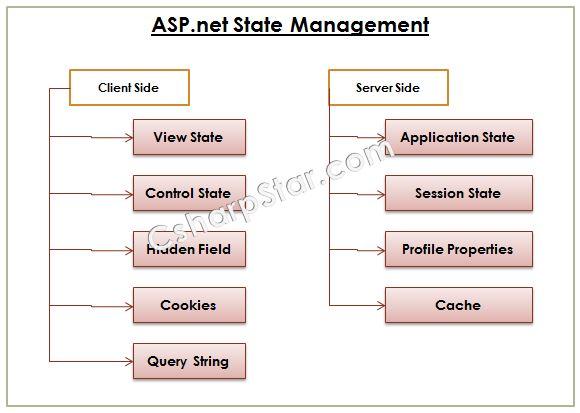
When we need to build an application that work for multiple cultures e.g en-US, ar-SAetc, this process of designing and building applications that work for more than one cultures is called globalization. However, customizing an application for a specific culture is localization. Both globalization and localization normally go together.

### 15.Cache Management in ASP.NET?

ASP.NET provided support for Cache Management in almost all versions. In .NET Framework 3.5 and older, the support for caching was provided through classes available in System.Web.Caching. But this support was limited to System.Web meaning for ASP.NET Web Applications only. Now, with .NET Framework 4.0 and later, this support is enhance to non-Web Applications also by providing APIs in System.Runtime.Caching.

* ASP.NET supports three types of Caching:
  + Page Output Caching
  + Partial Page Caching
  + Data Caching

### 17.What is State Management?



HTTP is a stateless protocol by nature. So, we need some mechanism to preserve state (i.e. state of a webpage, a control or an object etc.) between subsequent requests to server from one or more clients. and this mechanism is referred as State Management in ASP.net

You can read more detailed article on state management [**here**](http://www.csharpstar.com/state-management-in-asp-net/).

### 18.What are the State Management Techniques used in ASP.NET?

State Management techniques used in ASP.NET can be categorized in two types:

1. Client-Side State Management
   * View State
   * Control State
   * Hidden Fields
   * Cookies
   * Query String
2. Server-Side State Management
   * Application State
   * Session State
   * Profile Properties
   * Cache

You can read more detailed article on state management [**here**](http://www.csharpstar.com/state-management-in-asp-net/).

### 20.What is the difference between Application and Session State?

Application state is basically a common data repository for an application’s all users and their all sessions. On the other hand, Session state is specific to a single user session. You can read more detailed explanation [**here**](http://www.csharpstar.com/state-management-in-asp-net/).

### 21.What is the difference between Session.Clear() and Session.Abandon() in ASP.NET?

Session is a Collection and it stores data as Key/Value pair.

Session.Clear() clears all the session values but doesn’t destroy the Session.

Session.Abandon() destroys the session object.

In otherwords ,Session.Clear() is like deleting all files inside a folder (say “Root”) but Session.Abandon() means deleting the “Root” folder.

### 24.What is the difference between var and dynamic?

|  |  |
| --- | --- |
| **var** | **dynamic** |
| Introduced in C# 3.0 | Introduced in C# 4.0 |
| Statically typed – This means the type of variable declared is decided by the compilerat compile time. | Dynamically typed – This means the type of variable declared is decided by the compiler at run time. |
| var type of variables are required to be initialized at the time of declaration or else they encounter the compile time error: Implicitly-typed local variables must be initialized. | No need to initialize at the time of declaration. |
| e.g., var str=”I am a string”; | e.g., dynamic str; |
| Looking at the value assigned to the variable str, the compiler will treat the variable str as string. | str=”I am a string”; //Works fine and compiles |
| Errors are caught at compile time. | Errors are caught at runtime |
| Since the compiler knows about the type and the methods and properties of the type at the compile time itself | Since the compiler comes to about the type and the methods and properties of the type at the run time. |
| Intellisense help is available for the var type of variables. This is because, its type is inferred by the compiler from the type of value it is assigned and as a result, the compiler has all the information related to the type | Intellisense help is not available for dynamic type of variables since their type is unknown until run time. So intellisense help is not available. Even if you are informed by the compiler as “This operation will be resolved at run-time”. |
| will  throw a compile error since the variable is not initialized. The compiler needs that this variable should be initialized so that it can infera type from the value. | Will compile |

### 2. What is the difference between Static class and Singleton instance?

1. A singleton classes allowed to create a only single instance or particular class. That instance can be treated as normal object. You can pass that object to a method as parameter or you can call the class method with that Singleton object. While static class can have only static methods and you can not pass static class as parameter.
2. We can implement the interfaces with the Singleton class while we can not implement the interfaces with static classes.
3. We can clone the object of Singleton classes we can not clone the object of static classes.
4. Singleton objects stored on heap while static class stored in stack.
5. A Singleton class can extend the classes(support inheritance) while static class can not inherit classes.
6. Singleton class can initialize lazy way while static class initialize when it loaded first
7. Static classes are sealed class while Single ton classes are not sealed.

### 4. Can you serialize hashtable and Why?

No, You can’t Serialize Hash table.Because, the .NET Framework does not allow serialization of any object that implements the IDictionary interface

### 6. Why singleton pattern is considered an Anti-pattern ?

– Singletons aren’t easy to handle with unit tests. You can’t control their instantiation and they may retain state across invocations.  
– Memory allocated to an Singleton can’t be freed.  
– In multithreaded environment, access to the singleton object may have to be guarded (e.g. via synchronization).  
– Singletons promote tight coupling between classes, so it is hard to test

### 7. How to use Nullable Types in .NET?

s you know, a value type cannot be assigned a null value. For example, *int i = null* will give you a compile time error.

C# 2.0 introduced nullable types that allow you to assign null to value type variables. You can declare nullable types using Nullable<t> where T is a type

int? i = null;

double? D = null;

### Points to Remember :

1. Nullable<T> type allows assignment of null to value types.
2. **?** operator is a shorthand syntax for Nullable types.
3. Use value property to get the value of nullable type.
4. Use *HasValue* property to check whether value is assigned to *nullable type* or not.

### 8. What are extension methods and where can we use them?

Extension methods enables you to add new capabilities to an existing type. You don’t need to make any modifications to the existing type, just bring the extension method into scope and you can call it like a regular instance method.  
Extension methods need to be declared in a nongeneric, non-nested, static class.

#### **Notes:**

* The difference between a regular static method and an extension method is the special this keyword for the first argument.
* Extension method cannot be declared on a class or struct.
* It can also be declared on an interface (such as IEnumerable). Normally, an interface wouldn’t have any implementation. With extension methods, however, you can add methods that will be available on every concrete implementation of the interface
* Language Integrated Query (LINQ) is one of the best examples of how you can use this technique to enhance existing code.

You can read the implementation of Extension Methods in C# [here](http://www.csharpstar.com/extension-methods-csharp/).

### 9. How to implement singleton design pattern in C#?

There are several ways for implementing Singleton in C#.

* Standard Implementation
* Double checked locking
* Early Instance Creation
* Fully Lazy Instantiation
* Using Generics
* using Lazy<T> type

You can read each way of Singleton Implementation [here](http://www.csharpstar.com/singleton-design-pattern-csharp/).

### 10.What is difference between the “throw” and “throw ex” in .NET?

Throw: In Throw, the original exception stack trace will be retained.

To keep the original stack trace information, the correct syntax is 'throw' without specifying an exception.

Eg: try { // do some operation that can fail } catch (Exception ex) { // do some local cleanup throw; }

Throw ex:In Throw ex, the original stack trace information will get override and you will lose the original exception stack trace.

I.e. 'throw ex' resets the stack trace.Eg: try { // do some operation that can fail } catch (Exception ex) { // do some local cleanup throw ex; } }

<http://www.geekinterview.com/Interview-Questions/Microsoft/ASP-NET>

# Difference Between Dictionary And Hashtable In C#

**Dictionary**

1. Dictionary is generic type Dictionary<TKey,TValue>
2. Dictionary class is a strong type < TKey,TValue > Hence, you must specify the data types for key and value.
3. There is no need of boxing/unboxing.
4. When you try to access non existing key dictionary, it gives runtime error.
5. Dictionary maintains an order of the stored values.
6. There is no need of boxing/unboxing, so it is faster than Hashtable.

**Hashtable**

1. Hashtable is non-generic type.
2. Hashtable is a weakly typed data structure, so you can add keys and values of any object type.
3. Values need to have boxing/unboxing.
4. When you try to access non existing key Hashtable, it gives null values.
5. Hashtable never maintains an order of the stored values.
6. Hashtable needs boxing/unboxing, so it is slower than Dictionary.

## Difference between Hash Table and Arraylist?

1. Arraylist is index based collectionas hashtable is key value based collection 2. you can use index for get perticular element in arraylistas you have to use key for search element in hashtable. 3. both have to explicitly typecast at the time of using it

# Difference between Array and ArrayList in C#:

The following table lists the difference between Array and ArrayList in C#.

|  |  |
| --- | --- |
| **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. |

Visit [Array](http://www.tutorialsteacher.com/csharp/array-csharp) or [ArrayList](http://www.tutorialsteacher.com/csharp/csharp-arraylist) of C# tutorials for more information.

## [Difference between Static Constructor and Private Constructor](http://helpingdotnet.blogspot.in/2012/09/difference-between-static-constructor.html)

Posted by Rahul Kharde at 4:53 AM

|  |  |
| --- | --- |
| **Static Constructor** | **Private Constructor** |
| The static constructor will only be executed once. | The private constructor will be executed each time it is called. |
| The static constructor cannot have parameters. | The private Constructor may have parameters |
| A static constructor is called before the first instance is created. So it’s kind of global initialize. | Private constructor is called after the instance of the class is created. Inheritance-wise both are same. |
| A class can have only one static constructor | A class can have multiple private constructors |
| public static class Class1  {  static Class1()  {  }  } | public class Class2  {  private Class2()  {  }  } |

**SEALED CLASS**

sealed classes are used to restrict the inheritance feature of object oriented programming. Once a class is defined as a **sealed class,** the class cannot be inherited

1. A class, which restricts inheritance for security reason is declared, sealed class.

2. Sealed class is the last class in the hierarchy.

3. Sealed class can be a derived class but can't be a base class.

4. A sealed class cannot also be an abstract class. Because abstract class has to provide functionality and here we are

restricting it to inherit.

# Static Classes

The following list provides the main features of a static class:

* Contains only static members.
* Cannot be instantiated.
* Is sealed.
* Cannot contain [Instance Constructors](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/instance-constructors).

Creating a static class is therefore basically the same as creating a class that contains only static members and a private constructor.

A private constructor prevents the class from being instantiated. The advantage of using a static class is that the compiler can check to make sure that no instance members are accidentally added. The compiler will guarantee that instances of this class cannot be created.

Static classes are sealed and therefore cannot be inherited.

They cannot inherit from any class except [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).

Static classes cannot contain an instance constructor; however, they can contain a static constructor. Non-static classes should also define a static constructor if the class contains static members that require non-trivial initialization

**SELF JOIN**

**You use a self join when a table references data in itself.**

**E.g., an Employee table may have a SupervisorID column that points to the employee that is the boss of the current employee.**

**To query the data and get information for both people in one row, you could self join like this:**

**select e1.EmployeeID, e1.FirstName, e1.LastName, e1.SupervisorID, e2.FirstName as SupervisorFirstName, e2.LastName as SupervisorLastName from Employee e1 left outer join Employee e2 on e1.SupervisorID = e2.EmployeeID**

**Diff b/w override and interface**

A virtual method has an implementation in the base class that can optionally be overridden, while an abstract method hasn't and *must* be overridden in a child class.

Otherwise they are the same.

Choosing between them depends on the situation. If you got a base implementation, you use virtual.

If you don't, and you need every descendant to implement it for itself, you choose abstract

# Lazy Loading vs Eager Loading

In Entity Framework, it is very normal to have entities that are related to each other.

For example, we may have a User table that contains basic user information like UserId, Username, Password, emailaddress etc. and another table UserDetails that contains contact details of the same user with attributes like Id, UserId (Foreign key from Users table), Contact Address, State, City etc. So here, the user table has a one-to-many relation with the UserDetails table.

Entity Framework provides the ability to load the data of a parent entity as well as its related child entity at the same time,

in other words when we load the data of the user table, we also get the related data of each User from the UserDetails table. This is known as Eager Loading.

On the other hand, if we do not want to load the related entity data at the same time as when the main entity is fetched, we use Lazy Loading

**Important points of private constructor**

- One use of private construct is when we have only static member.

- Once we provide a constructor that is either private or public or any, the compiler will not allow us to add public constructor without parameters to the class.

- If we want to create object of class even if we have private constructors then we need to have public constructor along with private constructor