ASP.NET Interview Questions

This is a list of questions I have gathered and created over a period of time from my experience, many of which I felt where incomplete or simply wrong. I have finally taken the time to go through each question and correct them to the best of my ability. However, please feel free to post feedback to challenge, improve, or suggest new questions. I want to thank those of you that have contributed quality questions and corrections thus far.

There are some questions in this list that I do not consider to be good questions for an interview. However, they do exist on other lists available on the Internet so I felt compelled to keep them here for easy access.

1. **Describe the role of *inetinfo.exe, aspnet_isapi.dll* and *aspnet_wp.exe* in the page loading process**.
   inetinfo.exe is theMicrosoft IIS server running, handling ASP.NET requests among other things.When an ASP.NET request is received (usually a file with .aspx extension), the ISAPI filter aspnet_isapi.dll takes care of it by passing the request tothe actual worker process aspnet_wp.exe.

2. **What's the difference between Response.Write() andResponse.Output.Write()?**
   Response.Output.Write() allows you to write formatted output.

3. **What methods are fired during the page load?**
   Init() - when the page is instantiated
   Load() - when the page is loaded into server memory
   PreRender() - the brief moment before the page is displayed to the user as HTML
   Unload() - when page finishes loading.

4. **When during the page processing cycle is ViewState available?**
   After the Init() and before the Page_Load(), or OnLoad() for a control.

5. **What namespace does the Web page belong in the .NET Framework class hierarchy?**
   System.Web.UI.Page

6. **Where do you store the information about the user's locale?**
   System.Web.UI.Page.Culture

7. **What's the difference between Codebehind="MyCode.aspx.cs" andSrc="MyCode.aspx.cs"?**
   CodeBehind is relevant to Visual Studio.NET only.

8. **What's a bubbled event?**
   When you have a complex control, like DataGrid, writing an event processing routine for each object (cell, button, row, etc.) is quite tedious. The controls can bubble up their eventhandlers, allowing the main DataGrid event handler to take care of its constituents.

9. **Suppose you want a certain ASP.NET function executed on MouseOver for a certain button. Where do you add an event handler?**
   Add an OnMouseOver attribute to the button. Example:
   btnSubmit.Attributes.Add("onmouseover","someClientCodeHere();");

10. **What data types do the RangeValidator control support?**
    Integer, String, and Date.

11. **Explain the differences between Server-side and Client-side code?**
    Server-side code executes on the server. Client-side code executes in the client's browser.

12. **What type of code (server or client) is found in a Code-Behind class?**
    The answer is server-side code since code-behind is executed on the server. However, during the code-behind's execution on the server, it can render client-side code such as JavaScript to be processed in the clients browser. But just to be clear, code-behind executes on the server, thus making it server-side code.

13. **Should user input data validation occur server-side or client-side? Why?**
    All user input data validation should occur on the server at a minimum. Additionally, client-side validation can be performed where deemed appropriate and feasable to provide a richer, more responsive experience for the user.

14. **What is the difference between Server.Transfer and Response.Redirect? Why would I choose one over the other?**
    Server.Transfer transfers page processing from one page directly to the next page without making a round-trip back to the client's browser. This provides a faster response with a little less overhead on the server. Server.Transfer does not update the clients url history list or current url. Response.Redirect is used to redirect the user's browser to another page or site. This performas a trip back to the client where the client's browser is redirected to the new page. The user's browser history list is updated to reflect the new address.

15. **Can you explain the difference between an ADO.NET Dataset and an ADO Recordset?**
    Valid answers are:
    · A DataSet can represent an entire relational database in memory, complete with tables,

relations, and views.

- A DataSet is designed to work without any continuing connection to the original data source.
- Data in a DataSet is bulk-loaded, rather than being loaded on demand.
- There's no concept of cursor types in a DataSet.
- DataSets have no current record pointer You can use For Each loops to move through the data.
- You can store many edits in a DataSet, and write them to the original data source in a single operation.
- Though the DataSet is universal, other objects in ADO.NET come in different versions for different data sources.

16. **What is the Global.asax used for?**
The Global.asax (including the Global.asax.cs file) is used to implement application and session level events.

17. **What are the Application_Start and Session_Start subroutines used for?**
This is where you can set the specific variables for the Application and Session objects.

18. **Can you explain what inheritance is and an example of when you might use it?**
When you want to inherit (use the functionality of) another class. Example: With a base class named Employee, a Manager class could be derived from the Employee base class.

19. **Whats an assembly?**
Assemblies are the building blocks of the .NET framework. Overview of assemblies from MSDN

20. **Describe the difference between inline and code behind.**
Inline code written along side the html in a page. Code-behind is code written in a separate file and referenced by the .aspx page.

21. **Explain what a diffgram is, and a good use for one?**
The DiffGram is one of the two XML formats that you can use to render DataSet object contents to XML. A good use is reading database data to an XML file to be sent to a Web Service.

22. **Whats MSIL, and why should my developers need an appreciation of it if at all?**
MSIL is the Microsoft Intermediate Language. All .NET compatible languages will get converted to MSIL. MSIL also allows the .NET Framework to JIT compile the assembly on the installed computer.

23. **Which method do you invoke on the DataAdapter control to load your generated dataset with data?**
The Fill() method.

24. **Can you edit data in the Repeater control?**
No, it just reads the information from its data source**.**

25. **Which template must you provide, in order to display data in a Repeater control?**
ItemTemplate.

26. **How can you provide an alternating color scheme in a Repeater control?**
Use the AlternatingItemTemplate**.**

27. **What property must you set, and what method must you call in your code, in order to bind the data from a data source to the Repeater control?**
You must set the DataSource property and call the DataBind method.

28. **What base class do all Web Forms inherit from?**
The Page class.

29. **Name two properties common in every validation control?**
ControlToValidate property and Text property.

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?**
DataTextField property.

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

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

Web Service Questions

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

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

2. **True or False: A Web service can only be written in .NET?**
   False

3. **What does WSDL stand for?**
   Web Services Description Language.

4. **Where on the Internet would you look for Web services?**
   http://www.uddi.org

5. **True or False: To test a Web service you must create a Windows application or Web application to consume this service?**
   False, the web service comes with a test page and it provides HTTP-GET method to test.

State Management Questions

1. **What is ViewState?**
   ViewState allows the state of objects (serializable) to be stored in a hidden field on the page. ViewState is transported to the client and back to the server, and is not stored on the server or any other external source. ViewState is used the retain the state of server-side objects between postabacks.

2. **What is the lifespan for items stored in ViewState?**
   Item stored in ViewState exist for the life of the current page. This includes postbacks (to the same page).

3. **What does the "EnableViewState" property do? Why would I want it on or off?**
   It allows the page to save the users input on a form across postbacks. It saves the server-side values for a given control into ViewState, which is stored as a hidden value on the page before sending the page to the clients browser. When the page is posted back to the server the server control is recreated with the state stored in viewstate.

4. **What are the different types of Session state management options available with ASP.NET?**
   ASP.NET provides In-Process and Out-of-Process state management. In-Process stores the session in memory on the web server. This requires the a "sticky-server" (or no load-balancing) so that the user is always reconnected to the same web server. Out-of-Process Session state management stores data in an external data source. The external data source may be either a SQL Server or a State Server service. Out-of-Process state management requires that all objects stored in session are serializable.

Interview Questions on DotNet Framework

1. **What is CLR?**
   The .NET Framework provides a runtime environment called the Common Language Runtime or CLR (similar to the Java Virtual Machine or JVM in Java), which handles the execution of code and provides useful services for the implementation of the program. CLR takes care of code management at program execution and provides various beneficial services such as memory management, thread management, security management, code verification, compilation, and other system services. The managed code that targets CLR benefits from useful features such as cross-language integration, cross-language exception handling, versioning, enhanced security, deployment support, and debugging.

2. What is CTS?
Common Type System (CTS) describes how types are declared, used and managed in the runtime and facilitates cross-language integration, type safety, and high performance code execution.

3. What is CLS?
The CLS is simply a specification that defines the rules to support language integration in such a way that programs written in any language, yet can interoperate with one another, taking full advantage of inheritance, polymorphism, exceptions, and other features. These rules and the specification are documented in the ECMA proposed standard document, "Partition I Architecture"

4. What is CLI?
The CLI is a set of specifications for a runtime environment, including a common type system, base class library, and a machine-independent intermediate code known as the Common Intermediate Language (CIL). (Source: Wikipedia.)

5. Explain Namespace.
Namespaces are logical groupings of names used within a program. There may be multiple namespaces in a single application code, grouped based on the identifiers□ use. The name of any given identifier must appear only once in its namespace.

6. Explain Assembly and Manifest.
An assembly is a collection of one or more files and one of them (DLL or EXE) contains a special metadata called Assembly Manifest. The manifest is stored as binary data and contains details like versioning requirements for the assembly, the author, security permissions, and list of files forming the assembly. An assembly is created whenever a DLL is built. The manifest can be viewed programmatically by making use of classes from the System.Reflection namespace. The tool Intermediate Language Disassembler (ILDASM) can be used for this purpose. It can be launched from the command prompt or via Start> Run.

7. What is Shadow Copy?
In order to replace a COM component on a live web server, it was necessary to stop the entire website, copy the new files and then restart the website. This is not feasible for the web servers that need to be always running. .NET components are different. They can be overwritten at any time using a mechanism called Shadow Copy. It prevents the Portable Executable (PE) files like

DLLs and EXEs from being locked. Whenever new versions of the PEs are released, they are automatically detected by the CLR and the changed components will be automatically loaded. They will be used to process all new requests not currently executing, while the older version still runs the currently executing requests. By bleeding out the older version, the update is completed.

8. What is DLL Hell?
DLL hell is the problem that occurs when an installation of a newer application might break or hinder other applications as newer DLLs are copied into the system and the older applications do not support or are not compatible with them. .NET overcomes this problem by supporting multiple versions of an assembly at any given time. This is also called side-by-side component versioning.

9. Explain Web Services.
Web services are programmable business logic components that provide access to functionality through the Internet. Standard protocols like HTTP can be used to access them. Web services are based on the Simple Object Access Protocol (SOAP), which is an application of XML. Web services are given the .asmx extension.

10. Explain Windows Forms.
Windows Forms is employed for developing Windows GUI applications. It is a class library that gives developers access to Windows Common Controls with rich functionality. It is a common GUI library for all the languages supported by the .NET Framework.

11. Define Boxing and UnBoxing
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 unboxing.

12. Define Value Types and Reference Types
Value Types Value types are primitive types that are mapped directly to the FCL. Like Int32 maps to System.Int32, double maps to System.double. All value types are stored on stack and all the value types are derived from System.ValueType. All structures and enumerated types that are derived from System.ValueType are created on stack, hence known as ValueType. Reference TypesReference Types are different from value types in such a way that memory is allocated to them from the heap. All the classes are of reference type. C# new operator returns the memory address of the object.

13. What are Service Oriented Architectures (SOA)?
SOA describes an information technology architecture that enables distributed computing environments with many different types of computing platforms and applications. Web services are one of the technologies that help make SOAs possible. As a concept, SOA has been around since the 1980s, but many early IT technologies failed to achieve the goal of linking different types of applications and systems. By making early investments with .NET, Microsoft has helped provide the building blocks that today are putting many enterprise customers on the path to successfully implementing SOAs. With SOAs, companies can benefit from the unimpeded flow of information that is the hallmark of connected systems.

14. What are Web Services Enhancements for Microsoft .NET (WSE)?

WSE is an add-on to Microsoft Visual Studio .NET and the Microsoft .NET Framework that helps developers build greater security features into Web services using the latest Web services protocol specifications and standards. With WSE 2.0 developers can create security-enhanced connected systems that help improve business processes within□and beyond□corporate trust boundaries and create new revenue-generating opportunities.

15. What is a Smart Client?
Smart clients are client applications that consume Web services and reside on user hardware such as desktop PCs, laptops, Pocket PCs, and Smartphones. They are easily deployed and managed and provide an adaptive, responsive, and rich interactive experience by taking advantage of the computing resources on the device and intelligently connecting to distributed data sources.

16. What is .NET Passport?
.NET Passport is a Web-based service that is designed to make signing in to Web sites fast and easy. Passport enables participating sites to authenticate a user with a single set of sign-in credentials, alleviating the need for users to remember numerous passwords and user names.

17. Does C# support multiple inheritance?
No, use interfaces instead

18. What□s the implicit name of the parameter that gets passed into the class□ set method?
Value, and its datatype depends on whatever variable we□re changing

19. What□s the top .NET class that everything is derived from?
System.Object.

20. How□s method overriding different from overloading?
When overriding, you change the method behavior for a derived class. Overloading simply involves having a method with the same name within the class.

21. What is strong name?
A name that consists of an assembly's identity□its simple text name, version number, and culture information (if provided)□strengthened by a public key and a digital signature generated over the assembly.

22. What is Application Domain?
The primary purpose of the AppDomain is to isolate an application from other applications. Win32 processes provide isolation by having distinct memory address spaces. This is effective, but it is expensive and doesn't scale well. The .NET runtime enforces AppDomain isolation by keeping control over the use of memory - all memory in the AppDomain is managed by the .NET runtime, so the runtime can ensure that AppDomains do not access each other's memory. Objects in different application domains communicate either by transporting copies of objects across application domain boundaries, or by using a proxy to exchange messages.

23. What is serialization in .NET? What are the ways to control serialization?
Serialization is the process of converting an object into a stream of bytes. Deserialization is the opposite process of creating an object from a stream of bytes. Serialization/Deserialization is

mostly used to transport objects (e.g. during remoting), or to persist objects (e.g. to a file or database).Serialization can be defined as the process of storing the state of an object to a storage medium. During this process, the public and private fields of the object and the name of the class, including the assembly containing the class, are converted to a stream of bytes, which is then written to a data stream. When the object is subsequently deserialized, an exact clone of the original object is created. Binary serialization preserves type fidelity, which is useful for preserving the state of an object between different invocations of an application. For example, you can share an object between different applications by serializing it to the clipboard. You can serialize an object to a stream, disk, memory, over the network, and so forth. Remoting uses serialization to pass objects "by value" from one computer or application domain to another. XML serialization serializes only public properties and fields and does not preserve type fidelity. This is useful when you want to provide or consume data without restricting the application that uses the data. Because XML is an open standard, it is an attractive choice for sharing data across the Web. SOAP is an open standard, which makes it an attractive choice. There are two separate mechanisms provided by the .NET class library - XmlSerializer and SoapFormatter/BinaryFormatter. Microsoft uses XmlSerializer for Web Services, and uses SoapFormatter/BinaryFormatter for remoting. Both are available for use in your own code.

24. What□s an interface class?
It□s an abstract class with public abstract methods all of which must be implemented in the inherited classes

25. What is the transport protocol you use to call a Web service
SOAP is the preferred protocol

26. What are Satellite Assemblies?
Satellite assemblies are often used to deploy language-specific resources for an application. These language-specific assemblies work in side-by-side execution because the application has a separate product ID for each language and installs satellite assemblies in a language-specific subdirectory for each language. When uninstalling, the application removes only the satellite assemblies associated with a given language and .NET Framework version. No core .NET Framework files are removed unless the last language for that .NET Framework version is being removed.

27. What is Global Assembly Cache (GAC) and what is the purpose of it?
Each computer where the common language runtime is installed has a machine-wide code cache called the global assembly cache. The global assembly cache stores assemblies specifically designated to be shared by several applications on the computer. You should share assemblies by installing them into the global assembly cache only when you need to.

28. What is Reflection in .NET?
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.

29. What is the managed and unmanaged code in .net?
The .NET Framework provides a run-time environment called the Common Language Runtime, which manages the execution of code and provides services that make the development process

easier. Compilers and tools expose the runtime's functionality and enable you to write code that benefits from this managed execution environment. Code that you develop with a language compiler that targets the runtime is called managed code; it benefits from features such as cross-language integration, cross-language exception handling, enhanced security, versioning and deployment support, a simplified model for component interaction, and debugging and profiling services

30. What are the access-specifiers available in c#?
Private, Protected, Public, Internal, Protected Internal.

31. Difference between OLEDB Provider and SqlClient ?
SQLClient .NET classes are highly optimized for the .net / sqlserver combination and achieve optimal results. The SqlClient data provider is fast. It's faster than the Oracle provider, and faster than accessing database via the OleDb layer. It's faster because it accesses the native library (which automatically gives you better performance), and it was written with lots of help from the SQL Server team.

32. Differences between dataset.clone and dataset.copy?
Clone - Copies the structure of the DataSet, including all DataTable schemas, relations, and constraints.Does not copy any data . Copy - Copies both the structure and data for this DataSet.

33. In a Webservice, need to display 10 rows from a table. So DataReader or DataSet is best choice?
WebService will support only DataSet.

34. What is Remoting?
The process of communication between different operating system processes, regardless of whether they are on the same computer. The .NET remoting system is an architecture designed to simplify communication between objects living in different application domains, whether on the same computer or not, and between different contexts, whether in the same application domain or not.

35. What□s the difference between System.String and System.StringBuilder classes?
System.String is immutable; System.StringBuilder was designed with the purpose of having a mutable string where a variety of operations can be performed.

36. What□s a delegate?
A delegate object encapsulates a reference to a method. In C++ they were referred to as function pointers.

37. What□s the implicit name of the parameter that gets passed into the class□ set method?
Value, and its datatype depends on whatever variable we□re changing.

38. How do you inherit from a class in C#?
Place a colon and then the name of the base class. Notice that it□s double colon in C++.

39. Does C# support multiple inheritance?
No, use interfaces instead.

40. When you inherit a protected class-level variable, who is it available to?
Classes in the same namespace.

41. Are private class-level variables inherited?
Yes, but they are not accessible, so looking at it you can honestly say that they are not inherited. But they are.

42. Describe the accessibility modifier protected internal.
It□s available to derived classes and classes within the same Assembly (and naturally from the base class it□s declared in).

43. C# provides a default constructor for me. I write a constructor that takes a string as a parameter, but want to keep the no parameter one. How many constructors should I write?
Two. Once you write at least one constructor, C# cancels the freebie constructor, and now you have to write one yourself, even if there□s no implementation in it.

44. How□s method overriding different from overloading?
When overriding, you change the method behavior for a derived class. Overloading simply involves having a method with the same name within the class.

45. What does the keyword virtual mean in the method definition?
The method can be over-ridden.

46. Can you declare the override method static while the original method is non-static?
No, you can□t, the signature of the virtual method must remain the same, only the keyword virtual is changed to keyword override.

47. Can you override private virtual methods?
No, moreover, you cannot access private methods in inherited classes, have to be protected in the base class to allow any sort of access.

48. Can you prevent your class from being inherited and becoming a base class for some other classes?
Yes, that□s what keyword sealed in the class definition is for. The developer trying to derive from your class will get a message: cannot inherit from Sealed class WhateverBaseClassName. It□s the same concept as final class in Java.

49. Can you allow class to be inherited, but prevent the method from being over-ridden?
Yes, just leave the class public and make the method sealed.

50. What□s an abstract class?
A class that cannot be instantiated. A concept in C++ known as pure virtual method. A class that must be inherited and have the methods over-ridden. Essentially, it□s a blueprint for a class without any implementation.

ASP .Net Page Life Cycle
Explain the life cycle of an ASP .NET page.
Latest Answer: 1)Page_PreInit2)Page_Init3)Page_InitComplete4)Page_PreLoad5)Page_Load6)Control Events7)Page_LoadComplete8)Page_PreRender9)SaveViewState10)Page_Render11)Page_Unload ...

Virtual function in c#
explain virtual function in c# with an example
Latest Answer: Some Facts about Virtual Keyword1)It is not compulsury to mark the derived/child class function with Override KeyWord while base/parent class contains a virtual method2)Instead of Virtual we can use New Keyword3)We will get a warning if we won't use ...

What is an application server?

As defined in Wikipedia, an application server is a software engine that delivers applications to client computers or devices. The application server runs your server code. Some well known application servers are IIS (Microsoft), WebLogic Server (BEA), JBoss (Red Hat), WebSphere (IBM).

Compare C# and VB.NET

A detailed comparison can be found over here.

What is a base class and derived class?

A class is a template for creating an object. The class from which other classes derive fundamental functionality is called a base class. For e.g. If Class Y derives from Class X, then Class X is a base class.

The class which derives functionality from a base class is called a derived class. If Class Y derives from Class X, then Class Y is a derived class.

What is an extender class?

An extender class allows you to extend the functionality of an existing control. It is used in Windows forms applications to add properties to controls.

A demonstration of extender classes can be found over here.

What is inheritance?

Inheritance represents the relationship between two classes where one type derives functionality from a second type and then extends it by adding new methods, properties, events, fields and constants.

C# support two types of inheritance:

- Implementation inheritance
- Interface inheritance

What is implementation and interface inheritance?

When a class (type) is derived from another class(type) such that it inherits all the members of

the base type it is Implementation Inheritance.

When a type (class or a struct) inherits only the signatures of the functions from another type it is Interface Inheritance.
In general Classes can be derived from another class, hence support Implementation inheritance. At the same time Classes can also be derived from one or more interfaces. Hence they support Interface inheritance.
Source: Exforsys.

What is inheritance hierarchy?

The class which derives functionality from a base class is called a derived class. A derived class can also act as a base class for another class. Thus it is possible to create a tree-like structure that illustrates the relationship between all related classes. This structure is known as the inheritance hierarchy.

How do you prevent a class from being inherited?

In VB.NET you use the **NotInheritable** modifier to prevent programmers from using the class as a base class. In C#, use the **sealed** keyword.

When should you use inheritance?

Read this.

Define Overriding?

Overriding is a concept where a method in a derived class uses the same name, return type, and arguments as a method in its base class. In other words, if the derived class contains its own implementation of the method rather than using the method in the base class, the process is called overriding.

Can you use multiple inheritance in .NET?

.NET supports only single inheritance. However the purpose is accomplished using multiple interfaces.

Why don't we have multiple inheritance in .NET?

There are several reasons for this. In simple words, the efforts are more, benefits are less. Different languages have different implementation requirements of multiple inheritance. So in order to implement multiple inheritance, we need to study the implementation aspects of all the languages that are CLR compliant and then implement a common methodology of implementing it. This is too much of efforts. Moreover multiple interface inheritance very much covers the benefits that multiple inheritance has.

What is an Interface?

An interface is a standard or contract that contains only the signatures of methods or events. The implementation is done in the class that inherits from this interface. Interfaces are primarily used to set a common standard or contract.

When should you use abstract class vs interface or What is the difference between an abstract class and interface?

I would suggest you to read this. There is a good comparison given over here.

What are events and delegates?
An event is a message sent by a control to notify the occurrence of an action. However it is not known which object receives the event. For this reason, .NET provides a special type called Delegate which acts as an intermediary between the sender object and receiver object.

What is business logic?
It is the functionality which handles the exchange of information between database and a user interface.

What is a component?
Component is a group of logically related classes and methods. A component is a class that implements the IComponent interface or uses a class that implements IComponent interface.

What is a control?
A control is a component that provides user-interface (UI) capabilities.

What are the differences between a control and a component?
The differences can be studied over here.

What are design patterns?
Design patterns are common solutions to common design problems.

What is a connection pool?
A connection pool is a 'collection of connections' which are shared between the clients requesting one. Once the connection is closed, it returns back to the pool. This allows the connections to be reused.

What is a flat file?
A flat file is the name given to text, which can be read or written only sequentially.

What are functional and non-functional requirements?
Functional requirements defines the behavior of a system whereas non-functional requirements specify how the system should behave; in other words they specify the quality requirements and judge the behavior of a system.

E.g.

Functional - Display a chart which shows the maximum number of products sold in a region.

Non-functional – The data presented in the chart must be updated every 5 minutes.

What is the global assembly cache (GAC)?
GAC is a machine-wide cache of assemblies that allows .NET applications to share libraries. GAC solves some of the problems associated with dll's (DLL Hell).

What is a stack? What is a heap? Give the differences between the two?
Stack is a place in the memory where value types are stored. Heap is a place in the memory where the reference types are stored.

Check this link for the differences.

What is instrumentation?
It is the ability to monitor an application so that information about the application's progress, performance and status can be captured and reported.

What is code review?
The process of examining the source code generally through a peer, to verify it against best practices.

What is logging?
Logging is the process of persisting information about the status of an application.

What are mock-ups?
Mock-ups are a set of designs in the form of screens, diagrams, snapshots etc., that helps verify the design and acquire feedback about the application's requirements and use cases, at an early stage of the design process.

What is a Form?
A form is a representation of any window displayed in your application. Form can be used to create standard, borderless, floating, modal windows.

What is a multiple-document interface(MDI)?
A user interface container that enables a user to work with more than one document at a time. E.g. Microsoft Excel.

What is a single-document interface (SDI) ?
A user interface that is created to manage graphical user interfaces and controls into single windows. E.g. Microsoft Word

What is BLOB ?
A BLOB (binary large object) is a large item such as an image or an exe represented in binary form.

What is ClickOnce?
ClickOnce is a new deployment technology that allows you to create and publish self-updating applications that can be installed and run with minimal user interaction.

What is object role modeling (ORM) ?
It is a logical model for designing and querying database models. There are various ORM tools in the market like CaseTalk, Microsoft Visio for Enterprise Architects, Infagon etc.

What is a private assembly?
A private assembly is local to the installation directory of an application and is used only by that application.

What is a shared assembly?
A shared assembly is kept in the global assembly cache (GAC) and can be used by one or more applications on a machine.

What is the difference between user and custom controls?
User controls are easier to create whereas custom controls require extra effort.

User controls are used when the layout is static whereas custom controls are used in dynamic layouts.

A user control cannot be added to the toolbox whereas a custom control can be.

A separate copy of a user control is required in every application that uses it whereas since custom controls are stored in the GAC, only a single copy can be used by all applications.

Where do custom controls reside?

In the global assembly cache (GAC).

What is a third-party control ?

A third-party control is one that is not created by the owners of a project. They are usually used to save time and resources and reuse the functionality developed by others (third-party).

What is a binary formatter?

Binary formatter is used to serialize and deserialize an object in binary format.

What is Boxing/Unboxing?

Boxing is used to convert value types to object.

E.g. int x = 1;

object obj = x ;

Unboxing is used to convert the object back to the value type.

E.g. int y = (int)obj;

Boxing/unboxing is quiet an expensive operation.

What is a COM Callable Wrapper (CCW)?

CCW is a wrapper created by the common language runtime(CLR) that enables COM components to access .NET objects.

What is a Runtime Callable Wrapper (RCW)?

RCW is a wrapper created by the common language runtime(CLR) to enable .NET components to call COM components.

What is a digital signature?

A digital signature is an electronic signature used to verify/gurantee the identity of the individual who is sending the message.

What is garbage collection?

Garbage collection is the process of managing the allocation and release of memory in your applications. Read this article for more information.

What is globalization?

Globalization is the process of customizing applications that support multiple cultures and regions.

What is localization?

Localization is the process of customizing applications that support a given culture and regions.

What is MIME?

The definition of MIME or Multipurpose Internet Mail Extensions as stated in MSDN is "MIME is a standard that can be used to include content of various types in a single message. MIME extends the Simple Mail Transfer Protocol (SMTP) format of mail messages to include multiple content, both textual and non-textual. Parts of the message may be images, audio, or text in different character sets. The MIME standard derives from RFCs such as 2821 and 2822". Quoted from here.

.NET Interview Questions

Basic .NET and ASP.NET interview questions

Submitter said questions were asked in a US company hiring a Web developer.

1. **Explain the .NET architecture.**
2. **How many languages .NET is supporting now?** - When .NET was introduced it came with several languages. VB.NET, C#, COBOL and Perl, etc. The site DotNetLanguages.Net says 44 languages are supported.
3. **How is .NET able to support multiple languages?** - a language should comply with the Common Language Runtime standard to become a .NET language. In .NET, code is compiled to Microsoft Intermediate Language (MSIL for short). This is called as Managed Code. This Managed code is run in .NET environment. So after compilation to this IL the language is not a barrier. A code can call or use a function written in another language.
4. **How ASP .NET different from ASP?** - Scripting is separated from the HTML, Code is compiled as a DLL, these DLLs can be executed on the server.
5. **Resource Files: How to use the resource files, how to know which language to use?**
6. **What is smart navigation?** - The cursor position is maintained when the page gets refreshed due to the server side validation and the page gets refreshed.
7. **What is view state?** - The web is stateless. But in ASP.NET, the state of a page is maintained in the in the page itself automatically. How? The values are encrypted and saved in hidden controls. this is done automatically by the ASP.NET. This can be switched off / on for a single control
8. **Explain the life cycle of an ASP .NET page.**
9. **How do you validate the controls in an ASP .NET page?** - Using special validation controls that are meant for this. We have Range Validator, Email Validator.
10. **Can the validation be done in the server side? Or this can be done only in the Client side?** - Client side is done by default. Server side validation is also possible. We can switch off the client side and server side can be done.
11. **How to manage pagination in a page?** - Using pagination option in DataGrid control. We have to set the number of records for a page, then it takes care of pagination by itself.
12. **What is ADO .NET and what is difference between ADO and ADO.NET?** - ADO.NET is stateless mechanism. I can treat the ADO.Net as a separate in-memory database where in I can use relationships between the tables and select insert and updates to the database. I can update the actual database as a batch.

Tough ASP.NET interview questions

1. Describe the difference between a Thread and a Process?
2. What is a Windows Service and how does its lifecycle differ from a .standard. EXE?
3. What is the maximum amount of memory any single process on Windows can address? Is this different than the maximum virtual memory for the system? How would this affect a system design?
4. What is the difference between an EXE and a DLL?
5. What is strong-typing versus weak-typing? Which is preferred? Why?
6. What.s wrong with a line like this? DateTime.Parse(myString
7. What are PDBs? Where must they be located for debugging to work?
8. What is cyclomatic complexity and why is it important?
9. Write a standard lock() plus double check to create a critical section around a variable access.
10. What is FullTrust? Do GAC'ed assemblies have FullTrust?
11. What benefit does your code receive if you decorate it with attributes demanding specific Security permissions?
12. What does this do? gacutil /l | find /i "about"
13. What does this do? sn -t foo.dll
14. What ports must be open for DCOM over a firewall? What is the purpose of Port 135?
15. Contrast OOP and SOA. What are tenets of each
16. How does the XmlSerializer work? What ACL permissions does a process using it require?
17. Why is catch(Exception) almost always a bad idea?
18. What is the difference between Debug.Write and Trace.Write? When should each be used?
19. What is the difference between a Debug and Release build? Is there a significant speed difference? Why or why not?
20. Does JITting occur per-assembly or per-method? How does this affect the working set?
21. Contrast the use of an abstract base class against an interface?
22. What is the difference between a.Equals(b) and a == b?
23. In the context of a comparison, what is object identity versus object equivalence?
24. How would one do a deep copy in .NET?
25. Explain current thinking around IClonable.
26. What is boxing?
27. Is string a value type or a reference type?

Interview questions for Web application developers

The following set was set in by a reader of the site:

> Following are the questions from an interview I attended for in C#, ASP.NET, XML and Sql Server. I will try to add some more as soon as I recollect. Hope these questions will be useful for people attending interviews in this area.

1. What is the maximum length of a varchar field in SQL Server?
2. How do you define an integer in SQL Server?
3. How do you separate business logic while creating an ASP.NET application?
4. If there is a calendar control to be included in each page of your application, and we do not intend to use the Microsoft-provided calendar control, how do you develop it? Do you copy and paste the code into each and very page of your application?
5. How do you debug an ASP.NET application?
6. How do you deploy an ASP.NET application?
7. Name a few differences between .NET application and a Java application?
8. Specify the best ways to store variables so that we can access them in various pages of ASP.NET application?
9. What are the XML files that are important in developing an ASP.NET application?
10. What is XSLT and what is its use?

Interview questions for C# developers

Useful for preparation, but too specific to be used in the interview.

1. **Is it possible to inline assembly or IL in C# code?** - No.
2. **Is it possible to have a static indexer in C#?** - No. Static indexers are not allowed in C#.
3. **If I return out of a try/finally in C#, does the code in the finally-clause run?** - Yes. The code in the finally always runs. If you return out of the try block, or even if you do a "goto" out of the try, the finally block always runs:

```
using System;

class main
{
        public static void Main()
        {
                try
                {
                        Console.WriteLine(\"In Try block\");
                        return;
                }
                finally
                {
                        Console.WriteLine(\"In Finally block\");
                }
        }
}
```

   Both "In Try block" and "In Finally block" will be displayed. Whether the return is in the try block or after the try-finally block, performance is not affected either way. The compiler treats it as if the return were outside the try block anyway. If it's a return without an expression (as it is above), the IL emitted is identical whether the return is inside or outside of the try. If the return has an expression, there's an extra store/load of the value of the expression (since it has to be computed within the try block).

4. **I was trying to use an "out int" parameter in one of my functions. How should I declare the variable that I am passing to it?** - You should declare the variable as an int, but when you

pass it in you must specify it as 'out', like the following: int i; foo(out i); where foo is declared as follows: [return-type] foo(out int o) { }

5. **How does one compare strings in C#?** - In the past, you had to call .ToString() on the strings when using the == or != operators to compare the strings' values. That will still work, but the C# compiler now automatically compares the values instead of the references when the == or != operators are used on string types. If you actually do want to compare references, it can be done as follows: if ((object) str1 == (object) str2) { … } Here's an example showing how string compares work:

```
using System;
public class StringTest
{
        public static void Main(string[] args)
        {
                Object nullObj = null; Object realObj = new StringTest();
                int i = 10;
                Console.WriteLine(\"Null Object is [\" + nullObj + \"]\n\"
                        + \"Real Object is [\" + realObj + \"]\n\"
                        + \"i is [\" + i + \"]\n\");
                        // Show string equality operators
                string str1 = \"foo\";
                string str2 = \"bar\";
                string str3 = \"bar\";
                Console.WriteLine(\"{0} == {1} ? {2}\", str1, str2, str1 ==
str2 );
                Console.WriteLine(\"{0} == {1} ? {2}\", str2, str3, str2 ==
str3 );
        }
}
```

Output:

```
Null Object is []
Real Object is [StringTest]
i is [10]
foo == bar ? False
bar == bar ? True
```

6. **How do you specify a custom attribute for the entire assembly (rather than for a class)?** - Global attributes must appear after any top-level using clauses and before the first type or namespace declarations. An example of this is as follows:

```
using System;
[assembly : MyAttributeClass] class X {}
```

Note that in an IDE-created project, by convention, these attributes are placed in AssemblyInfo.cs.

7. **How do you mark a method obsolete?** -

```
[Obsolete] public int Foo() {...}
```

or

```
[Obsolete(\"This is a message describing why this method is obsolete\")]
public int Foo() {...}
```

Note: The O in Obsolete is always capitalized.

8. **How do you implement thread synchronization (Object.Wait, Notify,and CriticalSection) in C#?** - You want the lock statement, which is the same as Monitor Enter/Exit:

```
lock(obj) { // code }
```

translates to

```
try {
        CriticalSection.Enter(obj);
        // code
}
finally
{
        CriticalSection.Exit(obj);
}
```

9. **How do you directly call a native function exported from a DLL?** - Here's a quick example of the DllImport attribute in action:

```
using System.Runtime.InteropServices; \
class C
{
        [DllImport(\"user32.dll\")]
        public static extern int MessageBoxA(int h, string m, string c, int
type);
        public static int Main()
        {
                return MessageBoxA(0, \"Hello World!\", \"Caption\", 0);
        }
}
```

This example shows the minimum requirements for declaring a C# method that is implemented in a native DLL. The method C.MessageBoxA() is declared with the static and external modifiers, and has the DllImport attribute, which tells the compiler that the implementation comes from the user32.dll, using the default name of MessageBoxA. For more information, look at the Platform Invoke tutorial in the documentation.

10. **How do I simulate optional parameters to COM calls?** - You must use the Missing class and pass Missing.Value (in System.Reflection) for any values that have optional parameters.

C# developer interview questions

A representative of a high-tech company in United Kingdom sent this in today noting that the list was used for interviewing a C# .NET developer. Any corrections and suggestions would be forwarded to the author. I won't disclose the name of the company, since as far as I know they might still be using this test for prospective employees. Correct answers are in green color.

1) The C# keyword .int. maps to which .NET type?

1. System.Int16
2. **System.Int32**
3. System.Int64
4. System.Int128

2) Which of these string definitions will prevent escaping on backslashes in C#?

1. string s = #.n Test string.;
2. string s = ..n Test string.;
3. **string s = @.n Test string.;**
4. string s = .n Test string.;

3) Which of these statements correctly declares a two-dimensional array in C#?

1. **int[,] myArray;**
2. int[][] myArray;
3. int[2] myArray;
4. System.Array[2] myArray;

4) If a method is marked as `protected internal` who can access it?

1. Classes that are both in the same assembly and derived from the declaring class.
2. Only methods that are in the same class as the method in question.
3. Internal methods can be only be called using reflection.
4. **Classes within the same assembly, and classes derived from the declaring class.**

5) What is boxing?

a) Encapsulating an object in a value type.
b) Encapsulating a copy of an object in a value type.
c) Encapsulating a value type in an object.
d) **Encapsulating a copy of a value type in an object.**

6) What compiler switch creates an xml file from the xml comments in the files in an assembly?

1. /text
2. **/doc**
3. /xml
4. /help

7) What is a satellite Assembly?

1. A peripheral assembly designed to monitor permissions requests from an application.
2. Any DLL file used by an EXE file.
3. **An assembly containing localized resources for another assembly.**
4. An assembly designed to alter the appearance or .skin. of an application.

8) What is a delegate?

1. **A strongly typed function pointer.**
2. A light weight thread or process that can call a single method.
3. A reference to an object in a different process.
4. An inter-process message channel.

9) How does assembly versioning in .NET prevent DLL Hell?

1. The runtime checks to see that only one version of an assembly is on the machine at any one time.
2. **.NET allows assemblies to specify the name AND the version of any assemblies they need to run.**
3. The compiler offers compile time checking for backward compatibility.
4. It doesn.t.

10) Which .Gang of Four. design pattern is shown below?

```
public class A {
  private A instance;
  private A() {
  }
  public
static A Instance {
```

```
        get
        {
          if ( A == null )
            A = new A();
          return instance;
        }
      }
    }
```

1. Factory
2. Abstract Factory
3. **Singleton**
4. Builder

11) In the NUnit test framework, which attribute must adorn a test class in order for it to be picked up by the NUnit GUI?

1. TestAttribute
2. TestClassAttribute
3. **TestFixtureAttribute**
4. NUnitTestClassAttribute

12) Which of the following operations can you NOT perform on an ADO.NET DataSet?

1. A DataSet can be synchronised with the database.
2. **A DataSet can be synchronised with a RecordSet.**
3. A DataSet can be converted to XML.
4. You can infer the schema from a DataSet.

13) In Object Oriented Programming, how would you describe encapsulation?

1. The conversion of one type of object to another.
2. The runtime resolution of method calls.
3. The exposition of data.
4. **The separation of interface and implementation.**

^Back to Top

.NET deployment questions

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.

   <probing privatePath="c:\mylibs; bin\debug" />

   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.

What Great .NET Developers Ought To Know

Everyone who writes code

- Describe the difference between a Thread and a Process?
- What is a Windows Service and how does its lifecycle differ from a "standard" EXE?
- What is the maximum amount of memory any single process on Windows can address? Is this different than the maximum virtual memory for the system? How would this affect a system design?
- What is the difference between an EXE and a DLL?
- What is strong-typing versus weak-typing? Which is preferred? Why?
- Corillian's product is a "Component Container." Name at least 3 component containers that ship now with the Windows Server Family.
- What is a PID? How is it useful when troubleshooting a system?
- How many processes can listen on a single TCP/IP port?
- What is the GAC? What problem does it solve?

Mid-Level .NET Developer

- Describe the difference between Interface-oriented, Object-oriented and Aspect-oriented programming.
- Describe what an Interface is and how it's different from a Class.

- What is Reflection?
- What is the difference between XML Web Services using ASMX and .NET Remoting using SOAP?
- Are the type system represented by XmlSchema and the CLS isomorphic?
- Conceptually, what is the difference between early-binding and late-binding?
- Is using Assembly.Load a static reference or dynamic reference?
- When would using Assembly.LoadFrom or Assembly.LoadFile be appropriate?
- What is an Asssembly Qualified Name? Is it a filename? How is it different?
- Is this valid? Assembly.Load("foo.dll");
- How is a strongly-named assembly different from one that isn't strongly-named?
- Can DateTimes be null?
- What is the JIT? What is NGEN? What are limitations and benefits of each?
- How does the generational garbage collector in the .NET CLR manage object lifetime? What is non-deterministic finalization?
- What is the difference between Finalize() and Dispose()?
- How is the using() pattern useful? What is IDisposable? How does it support deterministic finalization?
- What does this useful command line do? tasklist /m "mscor*"
- What is the difference between in-proc and out-of-proc?
- What technology enables out-of-proc communication in .NET?
- When you're running a component within ASP.NET, what process is it running within on Windows XP? Windows 2000? Windows 2003?

Senior Developers/Architects

- What's wrong with a line like this? DateTime.Parse(myString);
- What are PDBs? Where must they be located for debugging to work?
- What is cyclomatic complexity and why is it important?
- Write a standard lock() plus "double check" to create a critical section around a variable access.
- What is FullTrust? Do GAC'ed assemblies have FullTrust?
- What benefit does your code receive if you decorate it with attributes demanding specific Security permissions?
- What does this do? gacutil /l | find /i "Corillian"
- What does this do? sn -t foo.dll
- What ports must be open for DCOM over a firewall? What is the purpose of Port 135?
- Contrast OOP and SOA. What are tenets of each?
- How does the XmlSerializer work? What ACL permissions does a process using it require?
- Why is catch(Exception) almost always a bad idea?
- What is the difference between Debug.Write and Trace.Write? When should each be used?
- What is the difference between a Debug and Release build? Is there a significant speed difference? Why or why not?
- Does JITting occur per-assembly or per-method? How does this affect the working set?
- Contrast the use of an abstract base class against an interface?
- What is the difference between a.Equals(b) and a == b?
- In the context of a comparison, what is object identity versus object equivalence?
- How would one do a deep copy in .NET?
- Explain current thinking around IClonable.

- What is boxing?
- Is string a value type or a reference type?
- What is the significance of the "PropertySpecified" pattern used by the XmlSerializer? What problem does it attempt to solve?
- Why are out parameters a bad idea in .NET? Are they?
- Can attributes be placed on specific parameters to a method? Why is this useful?

C# Component Developers

- Juxtapose the use of override with new. What is shadowing?
- Explain the use of virtual, sealed, override, and abstract.
- Explain the importance and use of each component of this string: Foo.Bar, Version=2.0.205.0, Culture=neutral, PublicKeyToken=593777ae2d274679d
- Explain the differences between public, protected, private and internal.
- What benefit do you get from using a Primary Interop Assembly (PIA)?
- By what mechanism does NUnit know what methods to test?
- What is the difference between: catch(Exception e){throw e;} and catch(Exception e){throw;}
- What is the difference between typeof(foo) and myFoo.GetType()?
- Explain what's happening in the first constructor: public class c{ public c(string a) : this() {;}; public c() {;} } How is this construct useful?
- What is *this*? Can this be used within a static method?

ASP.NET (UI) Developers

- Describe how a browser-based Form POST becomes a Server-Side event like Button1_OnClick.
- What is a PostBack?
- What is ViewState? How is it encoded? Is it encrypted? Who uses ViewState?
- What is the <machinekey> element and what two ASP.NET technologies is it used for?
- What three Session State providers are available in ASP.NET 1.1? What are the pros and cons of each?
- What is Web Gardening? How would using it affect a design?
- Given one ASP.NET application, how many application objects does it have on a single proc box? A dual? A dual with Web Gardening enabled? How would this affect a design?
- Are threads reused in ASP.NET between reqeusts? Does every HttpRequest get its own thread? Should you use Thread Local storage with ASP.NET?
- Is the [ThreadStatic] attribute useful in ASP.NET? Are there side effects? Good or bad?
- Give an example of how using an HttpHandler could simplify an existing design that serves Check Images from an .aspx page.
- What kinds of events can an HttpModule subscribe to? What influence can they have on an implementation? What can be done without recompiling the ASP.NET Application?
- Describe ways to present an arbitrary endpoint (URL) and route requests to that endpoint to ASP.NET.
- Explain how cookies work. Give an example of Cookie abuse.
- Explain the importance of HttpRequest.ValidateInput()?
- What kind of data is passed via HTTP Headers?
- Juxtapose the HTTP verbs GET and POST. What is HEAD?
- Name and describe at least a half dozen HTTP Status Codes and what they express to the requesting client.
- How does if-not-modified-since work? How can it be programmatically implemented with

ASP.NET?
Explain <@OutputCache%> and the usage of VaryByParam, VaryByHeader.

- How does VaryByCustom work?
- How would one implement ASP.NET HTML output caching, caching outgoing versions of pages generated via all values of q= except where q=5 (as in http://localhost/page.aspx?q=5)?

Developers using XML

- What is the purpose of XML Namespaces?
- When is the DOM appropriate for use? When is it not? Are there size limitations?
- What is the WS-I Basic Profile and why is it important?
- Write a small XML document that uses a default namespace and a qualified (prefixed) namespace. Include elements from both namespace.
- What is the one fundamental difference between Elements and Attributes?
- What is the difference between Well-Formed XML and Valid XML?
- How would you validate XML using .NET?
- Why is this almost always a bad idea? When is it a good idea? myXmlDocument.SelectNodes("//mynode");
- Describe the difference between pull-style parsers (XmlReader) and eventing-readers (Sax)
- What is the difference between XPathDocument and XmlDocument? Describe situations where one should be used over the other.
- What is the difference between an XML "Fragment" and an XML "Document."
- What does it meant to say “the canonical” form of XML?
- Why is the XML InfoSet specification different from the Xml DOM? What does the InfoSet attempt to solve?
- Contrast DTDs versus XSDs. What are their similarities and differences? Which is preferred and why?
- Does System.Xml support DTDs? How?
- Can any XML Schema be represented as an object graph? Vice versa?