[**SQL Server**](http://silverlightvsaspnet.blogspot.com/2010/02/sql-server.html)

• **SQL: Queries**

**1. Write a query to select the second highest salary from a table.**

Answer: SELECT max(salary) AS salary2 FROM orders WHERE salary < (SELECT max(salary) AS salary1 FROM orders)

**2. Write a query to select the 5th highest salary from a table.**

Answer: SELECT min(salary) AS high5 FROM employee WHERE salary IN(SELECT DISTINCT TOP 5 salary FROM orders ORDER BY salary DESC)

**3. How to find duplicate records with the number they are duplicated?**

Answer: SELECT Id, count (\*) as number\_records from table group by id having count (\*) > 1.

**• SQL: Questions**

**1. What is the difference between Delete and Truncate command in SQL?**

Answer: Delete command and truncate command both will delete the data, however the truncate command can not be rolled back as delete can be. The delete command can be used for selected records using the where clause but with the truncate command we have to loose data. DELETE statement is a logged operation and hence takes more time then truncate.

**2. What is Magic Table in SQL?**

Answer: The insert and Delete commands are known as magic tables in SQL.

**3. Can Primary key is a Foreign Key on the same table?**

Answer: Yes, Consider a category table in a e-commerce web site.
Category\_Id, Category\_Name, Parent\_Category\_ID. In this table all the parent categories are also categories. When we create a self join category id will be treated as foreign key to the same table.

 **4. What is Normalization? What are it’s rules?**

Answer: Normalisation is the technique in the database design wher ethe idea is to reduce the redundancy of non key data items across the table.
♣ Rule 1: There should be a one-to-one relationship between the instances of an entity and the rows of the table.
♣ Rule 2: A field should have the same meaning in each row of the table.
♣ Rule 3: Each table should represent at most one entity.
♣ Rule 4: Multiple instances of an entity should be represented by multiple rows in a table.
♣ Rule 5: Joins should be based only on primary and foreign-key equality.
♣ Rule 6: Make sure keys are linked correctly.

**5. What are the advantages and disadvantages of Normalization?**

Answer: There are several advantages of normalization as under:
♣ Faster sorting and index creation.
♣ A larger number of clustered indexes.
♣ Narrower and more compact indexes.
♣ Fewer indexes per table, which improves the performance of INSERT, UPDATE, and DELETE statements
♣ Fewer null values and less opportunity for inconsistency, which increase database compactness.

Beside the above benefits there are few disadvantages as well:
♣ Increased amount of Normalization increases the amount of complexity of joins between tables and that hinders the performance.

**6. What are the conditions to achieve the normalization?**

Answer: There are few conditions to achieve the normalization:
♣ There should be a unique row identifier.
♣ A table should store only data for a single type of entity. For e.g. details for book’s publisher and book’s author should be saved under different table.
♣ A table should avoid columns which can be null-able.
♣ A table should avoid duplication of data and columns.

**7. What is a Stored Procedure? State its advantage.**

Answer: A stored procedure is a set of pre-compiled SQL commands (query statements), which are stored in the server. It is faster then the loose SQL statements processed on client, as it is pre-compiled. It can execute more then one SQL commands once as they are bundled in a single entity. We can use control statements within the stored procedure, which will allow us to repeat some SQL command. It can send return values depending upon the result. Stored procedures are used to reduce network traffic.

**8. What is a Trigger?**

Answer: Triggers are a special type of stored procedure, which gets invoked upon a certain event. They can be performed upon an INSERT, UPDATE and DELETE.

**9. What is a Clustered Index?**

Answer: The data rows are stored in order based on the clustered index key. Data stored is in a sequence of the index. In a clustered index, the physical order of the rows in the table is the same as the logical (indexed) order of the key values. A table can contain only one clustered index. A clustered index usually provides faster access to data than does a non-clustered index

**10. What is a Non-Clustered Index?**

Answer: The data rows are not stored in any particular order, and there is no particular order to the sequence of the data pages. In a clustered index, the physical order of the rows in the table is not same as the logical (indexed) order of the key values.

**11. Describe the three levels of data abstraction?**

The are three levels of abstraction:
♣ Physical level: The lowest level of abstraction describes how data are stored.
♣ Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.
♣ View level: The highest level of abstraction describes only part of entire database

**12. What is DDL (Data Definition Language)?**

Answer: A data base schema which is specified by a set of definitions expressed by a special language is called DDL. Data Definition Language (DDL) is used to define and manage all the objects in an SQL database.

**13. What is DML?**

Answer: It is a special language used to manipulate the Data. Data Manipulation Language (DML), which is used to select, insert, update, and delete data in the objects defined using DDL.

**14. What is a PRIMARY KEY?**

Answer: The PRIMARY KEY is the column(s) used to uniquely identify each row of a table.

**15. What is a FOREIGN KEY?**

Answer: A FOREIGN KEY is one or more columns whose values are based on the PRIMARY or CANDITATE KEY values from the database.

**16. What is a UNIQUE KEY?**

Answer: A UNIQUE KEY is one or more columns that must be unique for each row of the table.

**17. What is the difference between UNIQUE and PRIMARY KEY?**

Answer: The UNIQUE KEY column restricts entry of duplicate values but entry of NULL value is allowed. In case of PRIMARY KEY columns entry of duplicate as well as value is also restricted.

**18. What is a VIEW?**

Answer: A View is a database object that is a logical representation of a table. It is derived from a table but has no storage space of its own and often may be used in the same manner as a table.

**19. What is a ROWID?**

Answer: ROWID is the logical address of a row, and it is unique within the database.

**20. What is INDEX?**

Answer: INDEX is a general term for an SQL feature used primarily to speed up execution and impose UNIQUENESS upon data. You can use an index to gain fast access to specific information in a database table. An index is a structure that orders the values of one or more columns in a database table. The index provides pointers to the data values stored in specified columns of the table, and then orders those pointers according to the sort order you specify.

**21. What is a cursor?**

Answer: An entity that maps over a result set and establishes a position on a single row within the result set. After the cursor is positioned on a row, operations can be performed on that row, or on a block of rows starting at that position. The most common operation is to fetch (retrieve) the current row or block of rows.

**22. The Difference between 'Count' and 'Count (\*)'?**

Answer: 'Count': Counts the number of non-null values. 'Count (\*)': Counts the number of rows in the table, including null values and duplicates.

at [10:09 AM](http://silverlightvsaspnet.blogspot.com/2010/02/sql-server.html)

Labels: [SQL Server](http://silverlightvsaspnet.blogspot.com/search/label/SQL%20Server)

**[C# Interview Questions](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_9113.html)**

**1. What is a class?**

Answer: Class is concrete representation of an entity. It represents a group of objects, which hold similar attributes and behavior. It provides Abstraction and Encapsulations.

**2. What is an Object? What is Object Oriented Programming?**

Answer: Object represents/resembles a Physical/real entity. An object is simply something you can give a name. Object Oriented Programming is a Style of programming that represents a program as a system of objects and enables code-reuse.

**3. What is Encapsulation?**

Encapsulation is binding of attributes and behaviors. Hiding the actual implementation and exposing the functionality of any object. Encapsulation is the first step towards OOPS, is the procedure of covering up of data and functions into a single unit (called class). Its main aim is to protect the data from out side world.

**4. What is Abstraction?**

Answer: Hiding the complexity. It is a process of defining communication interface for the functionality and hiding rest of the things.

**5. What is Overloading?**

Answer: Adding a new method with the same name in same/derived class but with different number/types of parameters. It implements Polymorphism.

**6. What is Overloading?**

Answer: A process of creating different implementation of a method having a same name as base class, in a derived class. It implements Inheritance.

**7. What is Shadowing?**

Answer: When the method is defined as Final/sealed in base class and not override able and we need to provide different implementation for the same. This process is known as shadowing, uses shadows/new keyword.

**8. What is Inheritance?**

Answer: It is a process of acquiring attributes and behaviors from another object (normally a class or interface).

**9. What is an Abstract class?**

Answer: An abstract class is a special kind of class that cannot be instantiated. It normally contains one or more abstract methods or abstract properties. It provides body to a class.

**10. What is an Interface?**

Answer: An interface has no implementation; it only has the signature or in other words, just the definition of the methods without the body.

**11. What is Polymorphism?**

Answer: Mean by more than one form. Ability to provide different implementation based on different number/type of parameters.

**12. What is Pure-Polymorphism?**

Answer: When a method is declared as abstract/virtual method in a base class and which is overridden in a base class. If we create a variable of a type of a base class and assign an object of a derived class to it, it will be decided at a run time, which implementation of a method is to be called.
This is known as Pure-Polymorphism or Late-Binding.

**13. What is a Constructor?**

Answer: A special Method Always called whenever an instance of the class is created.

**14. What is a Destructor?**

Answer: A special method called by GC. just before object is being reclaimed by GC.

**15. How a base class method is hidden?**

Answer: Hiding a base class method by declaring a method in derived class with keyword new. This will override the base class method and old method will be suppressed.

**16. What Command is used to implement properties in C#?**

Answer: get & set access modifiers are used to implement properties in c#.

**17. What is method overloading?**

Answer: Method overloading is having methods with same name but carrying different signature, this is useful when you want a method to behave differently depending upon a data passed to it.

**18. Can constructors have parameters?**

**19. What are Static Assembly and Dynamic Assembly?**

Answer: Static assemblies can include .NET Framework types (interfaces and classes) as well as resources for the assembly (bitmaps, JPEG files, resource files, and so forth). Static assemblies are stored on disk. Dynamic assemblies run directly from memory and are not saved to disk before execution.

**20. Describe the functionality of an assembly.**

Answer: It is the smallest unit that has version control. All types and resources in the same assembly are versioned as a unit and support side by side execution. Assemblies contain the metadata and other identities which allow the common language runtime to execute. They are the boundaries providing the type check. They the unit where security permissions are requested and granted.

**21. What is serialization?**

Answer: Serialization is the process of converting an object into a stream of bytes. De-serialization is the opposite process of creating an object from a stream of bytes. Serialization/De-serialization is mostly used to transport objects (e.g. during remoting), or to persist objects (e.g. to a file or database). There are two separate mechanisms provided by the .NET class library for serialization - XmlSerializer and SoapFormatter and BinaryFormatter. Microsoft uses XmlSerializer for Web Services, and uses SoapFormatter/BinaryFormatter for remoting.

at [10:07 AM](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_9113.html)

Labels: [c#](http://silverlightvsaspnet.blogspot.com/search/label/c%23)

**[ASP.Net Interview Questions](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_823.html)**

**1. If I’m developing an application that must accommodate multiple security levels though secure login and my ASP.NET web application is spanned across three web-servers (using round-robin load balancing) what would be the best approach to maintain login-in state for the users?**

Answer: You can use the security state maintained using a database. (Use Authentication mode as database)

**2. What’s the difference between Codebehind="MyCode.aspx.cs" and Src="MyCode.aspx.cs"?**

Answer: Visual Studio uses the Codebehind attribute to distinguish the page source or programming logic from the design. Also the src attribute will make the page compile on every request. That is the page will not be compiled in advance and stored in the bin as a dll instead it will be compiled at run time.

**3. Suppose you want a certain ASP.NET function executed on MouseOver over a certain button or textbox. Where do you add an event handler?**

Answer: Every web control has a ability to add the attributes on client side which will execute on client side and run a client side script like a javascript function.

btnSubmit.Attributes.Add(“onMouseOver”,”someClientCode();”) //For on mouse over of a button
TextBox1.Attributes.Add(“onFocus”,“ClearText();”) //For on focus of a text box

**4. Explain what a diffgram is and a good use for one?**

Answer: The DiffGram is one of the two XML formats that you can use to render DataSet object contents to XML. For reading database data to an XML file to be sent to a Web Service.

**5. What base class do all Web Forms inherit from?**

Answer: The Page class.

**6. Name two properties common in every validation control?**

Answer: ControlToValidate and Text property.

**7. What tags do you need to add within the Datagrid tags to bind columns manually?**

Answer: Set AutoGenerateColumns Property to false on the Datagrid tag

**8. What tag do you use to add a hyperlink column to the DataGrid?**

Answer: asp:HyperLinkColumn

**9. What is the transport protocol you use to call a Web service?**

Answer: SOAP (Simple Object Access Protocol) is the preferred protocol.

**10. Where on the Internet would you look for Web services?**

Answer: http://www.uddi.org

**11. Which control would you use if you needed to make sure the values in two different controls matched?**

Answer: CompareValidator Control

**12. What are the assembly entry points? An Assembly can have how many entry points at a time?**

Answer: An assembly can have only one entry point from DllMain, WinMain or Main.

**13. What does an assembly contain?**

Answer:
• Manifest - The metadata describing the information below.
• Assembly name - Aids in versioning and visibility scope.
• Version information - The version number is integrated into the assembly's identity.
• Types - Boundaries and scopes of methods, classes, properties, events, attributes.
• Locale - Information describing language/culture.
• Cryptographic Hash - Public key encoded hash acting as version/security check.
• Security Permissions - The permissions within the assembly determine the permissions that can be granted for all aspects of the assembly contents.

**14. What does an assembly manifest contains?**

Answer: It contains assembly name, version number (major.minor.build.revision) and culture Information. It also specifies the strong name information, which is useful for shared assemblies, and list of files, an assembly contains. It also provides information for type references in an assembly and other referenced assemblies.

 **15. Which tool is used to deploy an assembly, so as the assembly is available to all the application?**

Answer: The GacUtil.exe is the tool, which allows us to add any assembly to the windows GAC.

**16. How many catch statements can be associated with single try statement?**

Answer: There can be a zero or more catch statement for each try statement. So it has not limit to the number of catch statement per try statement.

**17. What is Console and System a Class/a Data Member/a routine/a namespace or a type?**

Answer: Console is a class and System is namespace.

**18. How many values can be returned from a method in C#?**

Answer: Only one value can be returned from method, however you can use ref or out variable to change more than one value in called method.

**19. How to declare a variable named this in C#, with data type string?**

Answer: string @this;

**20. Can we change the dimension of Array at run time like Array [3, 4]?**

Answer: Yes, We can change only the first position of array dimension.

**21. What keyword is used to accept a variable number of parameter in a method?**

Answer: “params” keyword is used as to accept variable number of parameters.

**22. What is a Namespace? What is the use of a namespace?**

Answer: Namespaces are logical grouping of classes and other types in hierarchical structure. Namespaces are useful to avoid collision or ambiguity among the classes and type names. Another use of the namespace is to arrange a group of classes for a specific purpose.

**23. What does a keyword using works for?**

Answer: Using is just a convention or a short-cut method which allows us to access the classes in a namespace by referencing it once. So when ever we want use the classes or methods in them, we can avoid typing the entire namespace hierarchy. However it is not a good practice when there are likely chances to have name ambiguity or collision of class names.

**24. What is Enums in C#?**

Answer: Enums or Enumerators are used to declare a set of related constants (default start with 0); they are only available with primitive data types like int and short etc.

**25. What is Delegates?**

Answer: Delegates are a type-safe, object-oriented implementation of function pointers and are used in many situations where a component needs to call back to the component that is using it. Delegates are generally used as basis of events, which allow any delegate to easily be registered for as event.

**26. Which are the namespaces that are imported automatically by Visual Studio in ASP.Net?**

Answer: There are 7 namespaces which are imported automatically.
♣ System
♣ System.Collections
♣ System.IO
♣ System.web
♣ System.web.UI
♣ System.web.UI.HTMLControls
♣ System.web.UI.WebControls
27. Which namespaces are used for data access?
♣ System.Data
♣ System.Data.OleDB
♣ System.Data.SQLClient

**28. What do you mean by boxing and un-boxing?**

Answer: C# provides us with Value types and Reference Types. Value Types are stored on the stack and Reference types are stored on the heap. The conversion of value type to reference type is known as boxing and converting reference type back to the value type is known as un-boxing.
e.g.
int x = 10;

object o = x ; // Implicit boxing
object o = (object) x; // Explicit Boxing

x = o; // Implicit Un-Boxing
x = (int)o; // Explicit Un-Boxing

**29. What are the different methods available under sqlcommand class to access the data?**

♣ ExecuteReader –Used where one or more records are returned – SELECT Query.
♣ ExecuteNonQuery – Used where it affects a state of the table and no data is being queried - INSERT, UPDATE, DELETE, CREATE and SET queries.
♣ ExecuteScalar – Used where it returns a single record(a single value normally) – SQL Functions like MIN(), NAX()

**30. What are the different types of Session state management options available with ASP.NET?**

Answer: ASP.NET provides In-Process & Out-of-Process state management,
Also known as "In-Proc" and "Out-Proc". In-Proc stores the session in memory of the web server, that is on the same server the ASP.Net page is.
On the other hand Out-Proc session state management stores the session data on external data source, which can be a SQL Server or Server State Service. Out-of-Process state management requires the objects stored in session, must be serializable.

**31. What is Remoting? Give Example.**

Answer: Remoting is a means by which one operating system process, or program, can communicate with another process. The two processes can exist on the same computer or on two computers connected by a LAN or the Internet. Web services are probably the best known type of remoting, but they are not the only option.

**32. What is Marshalling?**

Answer: Marshaling is a process of making an object in one process (the server) available to another process (the client). There are two ways to achieve the marshalling.
i. Marshal by value: the server creates a copy of the object passes the copy to the client. When a client makes a call to an object marshaled by value (MBV), the server creates an exact copy and sends that copy to the client. The client can then use the object's data and executable functionality directly within its own process or application domain without making additional calls to the server. Objects that the application accesses frequently are best remoted using MBV.
ii. Marshal by reference: the client creates a proxy for the object and then uses the proxy to access the object. When a client makes a call to an object marshaled by reference (MBR), the .NET framework creates a proxy in the client's application domain and the client uses that proxy to access the original object on the server. Large objects that the application accesses relatively infrequently are good candidates for MBR.

**33. What is a Static class? What are its features?**

Answer: Static class is a class which can be used or accessed without creating an instance of the class.
Important Features:
i. Static class only contains static members and a private constructor.
ii. Static class cannot be instantiated.
iii. Static classes are sealed by default and therefore cannot be inherited.
34. What is sealed class? What are its features?
Answer: Sealed classes are those classes which can not be inherited and thus any sealed class member can not be derived in any other class. A sealed class cannot also be an abstract class.
In C# structs are implicitly sealed; therefore, they cannot be inherited.

**35. Can we declare a method as sealed?**

Answer: In C# a method can't be declared as sealed. However when we override a method in a derived class, we can declare the overridden method as sealed. By declaring it as sealed, we can avoid further overriding of this method.
E.g.
using System;
class MyClass1
{
public int x;
public int y;
public virtual void Method() {
Console.WriteLine("virtual method"); }
}
class MyClass : MyClass1
{
public override sealed void Method() {
Console.WriteLine("sealed method"); }
}
class MainClass
{ public static void Main() {
MyClass1 mC = new MyClass();
mC.x = 110;
mC.y = 150;
Console.WriteLine("x = {0}, y = {1}", mC.x, mC.y);
mC.Method(); }
}

**36. What is a DataSet ?**

Answer: A DataSet is an in memory representation of data loaded from any data source.

**37. What is a DataTable?**

Answer: A DataTable is a class in .NET Framework and in simple words a DataTable object represents a table from a database.

**38. If you want to view an Assembly how to you go about it? What is ILDASM?**

Answer: You can use the MSIL Disassembler (Ildasm.exe) to view Microsoft intermediate language (MSIL) information in a file. If the file being examined is an assembly, this information can include the assembly's attributes, as well as references to other modules and assemblies. This information can be helpful in determining whether a file is an assembly or part of an assembly, and whether the file has references to other modules or assemblies.

**39. Where is version information stored of an assembly?**

Answer: The version number is stored in the assembly manifest along with other identity information, including the assembly name and public key, as well as information on relationships and identities of other assemblies connected with the application.

**40. Is versioning applicable to private assemblies?**

Answer: No

**41. How to create a shared assembly or add an assembly to GAC?**

Answer: There are several ways an assembly can be added to GAC.
i. Use .msi installer designed to work with the global assembly cache.
ii. Use GACUtil.exe provided by the .NET Framework SDK.
iii. Use Windows Explorer to drag assemblies into the cache.

**43. What is reflection?**

Answer: All .NET compilers produce metadata about the types defined in the modules they produce. This metadata is packaged along with the module (modules in turn are packaged together in assemblies), and can be accessed by a mechanism called reflection. The System.Reflection namespace contains classes that can be used to interrogate the types for a module/assembly. Reflection is ability to find information about types contained in an assembly at run time.

**44. How can I produce an assembly?**

Answer: Simply compile your class/module with the following command.
C#.Net - CSC /t:library yourclassname.cs
VB.Net - VBC /t:library yourmodulename.vb

**45. What is an Application Domain? How they get created?**

Answer: An Application Domain can be thought of as a lightweight processes controlled by the .Net runtime. Application Domains are usually created by hosts like Windows Shell, ASP.NET and IE. When you run a .NET application from the command-line, the host is the Shell. The Shell creates a new Application Domain for every application.

**46. Do I have any control over the garbage collection algorithm?**

Answer: Yes, we have a limited control over the GC algorithm, For example, the System.GC class exposes a Collect method - this forces the garbage collector to collect all unreferenced objects immediately.

**47. What is a life span of a static variable?**

Answer: A static variable’s life span is till the class is in memory.

**48. What is a Page Life Cycle of an ASP.Net page?**

Answer: There are various stages described as under.
♣ Init
♣ LoadViewState
♣ LoadPostBackData
♣ Load
♣ RaisePostBackDataChangedEvent
♣ RaisePostBackEvents
♣ Pre-Render
♣ SaveViewState
♣ Render
♣ Unload

**49. Can the action attribute of a server-side form tag be set to a value and if not how can you possibly pass data from a form to a subsequent Page?**

Answer: No, Assigning value will not work because will be overwritten at the time of rendering. We can assign value to it by register a startup script which will set the action value of form on client-side. On other hand we can use Server.Transfer or Response.Redirect.

**50. How do you turn off cookies in one page of your asp.net application?**

Answer: We may not use them at the max, However to allow the cookies or not, is client side functionality.

**51. Which method do you use to redirect to user to another page without performing a round trip to Client?**

Answer: Server.Transfer(“AnotherPage.aspx”).

**52. How many namespaces are in .NET version 1.1?**

Answer: 124.

**53. Should Validation occur on Client/Server Side for Date Input?**

Answer: Both. Client-side reduces extra round-trip. Server-Side ensures prevention against hacking and failure against automated requests.

**54. What are the web form events?**

Answer:
i. The first event that occurs in the life of a Web Form is the Init event. This is raised so that we can have initialization code for the page. The controls on the page are not yet created at this point. This event is raised once for each user of the page.
ii. The Load event follows the Init event. Subsequently, it is raised each time the page is requested. When this event is raised, all child controls of the Web Form are loaded and accessible. You should be able to retrieve data and populate the controls so that they can render themselves on the page when sent back to the client.
iii. The PreRender event happens just before the page is rendered and sent back to the client. We don't often handle this event; however, it depends on the situation.
iv. The last event in the life of a Web Form is the Unload event. This happens when the page is unloaded from memory. Final cleanup should be done here.

**55. Which control would you use if you needed to make sure the values in two different controls matched?**

Use the CompareValidator control to compare the values of 2 different controls.

**56. True or False: To test a Web service you must create a windows application or Web application to consume this service?**

False.

**57. How many classes can a single .NET DLL contain?**

Unlimited.

at [10:06 AM](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_823.html)

Labels: [ASP.Net](http://silverlightvsaspnet.blogspot.com/search/label/ASP.Net)

**[ASP.Net Interview Questions](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_7017.html)**

**• Differences**

**1. Difference between Classic ASP and ASP.Net?**

Answer:
♣ ASP is Interpreted language based on scripting languages like Jscript or VBScript.
♣ ASP has Mixed HTML and coding logic.
♣ Limited development and debugging tools available.
♣ Limited OOPS support.
♣ Limited session and application state management.
♣ Poor Error handling system.
♣ No in-built support for XML.
♣ No fully distributed data source support.
while
♣ ASP.Net is supported by compiler and has compiled language support.
♣ Separate code and design logic possible.
♣ Variety of compilers and tools available including the Visual studio.Net.
♣ Completely Object Oriented.
♣ Complete session and application state management.
♣ Full proof error handling possible.
♣ Full XML Support for easy data exchange.
♣ Fully distributed data source support.

**2. What’s the difference between Response.Write() and Response.Output.Write()?**

Answer: Response.Outout.Write allows us to write the formatted out put.

**3. Can you explain the difference between an ADO.NET Dataset and an ADO Recordset?**

Answer:
♣ A DataSet can represent an entire relational database in memory, complete with tables, relations, and views, A Recordset can not.
♣ A DataSet is designed to work without any continuing connection to the original data source; Recordset maintains the contentious connection with the original data source.
♣ There's no concept of cursor types in a DataSet, They are bulk loaded, while Recordset work with cursors and they are loaded on demand.
♣ DataSets have no current record pointer, you can use For Each loops to move through the data. Recordsets have pointers to move through them.
♣ You can store many edits in a DataSet, and write them to the original data source in a single operation. Recordset can have a single edit at a time.
♣ Dataset can fetch source data from many tables at a time, for Recordset you can achieve the same only using the SQL joins.

**4. What is the difference between an abstract method & virtual method?**

Answer: An Abstract method does not provide an implementation and forces overriding to the deriving class (unless the deriving class also an abstract class), where as the virtual method has an implementation and leaves an option to override the it in the deriving class. Thus Virtual method has an implementation & provides the derived class with the option of overriding it. Abstract method does not provide an implementation & forces the derived class to override the method.

**5. What are the different types of assemblies? Explain.**

Answer: Assemblies can be static or dynamic. Static assemblies can include .NET Framework types (interfaces and classes), as well as resources for the assembly (bitmaps, JPEG files, resource files, and so on). Static assemblies are stored on disk in portable executable (PE) files. You can also use the .NET Framework to create dynamic assemblies, which are run directly from memory and are not saved to disk before execution. You can save dynamic assemblies to disk after they have executed.

**6. What are the difference between Structure and Class?**

Answer:
♣ Structures are value type and Classes are reference type.
♣ Structures can not have contractors or destructors. Classes can have both contractors and destructors.
♣ Structures do not support Inheritance, while Classes support Inheritance.

**7. What are the difference between const and readonly?**

Answer:
♣ A const can not be static, while readonly can be static.
♣ A const need to be declared and initialized at declaration only, while a readonly can be initialized at declaration or by the code in the constructor.
♣ A const’s value is evaluated at design time, while a readonly’s value is evaluated at runtime.

**8. Differences between dataset.clone and dataset.copy**

Answer: dataset.clone copies just the structure of dataset (including all the datatables, schemas, relations and constraints.), However it doesn’t copy the data. On the other hand dataset.copy, copies both the dataset structure and the data.

**9. Describe the difference between inline and code behind.**

Answer: Inline code written along with the html and design blocks in an .aspx page. Code-behind is code written in a separate file (.cs or .vb ) and referenced by the .aspx page.

**10. What is Difference between Namespace and Assembly?**

Answer: Namespace is a logical design-time naming convenience, whereas an assembly establishes the name scope for types at run time.

**11. What is the difference between early binding and late binding?**

Answer: Calling a non-virtual method, decided at a compile time is known as early binding. Calling a virtual method (Pure Polymorphism), decided at a runtime is known as late binding.

**12. What is the difference between User Control and Custom Control?**

Answer: Custom Controls are compiled code (Dlls), easier to use, difficult to create, and can be placed in toolbox. Drag and Drop controls. Attributes can be set visually at design time. Can be used by Multiple Applications (If Shared Dlls), Even if Private can copy to bin directory of web application add reference and use. Normally designed to provide common functionality independent of consuming Application. User Controls are similar to those of ASP include files, easy to create, can not be placed in the toolbox and dragged - dropped from it. A User Control is shared among the single application files.

**13. What is the difference between ASP Session State and ASP.Net Session State?**

Answer: ASP session state relies on cookies, Serialize all requests from a client, does not survive process shutdown, Can not maintained across machines in a Web farm.

**14. What is the difference bettween DataReader and DataSet?**

Answer: DataReader represents only one database record at a time. You must call the Read() method to fetch each new record from the underlying database table into memory. Each time you call Read() again, the previously fetched record is lost. DataSet is on the other hand, enables you to represent the results of a database query in your server's memory. Because a DataSet provides you with a memory-resident representation of data, you can work with the results of a database query as a whole. DataReader must remain connected to a database table. A DataReader is tied down to its underlying data source. The DataSet object is central to supporting disconnected, and distributed data scenarios with ADO.NET. The DataSet is a memory-resident representation of data that provides a consistent relational programming model regardless of the data source. It might be helpful to think of a DataReader as a forward-only recordset. A DataSet, on the other hand, is similar to a disconnected, client-side, static recordset. DataSets also require more overhead to create and populate than DataReaders.

• Abbreviations
1. CLR = Common Language Runtime
2. CLS = Common Language Specifications
3. CTS = Common Type Specifications
4. GC = Garbage Collector.
5. WSDL = Web Services Description Language.
6. MSIL = Microsoft Intermediate Language.
7. CIL = Common Intermediate Language - MSIL.
8. JIT = Just In Time.
9. PE = Portable Executable – A file format.
10. COFF = Common Object File Format – A file format.
11. GAC = Global Assembly Cache.
12. DDL = Data Definition Language.
13. DML = Data Manipulation Language.
14. CAS = Code Access Security.
15. RCW = Runtime Callable Wrapper.
16. COM = Component Object Model.
17. CCW = COM Callable Wrapper.
18. DOM = Document Object Model.
19. DNA = Distributed interNet Applications Architecture.
20. GUID = Globally Unique Identifier.
21. MS-DTC = Microsoft Distributed Transaction Coordinator.
22. OLTP = Online Transaction Processing.
23. OLAP = Online Analytical Processing.
24. RAD = Rapid Application Development.
25. SMTP = Simple Mail Transfer Protocol.
26. SOAP = Simple Object Access Protocol.
27. TCP = Transport Control Protocol.
28. TLB = Type Library.
29. UDF = Uniform Data Format.
30. UDDI = Universal Description, Discovery and Integration.

 **1. A Web service can only be written in .NET?**

Answer: False (Java also)

**2. To test a Web service you must create a windows application or Web application to consume this service?**

Answer: False, the webservice comes with a test page and it provides HTTP-GET method to test it.

**1. What is IL? (What is MSIL or CIL, What is JIT?)**

MSIL is the CPU –independent instruction set into which .Net framework programs are compiled. It contains instructions for loading, storing initializing, and calling methods on objects.

**2. What is CLR?**

Answer: CLR also known as Common Language Run time provides a environment in which program are executed, it activate object, perform security check on them, lay them out in the memory, execute them and garbage collect them.

**3. What is CTS?**

Answer: The common type system is a rich type system, built into the common language runtime, which supports the types and operations found in most programming languages.

**4. What is CLS?**

Answer: CLS also known as Common Language Specification defines the rules which all language must support, in order to be a part of .Net framework. The Common Language Specification is a set of constructs and constraints that serves as a guide for library writers and compiler writers. It allows libraries to be fully usable from any language supporting the CLS, and for those languages to integrate with each other. The Common Language Specification is a subset of the common type system. The Common Language Specification is also important to application developers who are writing code that will be used by other developers.

**5. Which class does the web page belong to in ASP.Net?**

Answer: System.Web.UI.Page

**6. Which class deals wit the user’s locale information?**

Answer: System.Web.UI.Page.Culture

**7. What data type does the RangeValidator control support?**

Answer: Integer, Date and String.

**8. What is the difference between Server.Transfer and Response.Redirect?**

Answer: Server.Transfer, transfers the control of a web page, posting a form data, while Response.Redirect simply redirects a page to another page, it can not post a form data to another page. Server.Transfer is more efficient over the Response.Redirect, because Response.Redirect causes a round trip to server as the page is processed once again on the client and a request is made to server there after.

**9. Can you give an example of what might be best suited to place in the Application\_Start and Session\_Start subroutines?**

Answer: All the global declarations or the variables used commonly across the application can be deployed under Application\_Start. All the user specific tasks or declarations can be dealt in the Session\_Start subroutine.

**10. What is viewState?**

Answer: ViewState is a .Net mechanism to store the posted data among post backs. ViewState allows the state of objects to be stored in a hidden field on the page, saved on client side and transported back to server whenever required.

**11. What is the lifespan for items stored in ViewState?**

Answer: Items stored in a ViewState exist for the life of the current page, including the post backs on the same page.

**12. Can we disable ViewState, If, yes how?**

Answer: ViewState can be disabled by using "EnableViewState" property set to false.

**13. What’s a bubbled event?**

Answer: When a complex control like datalist or datagrid, which contains a child control, using an itemcommand can listen to the events raised by the child control in the main control. The process of listening to the child control in the main or parent control is called as event bubbling.

**14. What is an assembly?**

Answer: Assemblies are the building blocks of the .NET framework. They are the logical grouping of the functionality in a physical file.

**15. What are different types of Assemblies?**

Answer: Single file and multi file assembly. Assemblies can be static or dynamic. Private assemblies and shared assemblies

**16. Which method do you invoke on the DataAdapter control to load your generated dataset with data?**

Answer: DataAdapter’s fill () method is used to fill load the data in dataset.

**17. Can you edit data in the Repeater control?**

Answer: No, it just reads the information from its data source

**18. Which template is to be provided in the Repeater control in order to display a data? Which template will display every other row in another color?**

Answer: ItemTemplate, AlternatingItemTemplate

**19. What are the most important property and most important method of a Repeater control?**

Answer: The DataSource property and DataBind() method.

**20. How many classes can a single .NET DLL contain?**

Answer: It can contain many classes.

**21. What are the advantages of an assembly?**

Answer: Increased performance. Better code management and encapsulation. It also introduces the n-tier concepts and business logic.

**22. What is the purpose of an Assembly?**

Answer: An assembly controls many aspects of an application. The assembly handles versioning, type and class scope, security permissions, as well as other metadata including references to other assemblies and resources. The rules described in an assembly are enforced at runtime

**23. What a static assembly consist of in general?**

Answer: In general, a static assembly consist of the following four elements:
♣ Assembly Manifest, which contains the assembly metadata.
♣ Type Metadata.
♣ MSIL code that implements the types.
♣ A set of resources.
From above all only the manifest is required, however the other types and resources add the additional functionality to the assembly.

**24. What is GAC or Global Assembly Cache?**

Answer: Global Assembly Cache (GAC) is a common place to share the .NET assemblies across many applications. GAC caches all strong named assembly references within it. All System assemblies that come with the .NET framework reside in the GAC.

**25. How to view an assembly?**

Answer: We can use the tool "ildasm.exe" known as "Assembly Disassembler" to view the assembly.

**26. What is Authentication and Authorization?**

Answer: Authentication is the process of identifying users. Authentication is identifying/validating the user against the credentials (username and password) and Authorization performs after authentication. Authorization is the process of granting access to those users based on identity. Authorization allowing access of specific resource to user.

**27. What are the types of Authentication? Describe**
.
Answer: There are 3 types of Authentication. Windows, Forms and Passport Authentication.
♣ Windows authentication uses the security features integrated into the Windows NT and Windows XP operating systems to authenticate and authorize Web application users.
♣ Forms authentication allows you to create your own list/database of users and validate the identity of those users when they visit your Web site.
♣ Passport authentication uses the Microsoft centralized authentication provider to identify users. Passport provides a way to for users to use a single identity across multiple Web applications. To use Passport authentication in your Web application, you must install the Passport SDK.

**28. What are the types of comment in C#?**

Answer: There are 3 types of comments in C#.
Single line (//), Multi line (/\* \*/)and Page Comments (///).

**29. What is an ArrayList?**

Answer: The ArrayList object is a collection of items containing a single data value.

**30. What is a HashTable?**

Answer: The Hashtable object contains items in key/value pairs. The keys are used as indexes, and very quick searches can be made for values by searching through their keys.

**31. What is SortedList?**

Answer: The SortedList object contains items in key/value pairs. A SortedList object automatically sorts items in alphabetic or numeric order.

**32. What is a Literal Control?**

Answer: The Literal control is used to display text on a page. The text is programmable. This control does not let you apply styles to its content!

**33. What is CAS or Code Access Security?**

Answer: CAS is the part of the .NET security model that determines whether or not a piece of code is allowed to run, and what resources it can use when it is running. For example, it is CAS that will prevent a .NET web applet from formatting a hard disk.

**34. What is Side-by-Side Execution**

Answer: The CLR allows any versions of the same-shared DLL (shared assembly) to execute at the same time, on the same system, and even in the same process. This concept is known as side-by-side execution.

**35. What are the different types of Caching?**

Answer: There are three types of Caching:
♣ Output Caching: stores the responses from an asp.net page.
♣ Fragment Caching: Only caches/stores the portion of page (User Control)
♣ Data Caching: is Programmatic way to Cache objects for performance.

**36. What are the different types of Validation Controls?**

Answer: There are six types of validation controls available:
♣ RequiredFieldValidator
♣ RangeValidator
♣ RegularExpressionValidator
♣ CompareValidator
♣ CustomValidator
♣ ValidationSummary

**37. How to Manage State in ASP.Net?**

There are several ways to manage a state.
♣ ViewState
♣ QueryString
♣ Cookies
♣ Session
♣ Application

**38. What base class all Web Forms inherit from?**

Answer: System.Web.UI.Page.

**39. What method do you use to explicitly kill a user’s Session?**

Answer: HttpContext.Current.Session.Abandon().

**40. What are the layouts of ASP.NET Pages?**

Answer: GridLayout and FlowLayout. GridLayout positions the form object on absolute x and y co-ordinates of the screen. FlowLayout positions the form objects relative to each other.

**41. What is the Web User Control?**

Answer: Combines existing Server and HTML controls by using VS.Net. to create functional units that encapsulate some aspects of UI. Resides in Content Files, which must be included in project in which the controls are used.

**42. What is the Composite Custom Control?**

Answer: combination of existing HTML and Server Controls.

at [10:05 AM](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_7017.html)

Labels: [ASP.Net](http://silverlightvsaspnet.blogspot.com/search/label/ASP.Net)

**[ASP.Net Interview Questions](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_2987.html)**

Explain the differences between Server-side and Client-side code? Server side scripting means that all the script will be executed by the server and interpreted as needed. ASP doesn’t have some of the functionality like sockets, uploading, etc. For these you have to make a custom components usually in VB or VC++. Client side scripting means that the script will be executed immediately in the browser such as form field validation, clock, email validation, etc. Client side scripting is usually done in VBScript or JavaScript. Download time, browser compatibility, and visible code - since JavaScript and VBScript code is included in the HTML page, then anyone can see the code by viewing the page source. Also a possible security hazards for the client computer.
• What type of code (server or client) is found in a Code-Behind class? C#
• Should validation (did the user enter a real date) occur server-side or client-side? Why? Client-side validation because there is no need to request a server side date when you could obtain a date from the client machine.
• What does the "EnableViewState" property do? Why would I want it on or off? Enable ViewState turns on the automatic state management feature that enables server controls to re-populate their values on a round trip without requiring you to write any code. This feature is not free however, since the state of a control is passed to and from the server in a hidden form field. You should be aware of when ViewState is helping you and when it is not. For example, if you are binding a control to data on every round trip (as in the datagrid example in tip #4), then you do not need the control to maintain it’s view state, since you will wipe out any re-populated data in any case. ViewState is enabled for all server controls by default. To disable it, set the EnableViewState property of the control to false.
• What is the difference between Server.Transfer and Response.Redirect? Why would I choose one over the other? 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.
• Can you give an example of when it would be appropriate to use a web service as opposed to a non-serviced .NET component? When to Use Web Services:
• Communicating through a Firewall When building a distributed application with 100s/1000s of users spread over multiple locations, there is always the problem of communicating between client and server because of firewalls and proxy servers. Exposing your middle tier components as Web Services and invoking the directly from a Windows UI is a very valid option.
• Application Integration When integrating applications written in various languages and running on disparate systems. Or even applications running on the same platform that have been written by separate vendors.
• Business-to-Business Integration This is an enabler for B2B intergtation which allows one to expose vital business processes to authorized supplier and customers. An example would be exposing electronic ordering and invoicing, allowing customers to send you purchase orders and suppliers to send you invoices electronically.
• Software Reuse This takes place at multiple levels. Code Reuse at the Source code level or binary componet-based resuse. The limiting factor here is that you can reuse the code but not the data behind it. Webservice overcome this limitation. A scenario could be when you are building an app that aggregates the functionality of serveral other Applicatons. Each of these functions could be performed by individual apps, but there is value in perhaps combining the the multiple apps to present a unifiend view in a Portal or Intranet.
• When not to use Web Services: Single machine Applicatons When the apps are running on the same machine and need to communicate with each other use a native API. You also have the options of using component technologies such as COM or .NET Componets as there is very little overhead.
• Homogeneous Applications on a LAN If you have Win32 or Winforms apps that want to communicate to their server counterpart. It is much more efficient to use DCOM in the case of Win32 apps and .NET Remoting in the case of .NET Apps.
• Can you explain the difference between an ADO.NET Dataset and an ADO Recordset? In ADO, the in-memory representation of data is the recordset. In ADO.NET, it is the dataset. There are important differences between them.
• A recordset looks like a single table. If a recordset is to contain data from multiple database tables, it must use a JOIN query, which assembles the data from the various database tables into a single result table. In contrast, a dataset is a collection of one or more tables. The tables within a dataset are called data tables; specifically, they are DataTable objects. If a dataset contains data from multiple database tables, it will typically contain multiple DataTable objects. That is, each DataTable object typically corresponds to a single database table or view. In this way, a dataset can mimic the structure of the underlying database. A dataset usually also contains relationships. A relationship within a dataset is analogous to a foreign-key relationship in a database —that is, it associates rows of the tables with each other. For example, if a dataset contains a table about investors and another table about each investor’s stock purchases, it could also contain a relationship connecting each row of the investor table with the corresponding rows of the purchase table. Because the dataset can hold multiple, separate tables and maintain information about relationships between them, it can hold much richer data structures than a recordset, including self-relating tables and tables with many-to-many relationships.
• In ADO you scan sequentially through the rows of the recordset using the ADO MoveNext method. In ADO.NET, rows are represented as collections, so you can loop through a table as you would through any collection, or access particular rows via ordinal or primary key index. DataRelation objects maintain information about master and detail records and provide a method that allows you to get records related to the one you are working with. For example, starting from the row of the Investor table for "Nate Sun," you can navigate to the set of rows of the Purchase table describing his purchases. A cursor is a database element that controls record navigation, the ability to update data, and the visibility of changes made to the database by other users. ADO.NET does not have an inherent cursor object, but instead includes data classes that provide the functionality of a traditional cursor. For example, the functionality of a forward-only, read-only cursor is available in the ADO.NET DataReader object. For more information about cursor functionality, see Data Access Technologies.
• Minimized Open Connections: In ADO.NET you open connections only long enough to perform a database operation, such as a Select or Update. You can read rows into a dataset and then work with them without staying connected to the data source. In ADO the recordset can provide disconnected access, but ADO is designed primarily for connected access. There is one significant difference between disconnected processing in ADO and ADO.NET. In ADO you communicate with the database by making calls to an OLE DB provider. In ADO.NET you communicate with the database through a data adapter (an OleDbDataAdapter, SqlDataAdapter, OdbcDataAdapter, or OracleDataAdapter object), which makes calls to an OLE DB provider or the APIs provided by the underlying data source. The important difference is that in ADO.NET the data adapter allows you to control how the changes to the dataset are transmitted to the database — by optimizing for performance, performing data validation checks, or adding any other extra processing. Data adapters, data connections, data commands, and data readers are the components that make up a .NET Framework data provider. Microsoft and third-party providers can make available other .NET Framework data providers that can be integrated into Visual Studio.
• Sharing Data Between Applications. Transmitting an ADO.NET dataset between applications is much easier than transmitting an ADO disconnected recordset. To transmit an ADO disconnected recordset from one component to another, you use COM marshalling. To transmit data in ADO.NET, you use a dataset, which can transmit an XML stream.
• Richer data types.COM marshalling provides a limited set of data types — those defined by the COM standard. Because the transmission of datasets in ADO.NET is based on an XML format, there is no restriction on data types. Thus, the components sharing the dataset can use whatever rich set of data types they would ordinarily use.
• Performance. Transmitting a large ADO recordset or a large ADO.NET dataset can consume network resources; as the amount of data grows, the stress placed on the network also rises. Both ADO and ADO.NET let you minimize which data is transmitted. But ADO.NET offers another performance advantage, in that ADO.NET does not require data-type conversions. ADO, which requires COM marshalling to transmit records sets among components, does require that ADO data types be converted to COM data types.
• Penetrating Firewalls.A firewall can interfere with two components trying to transmit disconnected ADO recordsets. Remember, firewalls are typically configured to allow HTML text to pass, but to prevent system-level requests (such as COM marshalling) from passing.
• Can you give an example of what might be best suited to place in the Application\_Start and Session\_Start subroutines? The Application\_Start event is guaranteed to occur only once throughout the lifetime of the application. It’s a good place to initialize global variables. For example, you might want to retrieve a list of products from a database table and place the list in application state or the Cache object. SessionStateModule exposes both Session\_Start and Session\_End events.
• If I’m developing an application that must accomodate multiple security levels though secure login and my ASP.NET web appplication is spanned across three web-servers (using round-robbin load balancing) what would be the best approach to maintain login-in state for the users?
• What are ASP.NET Web Forms? How is this technology different than what is available though ASP? Web Forms are the heart and soul of ASP.NET. Web Forms are the User Interface (UI) elements that give your Web applications their look and feel. Web Forms are similar to Windows Forms in that they provide properties, methods, and events for the controls that are placed onto them. However, these UI elements render themselves in the appropriate markup language required by the request, e.g. HTML. If you use Microsoft Visual Studio .NET, you will also get the familiar drag-and-drop interface used to create your UI for your Web application.
• How does VB.NET/C# achieve polymorphism? By using Abstract classes/functions.
• Can you explain what inheritance is and an example of when you might use it? Inheritance is a fundamental feature of an object oriented system and it is simply the ability to inherit data and functionality from a parent object. Rather than developing new objects from scratch, new code can be based on the work of other programmers, adding only new features that are needed.
• How would you implement inheritance using VB.NET/C#? When we set out to implement a class using inheritance, we must first start with an existing class from which we will derive our new subclass. This existing class, or base class, may be part of the .NET system class library framework, it may be part of some other application or .NET assembly, or we may create it as part of our existing application. Once we have a base class, we can then implement one or more subclasses based on that base class. Each of our subclasses will automatically have all of the methods, properties, and events of that base class ? including the implementation behind each method, property, and event. Our subclass can add new methods, properties, and events of its own - extending the original interface with new functionality. Additionally, a subclass can replace the methods and properties of the base class with its own new implementation - effectively overriding the original behavior and replacing it with new behaviors. Essentially inheritance is a way of merging functionality from an existing class into our new subclass. Inheritance also defines rules for how these methods, properties, and events can be merged.
ASP.NET questions, part 3

1. Whats an assembly? Assemblies are the building blocks of .NET Framework applications; they form the fundamental unit of deployment, version control, reuse, activation scoping, and security permissions. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. An assembly provides the common language runtime with the information it needs to be aware of type implementations. To the runtime, a type does not exist outside the context of an assembly.
2. Describe the difference between inline and code behind - which is best in a loosely coupled solution? ASP.NET supports two modes of page development: Page logic code that is written inside "script runat=server" blocks within an .aspx file and dynamically compiled the first time the page is requested on the server. Page logic code that is written within an external class that is compiled prior to deployment on a server and linked "behind" the .aspx file at run time.
3. Explain what a diffgram is, and a good use for one? A DiffGram is an XML format that is used to identify current and original versions of data elements. The DataSet uses the DiffGram format to load and persist its contents, and to serialize its contents for transport across a network connection. When a DataSet is written as a DiffGram, it populates the DiffGram with all the necessary information to accurately recreate the contents, though not the schema, of the DataSet, including column values from both the Original and Current row versions, row error information, and row order.
4. Where would you use an iHTTPModule, and what are the limitations of anyapproach you might take in implementing one? One of ASP.NET’s most useful features is the extensibility of the HTTP pipeline, the path that data takes between client and server. You can use them to extend your ASP.NET applications by adding pre- and post-processing to each HTTP request coming into your application. For example, if you wanted custom authentication facilities for your application, the best technique would be to intercept the request when it comes in and process the request in a custom HTTP module.
5. What are the disadvantages of viewstate/what are the benefits?
6. Describe session handling in a webfarm, how does it work and what are the limits?
7. How would you get ASP.NET running in Apache web servers - why would you even do this?
8. Whats MSIL, and why should my developers need an appreciation of it if at all?
9. In what order do the events of an ASPX page execute. As a developer is it important to undertsand these events? Every Page object (which your .aspx page is) has nine events, most of which you will not have to worry about in your day to day dealings with ASP.NET. The three that you will deal with the most are: Page\_Init, Page\_Load, Page\_PreRender.
10. Which method do you invoke on the DataAdapter control to load your generated dataset with data?
System.Data.Common.DataAdapter.Fill(System.Data.DataSet);
If my DataAdapter is sqlDataAdapter and my DataSet is dsUsers then it is called this way:
sqlDataAdapter.Fill(dsUsers);
11. ata in the Repeater control?
12. Which template must you provide, in order to display data in a Repeater control? ItemTemplate
13. How can you provide an alternating color scheme in a Repeater control?
AlternatingItemTemplate Like the ItemTemplate element, but rendered for every other
row (alternating items) in the Repeater control. You can specify a different appearance
for the AlternatingItemTemplate element by setting its style properties.
14. What property must you set, and what method must you call in your code, in order to bind the data from some data source to the Repeater control?
You must set the DataMember property which Gets or sets the specific table in the DataSource to bind to the control and the DataBind method to bind data from a source to a server control. This method is commonly used after retrieving a data set through a database query.
15. What base class do all Web Forms inherit from? System.Web.UI.Page
16. What method do you use to explicitly kill a user’s session?
The Abandon method destroys all the objects stored in a Session object and releases their resources.
If you do not call the Abandon method explicitly, the server destroys these objects when the session times out.

Syntax: Session.Abandon
17. How do you turn off cookies for one page in your site?
Use the Cookie.Discard Property which Gets or sets the discard flag set by the server. When true, this
property instructs the client application not to save the Cookie on the user’s hard disk when a session ends.
18. Which two properties are on every validation control? ControlToValidate & ErrorMessage properties
19. What tags do you need to add within the asp:datagrid tags to bind columns manually?
20. How do you create a permanent cookie? Setting the Expires property to MinValue means that the Cookie never expires.
21. What tag do you use to add a hyperlink column to the DataGrid?
22. What is the standard you use to wrap up a call to a Web service?
23. Which method do you use to redirect the user to another page without performing a round trip to the client? Server.transfer()
24. What is the transport protocol you use to call a Web service? SOAP. Transport Protocols: It is essential for the acceptance of Web Services that they are based on established Internet infrastructure. This in fact imposes the usage of of the HTTP, SMTP and FTP protocols based on the TCP/IP family of transports. Messaging Protocol: The format of messages exchanged between Web Services clients and Web Services should be vendor neutral and should not carry details about the technology used to implement the service. Also, the message format should allow for extensions and different bindings to specific transport protocols. SOAP and ebXML Transport are specifications which fulfill these requirements. We expect that the W3C XML Protocol Working Group defines a successor standard.
25. True or False: A Web service can only be written in .NET. False.
26. What does WSDL stand for? Web Services Description Language
27. What property do you have to set to tell the grid which page to go to when using the Pager object?
28. Where on the Internet would you look for Web services? UDDI repositaries like uddi.microsoft.com, IBM UDDI node, UDDI Registries in Google Directory, enthusiast sites like XMethods.net.
29. What tags do you need to add within the asp:datagrid tags to bind columns manually? Column tag and an ASP:databound tag.
30. Which property on a Combo Box do you set with a column name, prior to setting the DataSource, to display data in the combo box?
31. How is a property designated as read-only? In VB.NET:
Public ReadOnly Property PropertyName As ReturnType
Get ‘Your Property Implementation goes in here
End Get
End Property
in C#
public returntype PropertyName
{
get{
//property implementation goes here
}
// Do not write the set implementation
}

at [10:02 AM](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_2987.html)

Labels: [ASP.Net](http://silverlightvsaspnet.blogspot.com/search/label/ASP.Net)

**[Framework Interview Questions](http://silverlightvsaspnet.blogspot.com/2010/02/framework-interview-questions_04.html)**

1. What do you know about .NET assemblies? Assemblies are the smallest units of versioning and deployment in the .NET application. Assemblies are also the building blocks for programs such as Web services, Windows services, serviced components, and .NET remoting applications.
2. What’s the difference between private and shared assembly? Private assembly is used inside an application only and does not have to be identified by a strong name. Shared assembly can be used by multiple applications and has to have a strong name.
3. What’s a strong name? A strong name includes the name of the assembly, version number, culture identity, and a public key token.
4. How can you tell the application to look for assemblies at the locations other than its own install? Use the
directive in the XML .config file for a given application.

should do the trick. Or you can add additional search paths in the Properties box of the deployed application.
5. How can you debug failed assembly binds? Use the Assembly Binding Log Viewer (fuslogvw.exe) to find out the paths searched.
6. Where are shared assemblies stored? Global assembly cache.
7. How can you create a strong name for a .NET assembly? With the help of Strong Name tool (sn.exe).
8. Where’s global assembly cache located on the system? Usually C:\winnt\assembly or C:\windows\assembly.
9. Can you have two files with the same file name in GAC? Yes, remember that GAC is a very special folder, and while normally you would not be able to place two files with the same name into a Windows folder, GAC differentiates by version number as well, so it’s possible for MyApp.dll and MyApp.dll to co-exist in GAC if the first one is version 1.0.0.0 and the second one is 1.1.0.0.
10. So let’s say I have an application that uses MyApp.dll assembly, version 1.0.0.0. There is a security bug in that assembly, and I publish the patch, issuing it under name MyApp.dll 1.1.0.0. How do I tell the client applications that are already installed to start using this new MyApp.dll? Use publisher policy. To configure a publisher policy, use the publisher policy configuration file, which uses a format similar app .config file. But unlike the app .config file, a publisher policy file needs to be compiled into an assembly and placed in the GAC.
11. What is delay signing? Delay signing allows you to place a shared assembly in the GAC by signing the assembly with just the public key. This allows the assembly to be signed with the private key at a later stage, when the development process is complete and the component or assembly is ready to be deployed. This process enables developers to work with shared assemblies as if they were strongly named, and it secures the private key of the signature from being accessed at different stages of development.
• Windows code security questions
1. What’s the difference between code-based security and role-based security? Which one is better? Code security is the approach of using permissions and permission sets for a given code to run. The admin, for example, can disable running executables off the Internet or restrict access to corporate database to only few applications. Role-based security most of the time involves the code running with the privileges of the current user. This way the code cannot supposedly do more harm than mess up a single user account. There’s no better, or 100% thumbs-up approach, depending on the nature of deployment, both code-based and role-based security could be implemented to an extent.
2. How can you work with permissions from your .NET application? You can request permission to do something and you can demand certain permissions from other apps. You can also refuse permissions so that your app is not inadvertently used to destroy some data.
3. How can C# app request minimum permissions?
using System.Security.Permissions;
[assembly:FileDialogPermissionAttribute(SecurityAction.RequestMinimum, Unrestricted=true)]
4. What’s a code group? A code group is a set of assemblies that share a security context.
5. What’s the difference between authentication and authorization? Authentication happens first. You verify user’s identity based on credentials. Authorization is making sure the user only gets access to the resources he has credentials for.
6. What are the authentication modes in ASP.NET? None, Windows, Forms and Passport.
7. Are the actual permissions for the application defined at run-time or compile-time? The CLR computes actual permissions at runtime based on code group membership and the calling chain of the code.

at [10:01 AM](http://silverlightvsaspnet.blogspot.com/2010/02/framework-interview-questions_04.html)

Labels: [Framework](http://silverlightvsaspnet.blogspot.com/search/label/Framework)

**[ASP.Net Interview Questions](http://silverlightvsaspnet.blogspot.com/2010/02/aspnet-interview-questions_04.html)**

• What is datagrid? The DataGrid Web server control is a powerful tool for displaying information from a data source. It is easy to use; you can display editable data in a professional-looking grid by setting only a few properties. At the same time, the grid has a sophisticated object model that provides you with great flexibility in how you display the data.
• What’s the difference between the System.Web.UI.WebControls.DataGrid and and System.Windows.Forms.DataGrid? The Web UI control does not inherently support master-detail data structures. As with other Web server controls, it does not support two-way data binding. If you want to update data, you must write code to do this yourself. You can only edit one row at a time. It does not inherently support sorting, although it raises events you can handle in order to sort the grid contents. You can bind the Web Forms DataGrid to any object that supports the IEnumerable interface. The Web Forms DataGrid control supports paging. It is easy to customize the appearance and layout of the Web Forms DataGrid control as compared to the Windows Forms one.
• How do you customize the column content inside the datagrid? If you want to customize the content of a column, make the column a template column. Template columns work like item templates in the DataList or Repeater control, except that you are defining the layout of a column rather than a row.
• How do you apply specific formatting to the data inside the cells? You cannot specify formatting for columns generated when the grid’s AutoGenerateColumns property is set to true, only for bound or template columns. To format, set the column’s DataFormatString property to a string-formatting expression suitable for the data type of the data you are formatting.
• How do you hide the columns? One way to have columns appear dynamically is to create them at design time, and then to hide or show them as needed. You can do this by setting a column’s Visible property.
• How do you display an editable drop-down list? Displaying a drop-down list requires a template column in the grid. Typically, the ItemTemplate contains a control such as a data-bound Label control to show the current value of a field in the record. You then add a drop-down list to the EditItemTemplate. In Visual Studio, you can add a template column in the Property builder for the grid, and then use standard template editing to remove the default TextBox control from the EditItemTemplate and drag a DropDownList control into it instead. Alternatively, you can add the template column in HTML view. After you have created the template column with the drop-down list in it, there are two tasks. The first is to populate the list. The second is to preselect the appropriate item in the list — for example, if a book’s genre is set to “fiction,” when the drop-down list displays, you often want “fiction” to be preselected.
• How do you check whether the row data has been changed? The definitive way to determine whether a row has been dirtied is to handle the changed event for the controls in a row. For example, if your grid row contains a TextBox control, you can respond to the control’s TextChanged event. Similarly, for check boxes, you can respond to a CheckedChanged event. In the handler for these events, you maintain a list of the rows to be updated. Generally, the best strategy is to track the primary keys of the affected rows. For example, you can maintain an ArrayList object that contains the primary keys of the rows to update.