1. **What Is The Microsoft.net?**

.NET is a set of technologies designed to transform the internet into a full scale distributed platform. It provides new ways of connecting systems, information and devices through a collection of web services.It also provides a language independent, consistent programming model across all tiers of an application.

The goal of the .NET platform is to simplify web development by providing all of the tools and technologies that one needs to build distributed web applications

1. **What Is The .net Framework?**

The .NET Framework is set of technologies that form an integral part of the .NET Platform. It is Microsoft's managed code programming model for building applications that have visually stunning user experiences, seamless and secure communication, and the ability to model a range of business processes.

The .NET Framework has two main components: the common language runtime (CLR) and .NET Framework class library. The CLR is the foundation of the .NET framework and provides a common set of services for projects that act as building blocks to build up applications across all tiers. It simplifies development and provides a robust and simplified environment which provides common services to build application. The .NET framework class library is a collection of reusable types and exposes features of the runtime. It contains of a set of classes that is used to access common functionality

1. **What Is Clr?**

The .NET Framework provides a runtime environment called the Common Language Runtime or CLR. The CLR can be compared to the Java Virtual Machine or JVM in Java. CLR handles the execution of code and provides useful services for the implementation of the program. In addition to executing code, CLR provides services such as memory management, thread management, security management, code verification, compilation, and other system services. It enforces rules that in turn provide a robust and secure execution environment for .NET applications

1. **What Is Cts?**

Common Type System (CTS) describes the datatypes that can be used by managed code. CTS defines how these types are declared, used and managed in the runtime. It facilitates cross-language integration, type safety, and high performance code execution. The rules defined in CTS can be used to define your own classes and values.

1. **What Is Cls?**

Common Language Specification (CLS) defines the rules and standards to which languages must adhere to in order to be compatible with other .NET languages. This enables C# developers to inherit from classes defined in VB.NET or other .NET compatible languages

1. **What Is Managed Code?**

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. The code that runs within the common language runtime is called managed code

1. **What Is Msil?**

When the code is compiled, the compiler translates your code into Microsoft intermediate language (MSIL). The common language runtime includes a JIT compiler for converting this MSIL then to native code.

MSIL contains metadata that is the key to cross language interoperability. Since this metadata is standardized across all .NET languages, a program written in one language can understand the metadata and execute code, written in a different language.MSIL includes instructions for loading, storing,initializing, and calling methods on objects,as well as instructions for arithmetic and logical operations,control flow,direct memory access, exception handling, and other operations

1. **What Is Jit?**

JIT is a compiler that converts MSIL to native code. The native code consists of hardware specific instructions that can be executed by the CPU.

Rather than converting the entire MSIL (in a portable executable[PE]file) to native code, the JIT converts the MSIL as it is needed during execution. This converted native code is stored so that it is accessible for subsequent calls

1. **What Is Portable Executable (pe)?**

PE is the file format defining the structure that all executable files (EXE) and Dynamic Link Libraries (DLL) must use to allow them to be loaded and executed by Windows. PE is derived from the Microsoft Common Object File Format (COFF). The EXE and DLL files created using the .NET Framework obey the PE/COFF formats and also add additional header and data sections to the files that are only used by the CLR.

1. **What Is An Application Domain?**

Application domain is the boundary within which an application runs. A process can contain multiple application domains. Application domains provide an isolated environment to applications that is similar to the isolation provided by processes. An application running inside one application domain cannot directly access the code running inside another application domain. To access the code running in another application domain, an application needs to use a proxy

1. **How Does An Appdomain Get Created?**

AppDomains are usually created by hosts. Examples of hosts are the 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 AppDomain for every application. AppDomains can also be explicitly created by .NET applications

1. **What Is An Assembly?**

An assembly is a collection of one or more .exe or dll’s. An assembly is the fundamental unit for application development and deployment in the .NET Framework. An assembly contains a collection of types and resources that are built to work together and form a logical unit of functionality. An assembly provides the CLR with the information it needs to be aware of type implementations

1. **What Are The Contents Of Assembly?**

A static assembly can consist of four elements:

* + Assembly manifest-Contains the assembly metadata. An assembly manifest contains the information about the identity and version of the assembly. It also contains the information required to resolve references to types and resources.
  + Type metadata-Binary information that describes a program.
  + Microsoft intermediate language(MSIL)code.
  + A set of resources

1. **What Are The Different Types Of Assembly?**

Assemblies can also be private or shared. A private assembly is installed in the installation directory of an application and is accessible to that application only. On the other hand, a shared assembly is shared by multiple applications. A shared assembly has a strong name and is installed in the GAC.

We also have satellite assemblies that are often used to deploy language-specific resources for an application.

1. **What Is A Dynamic Assembly?**

A dynamic assembly is created dynamically at run time when an application requires the types within these assemblies

1. **What Is A Strong Name?**

You need to assign a strong name to an assembly to place it in the GAC and make it globally accessible. A strong name consists of a name that consists of an assembly's identity (text name, version number, and culture information), a public key and a digital signature generated over the assembly. The .NET Framework provides a tool called the Strong Name Tool (Sn.exe), which allows verification and key pair and signature generation

1. **What Is Gac? What Are The Steps To Create An Assembly And Add It To The Gac?**

The global assembly cache (GAC) is a machine-wide code cache that 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.

Steps

-Create a strong name using sn.exe tool eg: sn -k mykey.snk  
-in AssemblyInfo.cs, add the strong name eg: [assembly: AssemblyKeyFile("mykey.snk")]  
-recompile project, and then install it to GAC in two ways:  
drag & drop it to assembly folder (C:\WINDOWS\assembly OR C:\WINNT\assembly) (shfusion.dll tool)

gacutil -i abc.dll

1. **What Is The Caspol.exe Tool Used For?**

The caspol tool grants and modifies permissions to code groups at the user policy, machine policy, and enterprise policy levels.

1. **What Is A Garbage Collector?**

A garbage collector performs periodic checks on the managed heap to identify objects that are no longer required by the program and removes them from memory.

1. **What Are Generations And How Are They Used By The Garbage Collector?**

Generations are the division of objects on the managed heap used by the garbage collector. This mechanism allows the garbage collector to perform highly optimized garbage collection. The unreachable objects are placed in generation 0, the reachable objects are placed in generation 1, and the objects that survive the collection process are promoted to higher generations

1. **What Is Ilasm.exe Used For?**

Ilasm.exe is a tool that generates PE files from MSIL code. You can run the resulting executable to determine whether the MSIL code performs as expected.

1. **What Is Ildasm.exe Used For?**

Ildasm.exe is a tool that takes a PE file containing the MSIL code as a parameter and creates a text file that contains managed code.

1. **What Is The Resgen.exe Tool Used For?**

ResGen.exe is a tool that is used to convert resource files in the form of .txt or .resx files to common language runtime binary .resources files that can be compiled into satellite assemblies.

1. **What Is Boxing And Unboxing?**

==Boxing  
Value type to object type. Allocates memory on Heap.

==UnBoxing  
Object type to value type. Allocates memory on Stack

1. **How Can We Make A Thread Sleep For Infinite Period ?**

You can also place a thread into the sleep state for an indeterminate amount of time by calling Thread.Sleep (System.Threading.Timeout.Infinite). To interrupt this sleep you can call the Thread.Interrupt method

1. **In Which Format You Can Pass The Value In The Sleep Function?**

In milliseconds

1. **What's Thread.sleep() In Threading ?**

Thread's execution can be paused by calling the Thread.Sleep method. This method takes an integer value that determines how long the thread should sleep. Example Thread.CurrentThread.Sleep(2000). it will paused for 2 second.

1. **How Can You Reference Current Thread Of The Method ?**

"Thread.CurrentThread" refers to the current thread running in the method."CurrentThread" is a public static property.

1. **What Does Addressof Operator Do In Background ?**

The AddressOf operator creates a delegate object to the BackgroundProcess method. A delegate within VB.NET is a type-safe, object-oriented function pointer. After the thread has been instantiated, you begin the execution of the code by calling the Start() method of the thread

1. **Different Levels Of Priority Provided By .net.**

i)ThreadPriority.Highest  
ii)ThreadPriority.AboveNormal  
iii)ThreadPriority.Normal  
iv)ThreadPriority.BelowNormal  
v)ThreadPriority.Lowest

1. **Is There Any Thread In Our .net Programs?**

.NET program always has at least two threads running one is the main program and second is the garbage collector.

1. **Namespace For The Thread Class?**

All threading classes are defined in System.Threading namespace

1. **Can We Have Multiple Threads In One App Domain ?**

One or more threads run in an AppDomain. An AppDomain is a runtime representation of a logical process within a physical process. Each AppDomain is started with a single thread, but can create additional threads from any of its threads.

1. **Did Vb6 Support Multi-threading ?**

While VB6 supports multiple single-threaded apartments, it does not support a freethreading model, which allows multiple threads to run against the same set of data.

1. **What Is A Thread ?**

A thread is the basic unit to which the operating system allocates processor time

1. **What Is Multi-threading ?**

Multi-threading forms subset of Multi-tasking. Instead of having to switch between programs this feature switches between different parts of the same program. Example you are writing in word and at the same time word is doing a spell check in background.

1. **What Is Multi-tasking ?**

It's a feature of modern operating systems with which we can run multiple programs at same time example Word, Excel etc.

1. **What Is Equivalent For Regsvr32 Exe In .net ?**

Regasm

1. **How Do We Create Dcom Object In Vb6?**

Using the CreateObject method you can create a DCOM object. You have to put the server name in the registry.

1. **Can You Explain What Is Dcom ?**

DCOM differs from COM in that it allows for creating objects distributed across a network, a protocol for invoking that object's methods, and secures access to the object. DCOM provides a wrapper around COM, hence it is a backwards compatible extension. DCOM uses Remote Procedural Calls (RPC) using Open Software Foundation's Distributed Computing Environment.

1. **What Is Satellite Assembly? And Steps To Create Satellite Assembly?**

When you write a multilingual or multi-cultural application in .NET, and want to distribute the core application separately from the localized modules, the localized assemblies that modify the core application are called satellite assemblies.

Steps to Create Satellite Assembly  
a. Set the paths for resgen and al.exe:  
b. Create a .resources file.  
c. Create the satellite assembly.  
d. The assembly should have the naming convention for .NET to be able to search for it.  
e. Specify the settings for culture.  
f. Put the satellite assembly in the appropriate folder.  
g. Once the satellite assembly is created, physically copy it to the appropriate directory.  
h. Repeat the process for each language in which you are creating an assembly.

1. **How To Exclude A Property From Xml Serialization?**

[XmlIgnore]

(use attribute on the property)

Eg:

[XmlIgnore]

public string Name

{

..

}

1. **What Is Garbage Collector ?**

Garbage collection consists of the following steps:

* + The garbage collector searches for managed objects that are referenced in managed code.
  + The garbage collector attempts to finalize objects that are not referenced.
  + The garbage collector frees objects that are not referenced and reclaims their memory.

1. **What Is Difference Between Code Access And Role Based Security?**

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 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.

Neither is better. It depends on the nature of the application; both code-based and role-based security could be implemented to an extent.

1. **What Is Cas?**

The CAS security policy revolves around two key concepts-code groups and permissions. Each .NET assembly is a member of a particular code group, and each code group is granted the permissions specified in a named permission set.

See CAS objects -- Run 'caspol -lg' from command line.  
Add CAS objects -- caspol -ag 1.3   
Change CAS obj -- caspol -cg 1.3 FullTrust  
Turn Off -- caspol -s off

1. **Can We Customize The Serialization Process?**

Yes. XmlSerializer supports a range of attributes that can be used to configure serialization for a particular class. For example, a field or property can be marked with the [XmlIgnore] attribute to exclude it from serialization. Another example is the [XmlElement] attribute, which can be used to specify the XML element name to be used for a particular property or field.

Serialization via SoapFormatter/BinaryFormatter can also be controlled to some extent by attributes. For example, the [NonSerialized] attribute is the equivalent of XmlSerializer's [XmlIgnore] attribute. Ultimate control of the serialization process can be acheived by implementing the the ISerializable interface on the class whose instances are to be serialized.

1. **Types Of Serialization**

1. Binary Serialization - preserves type fidelity, which is useful for preserving the state of an object between different invocations of an application.  
2. 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.

1. **What Is Serialization?**

It is 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 de-serialized, an exact clone of the original object is created.

* + used to save session state in ASP.NET.
  + Copy objects to the Clipboard in Windows Forms
  + Remoting to pass objects by value from one application domain to another.

1. **What Is Namespace?**

It is a logical naming scheme for types in which a simple type name, such as MyType, is preceded with a dot-separated hierarchical name. Such a naming scheme is completely under control of the developer. This concept is not related to that of an assembly.

1. **What Is Gac? How To Put Assembly In Gac?**

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. Assemblies deployed in the global assembly cache must have a strong name. When an assembly is added to the global assembly cache, integrity checks are performed on all files that make up the assembly. The cache performs these integrity checks to ensure that an assembly has not been tampered with, for example, when a file has changed but the manifest does not reflect the change. Use a developer tool called the Global Assembly Cache tool (Gacutil.exe), provided by the .NET Framework SDK or Use Windows Explorer to drag assemblies into the cache. To install a strong-named assembly into the global assembly cache At the command prompt, type the following command: gacutil I In this command, assembly name is the name of the assembly to install in the global assembly cache.

1. **What Is An Assembly Loader?**

-Checks if the assembly is Strongly signed.

-If yes it will search in the GAC

-Loader will search the policy file name in the format of Policy. Assembly Major Version .Assembly Minor Version .Assembly Name

Eg. MyPolicy.1.2.Assembly1

-If such a file exists it will look inside of it if the version of the assembly that we are trying to load matches the version/versions range written in the policy file. If it does, it will try to load the assembly with the version specified there. If no such policy file exists, it will try to load the assembly from the GAC.

-If it will fail to find it in the GAC, it will start to search in the system's search path.  
In web applications it will also include the application's Bin directory in the search path.

1. **What Is An Assembly Qualified Name**

An assembly qualified name isn't the filename of the assembly; it's the internal name of the assembly combined with the assembly version, culture, and public key, thus making it unique.

Example:

(""System .Xml .Xml Document, System .Xml, Version =1.0.3300.0, Culture =neutral, PublicKey Token = b77 a5c561934 e089"")

1. **What Is Partial Assembly Reference**

We can dynamically reference an assembly by providing only partial information, such as specifying only the assembly name. When you specify a partial assembly reference, the runtime looks for the assembly only in the application directory.

We can make partial references to an assembly in your code one of the following ways:

-> Use a method such as System.Reflection.Assembly.Load and specify only a partial reference. The runtime checks for the assembly in the application directory.

-> Use the System.Reflection.Assembly.LoadWithPartialName method and specify only a partial reference. The runtime checks for the assembly in the application directory and in the global assembly cache.

1. **What Is Full Assembly Reference**

A full assembly reference includes the assembly's text name, version, culture, and public key token (if the assembly has a strong name). A full assembly reference is required if you reference any assembly that is part of the common language runtime or any assembly located in the global assembly cache.

1. **What Are The Contents Of An Assembly ?**
   * Type metadata.
   * The assembly manifest, which contains assembly metadata.
   * Microsoft intermediate language (MSIL) code that implements the types.
   * A set of resources
2. **What Are The Types Of Assembly Available**

1.Private-The assembly is intended only for one application  
2.Shared-If the assembly is to be made into a Shared Assembly  
3.Static-These are the .NET PE files that you create at compile time.  
4.Dynamic-These are PE-formatted, in-memory assemblies

1. **What Are Attributes?**

Attributes are declarative tags in code that insert additional metadata into an assembly. There exist two types of attributes in the .NET Framework: Predefined attributes such as Assembly Version, which already exist and are accessed through the Runtime Classes; and custom attributes, which you write yourself by extending the System.Attribute class.

Example-Custom Attribute for entire assembly

1.using System;  
2.[assembly : MyAttributeClass] class X {}

1. **What Is Manifest?**

An assembly manifest contains all the metadata needed to specify the assembly's version requirements and security identity, and all metadata needed to define the scope of the assembly and resolve references to resources and classes. The assembly manifest can be stored in either a PE (Portable Executable) file (an .exe or .dll) with Microsoft intermediate language (MSIL) code or in a standalone PE (Portable Executable) file that contains only assembly manifest information.

1. **What Is Jit Compiler**

It is a compiler which converts MS IL code to Native Code (ie.. CPU-specific code that runs on the same computer architecture). Just-In-Time compiler- it converts the language that you write in .Net into machine language that a computer can understand.

1. **What Debugging Tools Come With The .net Sdk?**

1.CorDBG - command-line debugger. To use CorDbg, you must compile the original C# file using the /debug switch.  
2.DbgCLR - graphic debugger. Visual Studio .NET uses the DbgCLR

1. **What's The .net Collection Class That Allows An Element To Be Accessed Using A Unique Key?**

HashTable.

1. **What Is Cls**

It is the collection of the rules and constraints that every language (that seeks to achieve .NET compatibility) must follow. It is a subsection of CTS and it specifies how it shares and extends one another libraries.

1. **What Is Clr**

It is the runtime that converts a MSIL code into the host machine language code, which is then executed appropriately. It is the execution engine for .NET Frameworkapplications. It provides a number of services, including:

* + Code management (loading and execution)
  + Application memory isolation
  + Verification of type safety
  + Conversion of IL to native code.
  + Access to metadata (enhanced type information)
  + Managing memory for managed objects
  + Enforcement of code access security
  + Exception handling, including cross-language exceptions
  + Interoperation between managed code, COM objects, and pre-existing DLL's (unmanaged code and data)
  + Automation of object layout
  + Support for developer services (profiling, debugging, and so on}.

1. **What Is Dependency Injection?**

It is the mechanism through with dynamic capabilities are added in the runtime. Most of the time it would be through proxy classes.

Some frameworks in this regard are:  
Spring.Net, NInject, Enterprise Library Policy Injection etc.

1. **Can You Describe Iuknown Interface In Short?**

Every COM object supports at least one interface, the IUnknown interface.

All interfaces are classes derived from the base class IUnknown. Each interface supports methods access data and perform operations transparently to the programmer.

For example, IUnknown supports three methods, AddRef, Release(), and QueryInterface().  
Suppose that pinterf is a pointer to an IUnknown.

pinterf->AddRef() increments the reference count. pinterf->Release() decrements the reference count, deleting the object when the reference count reaches zero.

pinterf->QueryInterface( IDesired,pDesired) checks to see if the current interface (IUnknown) supports another interface, IDesired, creates an instance (via a call to CoCreateInstance()) of the object if the reference count is zero (the object does not yet exist), and then calls pDesired->AddRef() to increment the reference count(where pDesired is a pointer to IDesired) and returns the pointer to the caller.

1. **What Is Reference Counting In Com?**

Reference counting is a memory management technique used to count how many times an object has a pointer referring to it. The first time it is created, the reference count is set to one. When the last reference to the object is nulled, the reference count is set to zero and the object is deleted.

Care must be exercised to prevent a context switch from changing the reference count at the time of deletion. In the methods that follow, the syntax is shortened to keep the scope of the discussion brief and manageable.

1. **What Is Com?**

Microsoft's COM is a technology for component software development. It is a binary standard which is language independent. DCOM is a distributed extension of COM.

1. **When We Use Windows Api In .net Is It Managed Or Unmanaged Code?**

Windows API in .NET is unmanaged code

1. **Once I Have Developed The Com Wrapper Do I Have To Still Register The Com In Registry?**

Yes

1. **In Which Order The Destructor Is Called For An Inherited Class?**

Child class destructor is called first

1. **In Which Order The Constructor Is Called For An Inherited Class?**

Constructor of Parent class is called

Then,

Constructor of Child class

1. **What Are Value Types And Reference Types ?**

Value types directly contain their data which are either allocated on the stack or allocated in-line in a structure.

Reference types store a reference to the value's memory address, and are allocated on the heap.  
Reference types can be self-describing types, pointer types, or interface types.

Variables that are value types each have their own copy of the data, and therefore operations on one variable do not affect other variables. Variables that are reference types can refer to the same object; therefore, operations on one variable can affect the same object referred to by another variable. All types derive from the System.Object base type.

1. **What Is The Difference Between Viewstate And Sessionstate?**

ViewState persist the values of controls of particular page in the client (browser) when post back operation done. When user requests another page previous page data no longer available.

SessionState persist the data of particular user in the server. This data available till user close the browser or session time completes.

ViewState Relate to Controls, it means when Client send request in the form of Controls' value to server,In case any validation problem relating to Clients data then the entire data is not cleaned as it was occur is asp,it can restore the existing data in the form of hidden control and render data by server to respective controls,as a result user will not enter the complete information again, where as SessionState relates to individual user,it manages the user related information and maintain the session between client ans server.

1. **Difference Between Machine.config And Web.config?**

Machine.config->Configuration file for all the applications in the system.  
Web.config->Config file for single application.

1. **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.

1. **What Is View State?**

The web is stateless. But in ASP.NET, the state of a page is maintained in the page itself automatically.  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.

1. **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.

1. **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.

1. **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.

1. **How To Get The Sum Of Last 3 Items In A List Using Lambda Expressions?**

Use  
int sum = list.Reverse().Take(3).Sum();

1. **You Have Got 1 Million Parking Slots. At A Time A Parking Slot Can Be Free Or Not. To Get Next Slot Easily Which Data Structure To Implement?**

Use Stack.

If you use Stack, we can just get the next slot by using stack.Pop()

If you use List, we have to iterate through all list and check the status and retrieve. So stack would be advantageous.

1. **Why Can't Struct Be Used Instead Of Class For Storing Entity?**

Struct is of value type. If we pass it across methods in layers, a new object will be created in the stack thus increasing memory and processing requirement.

So class should be used for creating entity like Employee, Supplier etc.

1. **What Is Assembly Version Series Sequence?**

major.minor.build.revision

1. **What Tool We Have To Use To Install Assembli In Gac Folder.**

we can use GACUTIL Tool to intall assemby in GAC folder

1. **Ho We Can See Assembly Information?**

We can see the assembly information using ILDASM Tool.

1. **What Is Private And Shared Assembly?**

The assembly which is used only by a single application is called as private assembly. Thus the assembly is private to your application.Suppose that you are creating a general purpose DLL which provides functionality which will be used by variety of applications. Now, instead of each client application having its own copy of DLL you can place the DLL in 'global assembly cache'. Such assemblies are called as shared assemblies.

1. **What Is Type Safety?**

Type safe code can access only the memory locations that it has permission to execute. Type safe code can never access any private members of an object. Type safe code ensures that objects are isolated from each other.

1. **What Is The Difference Between Datareader And Dataadapter?**

1. Data Reader is read only forward only and much faster than DataAdapter.  
2. If you use DataReader you have to open and close connection explicitly where as if you use DataAdapter the connection is automatically opened and closed.  
3. DataReader is connection oriented where as Data Adapter is disconnected

1. **What Is Func In .net 3.5?**

Func is a delegate in .Net that returns a value.

The return type is specified using the TResult argument.

There are 5 versions of Func in .Net.

Func<TResult>  
Func<T,TResult>  
Func<T1,T2,TResult>  
Func<T1,T2,T3,TResult>  
Func<T1,T2,T3,T4,TResult>

1. **What Is Action In C# 3.5?**

Action is a built-in delegate which returns void.

There are 5 versions of Action

Action Action<T>  
Action<T1,T2>  
Action<T1,T2,T3>  
Action<T1,T2,T3,T4>

1. **What Is The Advantage Of Mvc?**

More Manageability, Multiple View Support, More unit testing compatibility.

1. **Advantages Of Vb.net And C#**

Advantages VB.NET :-

\* Has support for optional parameters which makes COM interoperability much easy.  
\* With Option Strict off late binding is supported.Legacy VB functionalities can be used by using Microsoft.VisualBasic namespace.  
\* Has the WITH construct which is not in C#.  
\* The VB.NET part of Visual Studio .NET compiles your code in the background.

While this is considered an advantage for small projects, people creating very large projects have found that the IDE slows down considerably as the project gets larger.

Advantages of C#

\* XML documentation is generated from source code but this is now been incorporated in Whidbey.  
\*Operator overloading which is not in current VB.NET but is been introduced in Whidbey.  
\* The using statement, which makes unmanaged resource disposal simple.  
\* Access to Unsafe code. This allows pointer arithmetic etc, and can improve performance in some situations. However, it is not to be used lightly, as a lot of the normal safety of C# is lost (as the name implies).This is the major difference that you can access unmanaged code in C# and not in VB.NET

1. **What Is Garbage Collection?**

Garbage collection is a CLR feature which automatically manages memory. Programmers forget to release the objects while coding. CLR automatically releases objects when they are no longer referenced and in use. CLR runs on non-deterministic to see the unused objects and cleans them. One side effect of this non-deterministic feature is that we cannot assume an object is destroyed when it goes out of the scope of a function. Therefore, we should not put code into a class destructor to release resources.

1. **How Many Types Of Stored Procedures Are There In Sql Server?**

1.User Defined Stored Procedures.  
a.Transact-SQL stored procedure  
b.CLR Stored Procedure.

2.System Stored Procedures

1. **What Is The Purpose Of Linked Server Configuration In Sql Server?**

It enabled us to query remote database servers of different providers.

Thus we can copy or select table data between multiple servers.

1. **What Is Difference Between Tostring() Vs Convert.tostring() Vs (string) Cast**

There is a simple but important difference between these three.

ToString() raise exception when the object is null

So in the case of object.ToString(), if object is null, it raise NullReferenceException.

Convert.ToString() return string.Empty in case of null object

(string) cast assign the object in case of null

So in case of  
MyObject o = (string)NullObject;

But when you use o to access any property, it will raise NullReferenceException.

1. **In How Many Ways You Can Invoke Ssrs Reports?**

you can invoke it in 3 ways.  
1.Using rs script and command Prompt  
2.From web browser.  
3.Using SOAP API

1. **What Is A Static Constructor?**

A Static Constructor of a class gets initialised and invoked when the assembly loads itself.

1. **Why Do I Get Errors When I Try To Serialize A Hashtable?**

XmlSerializer will refuse to serialize instances of any class that implements IDictionary, e.g. Hashtable. SoapFormatter and BinaryFormatter do not have this restriction.

1. **Why Is Xmlserializer So Slow?**

There is a once-per-process-per-type overhead with XmlSerializer. So the first time you serialize or deserialize an object of a given type in an application, there is a significant delay. This normally doesn't matter, but it may mean, for example, that XmlSerializer is a poor choice for loading configuration settings during startup of a GUI application.

1. **Can I Customise The Serialization Process?**

Yes. XmlSerializer supports a range of attributes that can be used to configure serialization for a particular class. For example, a field or property can be marked with the [XmlIgnore] attribute to exclude it from serialization. Another example is the [XmlElement] attribute, which can be used to specify the XML element name to be used for a particular property or field. Serialization via SoapFormatter/BinaryFormatter can also be controlled to some extent by attributes. For example, the [NonSerialized] attribute is the equivalent of XmlSerializer's [XmlIgnore] attribute. Ultimate control of the serialization process can be acheived by implementing the the ISerializable interface on the class whose instances are to be serialized.

1. **Does The .net Framework Have In-built Support For Serialization?**

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

1. **Is The Lack Of Deterministic Destruction In .net A Problem?**

It's certainly an issue that affects component design. If you have objects that maintain expensive or scarce resources (e.g. database locks), you need to provide some way for the client to tell the object to release the resource when it is done. Microsoft recommend that you provide a method called Dispose() for this purpose. However, this causes problems for distributed objects - in a distributed system who calls the Dispose() method? Some form of reference-counting or ownership-management mechanism is needed to handle distributed objects - unfortunately the runtime offers no help with this

1. **Why Doesn't The .net Runtime Offer Deterministic Destruction?**

Because of the garbage collection algorithm. The .NET garbage collector works by periodically running through a list of all the objects that are currently being referenced by an application. All the objects that it doesn't find during this search are ready to be destroyed and the memory reclaimed. The implication of this algorithm is that the runtime doesn't get notified immediately when the final reference on an object goes away - it only finds out during the next sweep of the heap.

Futhermore, this type of algorithm works best by performing the garbage collection sweep as rarely as possible. Normally heap exhaustion is the trigger for a collection sweep.

1. **How Does Assembly Versioning Work?**

Each assembly has a version number called the compatibility version. Also each reference to an assembly (from another assembly) includes both the name and version of the referenced assembly.The version number has four numeric parts (e.g. 5.5.2.33). Assemblies with either of the first two parts different are normally viewed as incompatible. If the first two parts are the same, but the third is different, the assemblies are deemed as 'maybe compatible'. If only the fourth part is different, the assemblies are deemed compatible. However, this is just the default guideline - it is the version policy that decides to what extent these rules are enforced. The version policy can be specified via the application configuration

1. **How Do Assemblies Find Each Other?**

By searching directory paths. There are several factors which can affect the path (such as the AppDomain host, and application configuration files), but for private assemblies the search path is normally the application's directory and its sub-directories. For shared assemblies, the search path is normally same as the private assembly path plus the shared assembly cache.

1. **Where Is The Output Of Textwritertracelistener Redirected?**

To the Console or a text file depending on the parameter passed to the constructor.

1. **Why Are There Five Tracing Levels In System.diagnostics.traceswitcher?**

The tracing dumps can be quite verbose. For applications that are constantly running you run the risk of overloading the machine and the hard drive. Five levels range from None to Verbose, allowing you to fine-tune the tracing activities.

1. **What's The Difference Between The Debug Class And Trace Class?**

Documentation looks the same. Use Debug class for debug builds, use Trace class for both debug and release builds.

1. **What Does Assert() Method Do?**

In debug compilation, assert takes in a Boolean condition as a parameter, and shows the error dialog if the condition is false. The program proceeds without any interruption if the condition is true.

1. **Why String Are Called Immutable Data Type ?**

The memory representation of string is an Array of Characters, So on re-assigning the new array of Char is formed & the start address is changed . Thus keeping the Old string in Memory for Garbage Collector to be disposed.

1. **What Is Side-by-side Execution? Can Two Application One Using Private Assembly And Other Using Shared Assembly Be Stated As A Side-by-side Executables?**

Side-by-side execution is the ability to run multiple versions of an application or component on the same computer. You can have multiple versions of the common language runtime, and multiple versions of applications and components that use a version of the runtime, on the same computer at the same time. Since versioning is only applied to shared assemblies, and not to private assemblies, two application one using private assembly and one using shared assembly cannot be stated as side-by-side executables,

1. **Changes To Which Portion Of Version Number Indicates An Incompatible Change?**

Major or minor. Changes to the major or minor portion of the version number indicate an incompatible change. Under this convention then, version 2.0.0.0 would be considered incompatible with version 1.0.0.0. Examples of an incompatible change would be a change to the types of some method parameters or the removal of a type or method altogether. Build. The Build number is typically used to distinguish between daily builds or smaller compatible releases. Revision. Changes to the revision number are typically reserved for an incremental build needed to fix a particular bug. You'll sometimes hear this referred to as the "emergency bug fix" number in that the revision is what is often changed when a fix to a specific bug is shipped to a customer.

1. **What Is Partial Assembly References?**

Full Assembly reference: A full assembly reference includes the assembly's text name, version, culture, and public key token (if the assembly has a strong name). A full assembly reference is required if you reference any assembly that is part of the common language runtime or any assembly located in the global assembly cache.

Partial Assembly reference: We can dynamically reference an assembly by providing only partial information, such as specifying only the assembly name. When you specify a partial assembly reference, the runtime looks for the assembly only in the application directory.

We can make partial references to an assembly in your code one of the following ways:

-> Use a method such as System.Reflection.Assembly.Load and specify only a partial reference. The runtime checks for the assembly in the application directory.

-> Use the System.Reflection.Assembly.LoadWithPartialName method and specify only a partial reference. The runtime checks for the assembly in the application directory and in the global assembly cache

1. **What Is The Difference Between Finalize And Dispose (garbage Collection) ?**

Class instances often encapsulate control over resources that are not managed by the runtime, such as window handles (HWND), database connections, and so on. Therefore, you should provide both an explicit and an implicit way to free those resources. Provide implicit control by implementing the protected Finalize Method on an object (destructor syntax in C# and the Managed Extensions for C++). The garbage collector calls this method at some point after there are no longer any valid references to the object. In some cases, you might want to provide programmers using an object with the ability to explicitly release these external resources before the garbage collector frees the object. If an external resource is scarce or expensive, better performance can be achieved if the programmer explicitly releases resources when they are no longer being used. To provide explicit control, implement the Dispose method provided by the IDisposable Interface. The consumer of the object should call this method when it is done using the object.

Dispose can be called even if other references to the object are alive. Note that even when you provide explicit control by way of Dispose, you should provide implicit cleanup using the Finalize method. Finalize provides a backup to prevent resources from permanently leaking if the programmer fails to call Dispose.

1. **What Is Reflection?**

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.

Using reflection to access .NET metadata is very similar to using ITypeLib/ITypeInfo to access type library data in COM, and it is used for similar purposes - e.g. determining data type sizes for marshaling data across context/process/machine boundaries. Reflection can also be used to dynamically invoke methods (see System.Type.Invoke Member), or even create types dynamically at run-time. (see System. Reflection. Emit.Type Builder).

1. **What Does 'managed' Mean In The .net Context?**

The term 'managed' is the cause of much confusion. It is used in various places within .NET, meaning slightly different things.Managed code: The .NET framework provides several core run-time services to the programs that run within it - for example exception handling and security.

For these services to work, the code must provide a minimum level of information to the runtime.  
Such code is called managed code. All C# and Visual Basic.NET code is managed by default. VS7 C++ code is not managed by default, but the compiler can produce managed code by specifying a command-line switch (/com+).

Managed data: This is data that is allocated and de-allocated by the .NET runtime's garbage collector. C# and VB.NET data is always managed. VS7 C++ data is unmanaged by default, even when using the /com+ switch, but it can be marked as managed using the \_\_gc keyword.Managed classes: This is usually referred to in the context of Managed Extensions (ME) for C++. When using ME C++, a class can be marked with the \_\_gc keyword. As the name suggests, this means that the memory for instances of the class is managed by the garbage collector, but it also means more than that. The class becomes a fully paid-up member of the .NET community with the benefits and restrictions that brings. An example of a benefit is proper interop with classes written in other languages - for example, a managed C++ class can inherit from a VB class. An example of a restriction is that a managed class can only inherit from one base class.

1. **What Platforms Does The .net Framework Run On?**

The runtime supports Windows XP, Windows 2000, NT4 SP6a and Windows ME/98. Windows 95 is not supported. Some parts of the framework do not work on all platforms - for example, ASP.NET is only supported on Windows XP and Windows 2000. Windows 98/ME cannot be used for development.

IIS is not supported on Windows XP Home Edition, and so cannot be used to host ASP.NET. However, the ASP.NET Web Matrix web server does run on XP Home.

The Mono project is attempting to implement the .NET framework on Linux.

1. **When Was The First Version Of .net Released?**

The final version of the 1.0 SDK and runtime was made publicly available around 6pm PST on 15-Jan-2002. At the same time, the final version of Visual Studio.NET was made available to MSDN subscribers.

1. **When Was .net Announced?**

Bill Gates delivered a keynote at Forum 2000, held June 22, 2000, outlining the .NET 'vision'. The July 2000 PDC had a number of sessions on .NET technology, and delegates were given CDs containing a pre-release version of the .NET framework/SDK and Visual Studio.NET.

1. **What Is .net Remoting?**

Netremoting replaces DCOM. Web Services that uses remoting can run in anyApplication type i.e. Console Application, Windows Form Applications,Window Services etc. In CLR Object Remoting we can call objectsacross network.

1. **How Do You Generate A Strong Name?**

.NET provides an utility called strong name tool. You can run this toolfrom the VS.NET command prompt to generate a strong name with an option "-k" and providing the strong key file name. i.e. sn- -k < file-name >

1. **What Is The Gac? What Problem Does It Solve?**

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 that are to be shared by several applications on the computer. This area is typically the folder under windows or winnt in the machine.

All the assemblies that need to be shared across applications need to be done through the Global assembly Cache only. However it is not necessary to install assemblies into the global assembly cache to make them accessible to COM interop or unmanaged code.

There are several ways to deploy an assembly into the global assembly cache:

* + Use an installer designed to work with the global assembly cache. This is the preferred option for installing assemblies into the global assembly cache.
  + Use a developer tool called the Global Assembly Cache tool (Gacutil.exe), provided by the .NET Framework SDK.
  + Use Windows Explorer to drag assemblies into the cache.
  + GAC solves the problem of DLL Hell and DLL versioning. Unlike earlier situations, GAC can hold two assemblies of the same name but different version. This ensures that the applications which access a particular assembly continue to access the same assembly even if another version of that assembly is installed on that machine.

1. **What Is Strong-typing Versus Weak-typing? Which Is Preferred? Why?**

Strong typing implies that the types of variables involved in operations are associated to the variable, checked at compile-time, and require explicit conversion; weak typing implies that they are associated to the value, checked at run-time, and are implicitly converted as required. (Which is preferred is a disputable point, but I personally prefer strong typing because I like my errors to be found as soon as possible.)

1. **What Is The Difference Between An Exe And A Dll?**
   * You can create an objects of Dll but not of the EXE.
   * Dll is an In-Process Component whereas EXE is an OUt-Process Component.
   * Exe is for single use whereas you can use Dll for multiple use.
   * Exe can be started as standalone where dll cannot be.
2. **Describe The Advantages Of Writing A Managed Code Application Instead Of Unmanaged One. What's Involved In Certain Piece Of Code Being Managed?**

"Advantage includes automatic garbage collection,memory management,security,type checking,versioning Managed code is compiled for the .NET run-time environment. It runs in the Common Language Runtime (CLR), which is the heart of the .NET Framework. The CLR provides services such as security,memory management, and cross-language integration. Managed applications written to take advantage of the features of the CLR perform more efficiently and safely, and take better advantage of developers existing expertise in languages that support the .NET Framework.

Unmanaged code includes all code written before the .NET Framework was introduced-this includes code written to use COM, native Win32, and Visual Basic 6. Because it does not run inside the .NET environment, unmanaged code cannot make use of any .NET managed facilities."

1. **What Is Public Or Shared Assemblies ?**

These are static assemblies that must have a unique shared name and can be used by any application.

An application uses a private assembly by referring to the assembly using a static path or through an XML-based application configuration file. While the CLR doesn't enforce versioning policies-checking whether the correct version is used-for private assemblies, it ensures that an application uses the correct shared assemblies with which the application was built. Thus, an application uses a specific shared assembly by referring to the specific shared assembly, and the CLR ensures that the correct version is loaded at runtime.

1. **What Is Metadata?**

Metadata is machine-readable information about a resource, or ""data about data."" Such information might include details on content, format, size, or other characteristics of a data source. In .NET, metadata includes type definitions, version information, external assembly references, and other standardized information.

1. **What Are The Mobile Devices Supported By .net Platform**

The Microsoft .NET Compact Framework is designed to run on mobile devices such as mobile phones, Personal Digital Assistants (PDAs), and embedded devices. The easiest way to develop and test a Smart Device Application is to use an emulator.

These devices are divided into two main divisions:

1) Those that are directly supported by .NET (Pocket PCs, i-Mode phones, and WAP devices)  
2) Those that are not (Palm OS and J2ME-powered devices).

1. **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.

1. **What Is Boxing And Unboxing ?**

Boxing:  
The conversion of a value type instance to an object, which implies that the instance will carry full type information at run time and will be allocated in the heap. The Microsoft intermediate language (MSIL) instruction set's box instruction converts a value type to an object by making a copy of the value type and embedding it in a newly allocated object.

Un-Boxing:  
The conversion of an object instance to a value type.

1. **What Is Msil, Il, Cts And, Clr ?**

MSIL: (Microsoft intermediate language):  
When compiling to managed code, the compiler translates your source code into Microsoft intermediate language (MSIL), which is a CPU-independent set of instructions that can be efficiently converted to native code. MSIL includes instructions for loading, storing, initializing, and calling methods on objects, as well as instructions for arithmetic and logical operations, control flow, direct memory access, exception handling, and other operations. Before code can be executed, MSIL must be converted to CPU-specific code, usually by a just-in-time (JIT) compiler. Because the common language runtime supplies one or more JIT compilers for each computer architecture it supports, the same set of MSIL can be JIT-compiled and executed on any supported architecture.

When a compiler produces MSIL, it also produces metadata. Metadata describes the types in your code, including the definition of each type, the signatures of each type's members, the members that your code references, and other data that the runtime uses at execution time. The MSIL and metadata are contained in a portable executable (PE) file that is based on and extends the published Microsoft PE and Common Object File Format (COFF) used historically for executable content. This file format, which accommodates

MSIL or native code as well as metadata, enables the operating system to recognize common language runtime images. The presence of metadata in the file along with the MSIL enables your code to describe itself, which means that there is no need for type libraries or Interface Definition Language (IDL). The runtime locates and extracts the metadata from the file as needed during execution.

IL: (Intermediate Language):  
A language used as the output of a number of compilers and as the input to a just-in-time (JIT) compiler. The common language runtime includes a JIT compiler for converting MSIL to native code.

CTS: (Common Type System):  
The specification that determines how the common language runtime defines, uses, and manages types

CLR: (Common Language Runtime):  
The engine at the core of managed code execution. The runtime supplies managed code with services such as cross-language integration, code access security, object lifetime management, and debugging and profiling support.

1. **What Is Gc (garbage Collection) And How It Works**

One of the good features of the CLR is Garbage Collection, which runs in the background collecting unused object references, freeing us from having to ensure we always destroy them. In reality the time difference between you releasing the object instance and it being garbage collected is likely to be very small, since the GC is always running.  
[The process of transitively tracing through all pointers to actively used objects in order to locate all objects that can be referenced, and then arranging to reuse any heap memory that was not found during this trace. The common language runtime garbage collector also compacts the memory that is in use to reduce the working space needed for the heap.]

Heap:  
A portion of memory reserved for a program to use for the temporary storage of data structures whose existence or size cannot be determined until the program is running.

1. **How Do I Unload An Application Domain?**

You should use the following static method in the System.AppDomain

class:  
public static void Unload(AppDomain domain)

The Unload method gracefully shuts down the specified application domain. During shutdown no new threads are allowed to enter the application domain and all application domain specific data structures are freed. You cannot unload an application domain in a finalizer or destructor.

If the application domain has run code from a domain-neutral assembly, the domains copy of the statics and related CLR data structures are freed, but the code for the domain-neutral assembly remains until the process is shutdown. There is no mechanism to fully unload a domain-neutral assembly other than shutting down the process.

1. **How Do I Unload An Assembly?**

CLR does not provide a way to unload an assembly. The only mechanism to remove the assembly is to unload the application domain in which the assembly is loaded. If you wish to remove an assembly after it has been used, you should create an application domain, load the assembly into the created application domain, and then unload the application domain when you no longer need the assembly.

1. **How Do I Run Managed Code In A Process?**

The first step in running managed code in a process is to get the CLR loaded and initialized using a CLR host. Typically, a host consists of both managed code and unmanaged code. The managed code which executes in the default domain is usually responsible for creating the application domains in which the user code exists. All CLR hosts must contain unmanaged code because execution must begin in unmanaged code. The .NET Frameworks provides a set of unmanaged APIs that the host can use to configure the CLR, load the CLR into a process, load the hosts managed code into the default domain, and transition from the unmanaged code to the managed code. The second step in running managed code in a process is to create application domains in which the user code will execute. The creator of the application domain can specify criteria which control code isolation, security, and loading of assemblies. The third step in running managed code in a process is to execute user code in one or more application domains created in the previous step. All code that is run in the CLR must be part of an assembly. There are three options for loading assemblies. First, precompiled assemblies can be loaded from disk. Second, precompiled assemblies can be loaded from an array of bytes. Third, assemblies can be built dynamically in an application domain using the BCL Reflection Emit APIs.

Note. For an application launched from the command-line, the shell host executes the steps described above on behalf of the user and hides the complexity from the user.

1. **What Is The Difference Between An Application Domain And A Process?**

An application domain is lighter than a process. Application domains are appropriate for scenarios that require isolation without the heavy cost associated with running an application within a process. A process runs exactly one application. In contrast, the CLR allows multiple applications to be run in a single process by loading them into separate application domains. Additionally, the CLR verifies that user code in an application domain is type safe.

1. **What Is A Clr Host?**

The Windows operating system does not provide support for running a CLR application. That support is provided by a CLR host. A CLR host is an application that is responsible for loading the CLR into a process, creating application domains within the process, and executing user code within the application domains. Examples of hosts that ship with the .NET Framework include:

ASP.NET. An ISAPI filter that ships with ASP.NET loads the CLR and does the initialization necessary to handle web requests.

Internet Explorer. A MIME filter hooks into Internet Explorer versions 5.01 and higher to execute managed controls referenced from HTML pages.

Shell Executables. When a managed application is launched from a shell, a small piece of unmanaged code loads the CLR and transitions control of the application to the CLR.

1. **Can We Maintain State In Webservice?**

Yes. Webservice don't have any mechanism.So, It can maintain state. By default, webservice class is derived for state maintenance.

1. **Difference Between Authentication And Authorization?**

Authentication is a process of identifying a user based on their credentials(means user id and password).

Authorization is process of determining whether an authenticated user is allowed to access a specific resource or not.

1. **What Is The Purpose Of Enumerable Class In .net?**

It contains various extension methods like Where(), OrderBy() etc. for the IList classes.

Inorder to get access for those methods, simply we have to include the System.Linq namespace.

1. **In The Page Load Event I Assigned Dropdownlist's Datasource Property To A Valid List. On The Submit Button Click.. The Same Datasource Property Is Coming As Null. Why?**

As ViewState is not storing the DataSource property.

We have to use Items property of DropDownList to get the items back.

1. **What Is Data Adapter?**

Data adapter are objects that connect one or more command objects to a dataset object. They provide logic that gets the data from the data store and populates the tables in the dataset, or pushes the changes in the dataset back into the data store.

1. **What Is Replication?**

Replication is way of keeping data synchronized in multiple databases. SQL server replication has two important aspects publisher and subscriber.

Publisher: Database server that makes data available for replication is called as publisher.  
Subscriber: Database servers that get data from the publishers is called as subscribers.

1. **What Are The Different Types Of Remote Object Creation Mode In .net?**

There are two types of remote object creation mode. They are,

1.SAO(server activated objects)  
2.CAO(client activated objects)

1. **What Is Application Domain?**

In application domains multiple application can run in the same process with out influencing each othere. If one of the application domains throws error it does not affect the other application domains.

1. **What Is Thread.sleep()?**

Thread's execution can be paused by calling the thread.sleep method. This method takes an integer value that determines how long the thread should sleep.  
Eg: Thread.CurrentThread.Sleep(2000)

1. **What Is Addressof Operator?**

Addressof operator creates a delegate object to a method.

example:  
Dim pthread1 as new thread(AddressOf thread1)  
Here thread1 is a method. pthread1 is a delegate object.

1. **What Is Multi-threading?**

Multi-threading forms subset of multi-tasking instead of having to switch between programs this feature switches between different parts of the same program. For example, we can write in word and at the same time word is doing a spell check in background.

1. **What Is Dcom?**

DCOM is a distributed extension of COM. It differs from COM in that it allows for creating objects distributed across a network, a protocol for invoking that object's methods, and secure to the object. DCOM provides a wrapper around COM, hence it is a backward compatible extension.

1. **How To Install Or Uninstall A Windows Service?**

For Install:-

1)Go to visual Studio command prompt  
(Start => All programs => Microsoft Visual Studio 2005 => Visual Studio Tools => Visual Studio 2005 Command Prompt)

2)installutil <Path of your Service>  
(e.g installutil C:\MyService.exe)

For UN Install:-

3)Go to visual Studio command prompt  
(Start => All programs => Microsoft Visual Studio 2005 => Visual Studio Tools => Visual Studio 2005 Command Prompt)

4)installutil -U <Path of your Service>  
(e.g installutil -U C:\MyService.exe)

1. **What Is Jit? What Are The Different Types Of Jit?**

JIT stands for Just-in-time compiler.It is a part of the runtime execution environment. It compiles the intermediate language and execute them. There are three types of JIT. They are,  
1. Pre-JIT : compiles complete source code into native code in a single compilation cycle. This is done by at the time of deployment of the application.  
2. Econo-JIT : compiles only those methods that they are called at runtime. However, These compiled methods are removed when they are not required.  
3. Normal-JIT : compiles only those methods that they are called at runtime. These methods are compiled the first time they are called and then they are stored in cache. When the same methods are called again, the compiled code from cache is used for execution.

1. **What Is Delay Signing?**

During development process we have to sign strong name with our application. But it is not good practice in the security point of view. For that, we can sign it later. It is called delay signing.

1. **What Is Strong Name?**

Strong name helps GAC(Global assembly cache) to differentiate between two versions.It is only needed when we deploy the assembly with GAC. It uses public key cryptocrophy to ensure that no one spoof it.

1. **What Is Code Verification?**

Code verification ensure that proper code execution and type safety while the code runs. It does not allow to perform illegal operation such as invalid memory location.

1. **What Is Code Access Security?**

Code access security grant rights to the program depends on security configuration of the machine. For example, Program can have access to create or edit a file but security configuration of the machine does not allow to delete a file.

1. **What Is Il?**

IL stands for Intermediate language. All .Net code is converted into IL. Then IL is converted into native code at run time by JIT(just in time) compiler and executed by them.

1. **What Is Class Library In .net**

class library is a collection of prewritten, ready-made software routines that act as templates for programmers to use in writing object-oriented application programs

1. **How To Block A Class From Being Inherited Further?**

Use sealed keyword.

Eg: public sealed class TheClass  
{  
}

1. **How To Find The Current Application File Path While Runtime?**

System.Reflection.Assembly.GetExecutingAssembly().Location

1. **What Is The Base Class Of All Classes In C#?**

It is object (Sytem.Object)

1. **You Are Designing A User Control. You Created New Property Called Backgroundimage Which Is Of Type Image. You Wanted To Disable Storing This Property In The User's Form. How To Achieve This?**

Use the attribute:  
[DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)]

1. **You Are Creating A Custom Usercontrol, Some Of The Newly Created Properties Are Shown In The Properties Window. How You Can Hide A New Property Named Theme From The Properties Window?**

Use the attribute [Browsable(false)] along with the property

[Browsable(false)]  
public string Theme  
{  
get;  
set;  
}

1. **How To Sort An Int Array In C#?**

Array class is having method to sort any arrays.

Example:  
int[] array = new int[] { 10, 9, 8 };  
Array.Sort(array);

Additional:  
Sort is overloaded method with IComparer as argument too.

1. **How To Prevent The Error While Updating Ui Control From Another Thread?**

Control.CheckForIllegalCrossThreadCalls = false;

1. **You Are Designing A Single Person Shooter Game Application. The Player Can Choose Between Multiple Guns. 1) Pistol With 5 Bullets 2) Shotgun With 100 Bullets 3) Grenade Launcher With 20 Grenades Each Gun Will Have Different Sound Effects And Graphics.pressing Ctrl+space Should Rotate Between Guns.which Design Pattern Should You Employ For This?**

Use Strategy Pattern.  
Explanation: Each gun having different performance and graphics.  
So this can be accomodated in the algorithm.  
Strategy pattern is best suited for shifting the guns/algorithms There will be a IGun interface implemented by 3 classes  
Pistol :IGun  
Shotgun :IGun  
GrenadeLauncher :IGun

1. **How To Find Whether The Application Is Run From Inside Ide Or Not?**

Use System.Diagnostics.Debugger.IsAttached property.If it is true that means a debugger is attached to the application else not

Useful: While creating applications we can check this property and skip the login page/form and directly launch the page/form we are debugging.

1. **What Is The Difference Between Var And Dynamic Types In C# 4.0?**

var gets processed in the compile time itself and shows any error during compile time itself.

dynamic gets processed in the runtime only and in case of errors it is hidden until runtime.

dynamic was introduced as part of the DLR (Dynamic Language Runtime)in .Net.

Example:  
dynamic d = new MyClass();  
d.InexistingMethod();

Eventhough the method "InexistingMethod" is not there - it will get compiled and thrown exception in runtime.

1. **How Can You Access A Private Method Of A Class?**

Use .Net Reflection. Using reflection we can invoke private methods too.

Example:

Car car = new Car();  
Type type = car.GetType();  
MethodInfo methodInfo = type.GetMethod("ShiftGearUp", BindingFlags.NonPublic | BindingFlags.Instance);  
methodInfo.Invoke(car, null);

 Pre: Create a class Car with private method "ShiftGearUp"

1. **Why Linq Is Having Select Clause At The End?**

Due to IntelliSense of Visual Studio.

If the select clause was in the first of LINQ, then the IDE cannot populate the properties inside the object.

Eg:from f in db.Forum where f.Id > 0 select f.Name;

If the select was in front - the "select f." could not populate any autocomplete properties or methods.

1. **Name The Method Of Servicebase Class?**

The following are the methods of ServiceBase class.  
• OnStart  
• OnStop  
• OnPause  
• OnContinue  
• OnShutdown  
• OnPowerEvent  
• OnCustomCommand

1. **Name The Two Classes Are Required For Implementing A Windows Service?**

The following two classes are required for implementing a Windows Service:  
• System.ServiceProcess.ServiceBase  
• System.Configuration.Install.Installer

1. **What Are The Benefits Of Using Windows Services:**

The following are the benefits of using Windows Services:  
• Network connection management  
• Disk access monitoring  
• Security control of application  
• Log messages related to the application's requirements

1. **What Are Windows Services?**

Windows services are long running executable applications that typically do not possess any user interface, are controlled by the Service Control Manager (SCM) and can even be configured to start automatically after the system boots. They typically execute in their own windows sessions. They can execute even if the user has not logged in to the system. They can even be started, paused, re-started manually. These applications are somewhat similar to the daemon processes of UNIX in the sense that they remain dormant most of the time and execute in the background as and when needed.

1. **By Default Security Setting In .net?**

anonymous access

1. **Can Namespace Contain The Private Class?**

no, having a single private class does not mean anything , it is applicable only in case of nested class.

1. **Wcf And What Is Difference Between Wcf And Web Services?**

WCF is windows communication foundation same as webservices. actually WCF =webservices + remoting

1. **Is Exe Is Machine Dependent?**

yes

1. **What Is Difference Between Il And Dll ?**

DLL contains the IL and other details (metadata) inside.

1. **How Will You Deploy The Dll File In Gac?**

first create the strong name for dll and add this sn name key into the application then use gacutil command to register the dll in GAC.

1. **If Dll And Exe Files Are Same It Means You Can Deploy Both The Files In Gac?**

No, deploying a exe file does not mean anything because classes and method are exposed in dll only and exe file is pure executable, any application can not call the function of exe file.

1. **What Is The Task Perform By Clr?**

It have the task like  
1.running the garbage collector  
2.Code verification  
3.code access security  
4.executing IL

1. **Suppose Two Interfaces Have Same Method, So How Will You Implement These Methods In Derive Class?**

by using interface name

1. **What Is Using Keyword?**

using keyword can be used in two format  
1.include the namespace in file  
2.handle the idisposable objects inside using block.

iFace obj = new myclass()

Myclass obj1 = new myclass()

Obj.hey();  
Obj1.hey();

1. **How To Get The Number After Decimal Point In .net?if The No Is 2.36. Result Should Be 36?**

string result=number.Substring(number.inderOf('.')+1);

It will first get the index of '.' and then add 1 to get the index of next charater then with the help of substring method we will get the no after the decimal point to the last

1. **Wht Executescaler Method Is Used?**

Executes the query, and returns the first column of the first row in the result set returned by the query. Additional columns or rows are ignored.

1. **What Is Literal Control**

A Literal Web Server control doesn't have any visual appearance on a Web Form but is used to insert literal text into a Web Form. This control makes it possible to add HTML code directly in the code designer window without switching to design view and clicking the HTML button to edit the HTML.

1. **How Many Languages .net Is Supporting Now?**

When .NET was introduced it came with several languages.  
VB.NET,  
C#,  
COBOL  
and  
Perl, etc.

1. **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.

1. **What Is Smart Navigation In .net?**

The cursor position is maintained when the page gets refreshed due to the server side validation and the page gets refreshed.

1. **What Is View State In .net?**

The web is stateless. But in ASP.NET, the state of a page is maintained in the in the page itself automatically.  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.

1. **How Do You Validate The Controls In An Asp .net Page?**

Using special validation controls that are meant for validation of any controle.We have Range Validator, Email Validator in .NET to validate any control.

1. **How To Manage Pagination In A Page Using .net?**

Using pagination option in DataGrid control is available in .NET. We have to set the number of records for a page, then it takes care of pagination by itself automatically.

1. **Observations Between Vb.net And Vc#.net?**

Choosing a programming language depends on your language experience and the scope of the application you are building. While small applications are often created using only one language, it is not uncommon to develop large applications using multiple languages.For example, if you are extending an application with existing XML Web services, you might use a scripting language with little or no programming effort. For client-server applications, you would probably choose the single language you are most comfortable with for the entire application. For new enterprise applications, where large teams of developers create components and services for deployment across multiple remote sites, the best choice might be to use several languages depending on developer skills and long-term maintenance expectations.The .NET Platform programming languages - including Visual Basic .NET, Visual C#, and Visual C++ with managed extensions, and many other programming languages from various vendors - use .NET Framework services and features through a common set of unified classes. The .NET unified classes provide a consistent method of accessing the platform's functionality. If you learn to use the class library, you will find that all tasks follow the same uniform architecture. You no longer need to learn and master different API architectures to write your applications.

1. **Advantages Of Migrating To Vb.net?**

Visual Basic .NET has many new and improved language features — such as inheritance, interfaces, and overloading that make it a powerful object-oriented programming language. As a Visual Basic developer, you can now create multithreaded, scalable applications using explicit multithreading. Other new language features in Visual Basic .NET include structured exception handling, custom attributes, and common language specification (CLS) compliance. The CLS is a set of rules that standardizes such things as data types and how objects are exposed and interoperate. Visual Basic .NET adds several features that take advantage of the CLS. Any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET. And you, as a Visual Basic user, can access classes, components, and objects from other CLS-compliant programming languages without worrying about language-specific differences such as data types.CLS features used by Visual Basic .NET programs include assemblies, namespaces, and attributes.

1. **Advantages Of Vb.net?**

1.First of all, VB.NET provides managed code execution that runs under the Common Language Runtime (CLR), resulting in robust, stable and secure applications. All features of the .NET framework are readily available in VB.NET.

2.VB.NET is totally object oriented. This is a major addition that VB6 and other earlier releases didn't have.

3.The .NET framework comes with ADO.NET, which follows the disconnected paradigm, i.e. once the required records are fetched the connection no longer exists. It also retrieves the records that are expected to be accessed in the immediate future. This enhances Scalability of the application to a great extent.

4.VB.NET uses XML to transfer data between the various layers in the DNA Architecture i.e. data are passed as simple text strings.

5.Error handling has changed in VB.NET. A new Try-Catch-Finally block has been introduced to handle errors and exceptions as a unit, allowing appropriate action to be taken at the place the error occurred thus discouraging the use of ON ERROR GOTO statement. This again credits to the maintainability of the code.

1. **Using Activex Control In .net?**

ActiveX control is a special type of COM component that supports a User Interface. Using ActiveX Control in your .Net Project is even easier than using COM component. They are bundled usually in .ocx files. Again a proxy assembly is made by .Net utility AxImp.exe (which we will see shortly) which your application (or client) uses as if it is a .Net control or assembly. Making Proxy Assembly For ActiveX Control: First, a proxy assembly is made using AxImp.exe (acronym for ActiveX Import) by writing following command on Command Prompt: C:> AxImp C:MyProjectsMyControl.ocx This command will make two dlls, e.g., in case of above command MyControl.dll AxMyControl.dll The first file MyControl.dll is a .Net assembly proxy, which allows you to reference the ActiveX as if it were non-graphical object. The second file AxMyControl.dll is the Windows Control, which allows u to use the graphical aspects of activex control and use it in the Windows Form Project.

1. **What Is Machine.config In .net?**

Machine configuration file: The machine.config file contains settings that apply to the entire computer. This file is located in the %runtime install path%Config directory. There is only one machine.config file on a computer. The Machine.Config file found in the "CONFIG" subfolder of your .NET Framework install directory (c:WINNT Microsoft.NETFramework{Version Number}CONFIG on Windows 2000 installations). The machine. config,which can be found in the directory $WINDIR $Microsoft.NETFrameworkv 1.0.3705CONFIG,is an XML-formatted configuration file that specifies configuration options for the machine. This file contains, among many other XML elements, a browserCaps element. Inside this element are a number of other elements that specify parse rules for the various User-Agents, and what properties each of these parsings supports.

For example, to determine what platform is used, a filter element is used that specifies how to set the platform property based on what platform name is found in the User-Agent string. Specifically, the machine.config file contains:

platform=Win95  
platform=Win98  
platform=WinNT

1. **What Is Web.config In .net?**

In classic ASP all Web site related information was stored in the metadata of IIS. This had the disadvantage that remote Web developers couldn't easily make Web-site configuration changes. For example, if you want to add a custom 404 error page, a setting needs to be made through the IIS admin tool, and you're Web host will likely charge you a flat fee to do this for you. With ASP.NET, however, these settings are moved into an XML-formatted text file (Web.config) that resides in the Web site's root directory. Through Web.config you can specify settings like custom 404 error pages, authentication and authorization settings for the Web sitempilation options for the ASP.NET Web pages, if tracing should be enabled, etc. The Web.config file is an XML-formatted file. At the root level is the tag. Inside this tag you can add a number of other tags, the most common and useful one being the system.web tag, where you will specify most of the Web site configuration parameters. However, to specify application-wide settings you use the tag.

For example, if we wanted to add a database connection string parameter we could have a Web.config file like so.

1. **What Is The Difference Between Ado And Ado.net?**

ADO uses Recordsets and cursors to access and modify data. Because of its inherent design, Recordset can impact performance on the server side by tying up valuable resources. In addition, COM marshalling - an expensive data conversion process - is needed to transmit a Recordset. ADO.NET addresses three important needs that ADO doesn't address:

1. Providing a comprehensive disconnected data-access model, which is crucial to the Web environment  
2. Providing tight integration with XML, and  
3. Providing seamless integration with the .NET Framework (e.g., compatibility with the base class library's type system). From an ADO.NET implementation perspective, the Recordset object in ADO is eliminated in the .NET architecture. In its place, ADO.NET has several dedicated objects led by the DataSet object and including the DataAdapter, and DataReader objects to perform specific tasks. In addition, ADO.NET DataSets operate in disconnected state whereas the ADO RecordSet objects operated in a fully connected state.

In ADO, the in-memory representation of data is the recordset. In ADO.NET, it is the dataset. 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.

1. **What Is The Difference Between Vb And Vb.net?**

Now VB.NET is object-oriented language. The following are some of the differences:

Data Type Changes

The .NET platform provides Common Type System to all the supported languages. This means that all the languages must support the same data types as enforced by common language runtime. This eliminates data type incompatibilities between various languages. For example on the 32-bit Windows platform, the integer data type takes 4 bytes in languages like C++ whereas in VB it takes 2 bytes. Following are the main changes related to data types in VB.NET:

* + Under .NET the integer data type in VB.NET is also 4 bytes in size.
  + VB.NET has no currency data type. Instead it provides decimal as a replacement.
  + VB.NET introduces a new data type called Char. The char data type takes 2 bytes and can store Unicode characters.
  + VB.NET do not have Variant data type. To achieve a result similar to variant type you can use Object data type. (Since every thing in .NET including primitive data types is an object, a variable of object type can point to any data type).
  + In VB.NET there is no concept of fixed length strings.
  + In VB6 we used the Type keyword to declare our user-defined structures. VB.NET introduces the structure keyword for the same purpose.
  + Declaring Variables

Consider this simple example in VB6:  
Dim x,y as integer

1. **What Is A Strong Name In .net?**

A strong name consists of the assembly's identity its simple text name, version number, and culture information (if provided) plus a public key and a digital signature. It is generated from an assembly file (the file that contains the assembly manifest, which in turn contains the names and hashes of all the files that make up the assembly), using the corresponding private key. Assemblies with the same strong name are expected to be identical.

Strong names guarantee name uniqueness by relying on unique key pairs. No one can generate the same assembly name that you can, because an assembly generated with one private key has a different name than an assembly generated with another private key.

When you reference a strong-named assembly, you expect to get certain benefits, such as versioning and naming protection. If the strong-named assembly then references an assembly with a simple name, which does not have these benefits, you lose the benefits you would derive from using a strong-named assembly and revert to DLL conflicts. Therefore, strong-named assemblies can only reference other strong-named assemblies.

There are two ways to sign an assembly with a strong name:

1.Using the Assembly Linker (Al.exe) provided by the .NET Framework SDK.  
2.Using assembly attributes to insert the strong name information in your code. You can use either the AssemblyKeyFileAttribute or the AssemblyKeyNameAttribute, depending

1. **What Is A Manifest In .net?**

An assembly manifest contains all the metadata needed to specify the assembly's version requirements and security identity, and all metadata needed to define the scope of the assembly and resolve references to resources and classes. The assembly manifest can be stored in either a PE (Portable Executable) file (an .exe or .dll) with Microsoft intermediate language (MSIL) code or in a standalone PE (Portable Executable) file that contains only assembly manifest information.

The following table shows the information contained in the assembly manifest. The first four items the assembly name, version number, culture, and strong name information make up the assembly's identity.

Assembly name: A text string specifying the assembly's name.

Version number: A major and minor version number, and a revision and build number. The common language runtime uses these numbers to enforce version policy.

Culture: Information on the culture or language the assembly supports. This information should be used only to designate an assembly as a satellite assembly containing culture- or language-specific information. (An assembly with culture information is automatically assumed to be a satellite assembly.) Strong name information: The public key from the publisher if the assembly has been given a strong name.

1. **Creating A Key Pair In .net?**

You can create a key pair using the Strong Name tool (Sn.exe). Key pair files usually have an .snk extension. To create a key pair At the command prompt, type the following command:

In this command, file name is the name of the output file containing the key pair. The following example creates a key pair called sgKey.snk

1. **What Is The Difference Between "using System.data;" And Directly Adding The Reference From "add References Dialog Box"?**

When u compile a program using command line, u add the references using /r switch. When you compile a program using Visual Studio, it adds those references to our assembly, which are added using "Add Reference" dialog box. While "using" statement facilitates us to use classes without using their fully qualified names.

For example: if u have added a reference to "System. Data. SqlClient" using "Add Reference" dialog box then u can use SqlConnection class like this:

System.Data.SqlClient.SqlConnection

But if u add a "using System. Data. SqlClient" statement at the start of ur code then u can directly use Sql Connection class.On the other hand if u add a reference using "using System.Data.SqlClient" statement, but don't add it using "Add Reference" dialog box, Visual Studio will give error message while we compile the program

1. **What Is Gac In .net?**

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. Assemblies deployed in the global assembly cache must have a strong name. When an assembly is added to the global assembly cache, integrity checks are performed on all files that make up the assembly. The cache performs these integrity checks to ensure that an assembly has not been tampered with, for example, when a file has changed but the manifest does not reflect the change. Use a developer tool called the Global Assembly Cache tool (Gacutil.exe), provided by the .NET Framework SDK or Use Windows Explorer to drag assemblies into the cache. To install a strong-named assembly into the global assembly cache At the command prompt, type the following command:

gacutil I

In this command, assembly name is the name of the assembly to install in the global assembly cache.

1. **What Is A Metadata In .net?**

Metadata is information about a PE. In COM, metadata is communicated through non-standardized type libraries.

In .NET, this data is contained in the header portion of a COFF-compliant PE and follows certain guidelines; it contains information such as the assembly’s name, version, language (spoken, not computera.k.a., culture), what external types are referenced, what internal types are exposed, methods, properties, classes, and much more.

The CLR uses metadata for a number of specific purposes. Security is managed through a public key in the PE’s header.

Information about classes, modules, and so forth allows the CLR to know in advance what structures are necessary. The class loader component of the CLR uses metadata to locate specific classes within assemblies, either locally or across networks.

Just-in-time (JIT) compilers use the metadata to turn IL into executable code.

Other programs take advantage of metadata as well.

A common example is placing a Microsoft Word document on a Windows 2000 desktop. If the document file has completed comments, author, title, or other Properties metadata, the text is displayed as a tool tip when a user hovers the mouse over the document on the desktop. You can use the Ildasm.exe utility to view the metadata in a PE. Literally, this tool is an IL disassembler.

1. **What Is Managed Code And Managed Data In .net?**

Managed code is code that is written to target the services of the Common Language Runtime.  
In order to target these services, the code must provide a minimum level of information (metadata) to the runtime.All C#, Visual Basic .NET, and JScript .NET code is managed by default. Visual Studio .NET C++ code is not managed by default, but the compiler can produce managed code by specifying a command-line switch (/CLR).Closely related to managed code is managed data--data that is allocated and de-allocated by the Common Language Runtime's garbage collector. C#, Visual Basic, and JScript .NET data is managed by default.C# data can, however, be marked as unmanaged through the use of special keywords.Visual Studio .NET C++ data is unmanaged by default (even when using the /CLR switch), but when using Managed Extensions for C++, a class can be marked as managed using the \_\_gc keyword. As the name suggests, this means that the memory for instances of the class is managed by the garbage collector.

In addition, the class becomes a full participating member of the .NET Framework community, with the benefits and restrictions that it brings. An example of a benefit is proper interoperability with classes written in other languages (for example, a managed C++ class can inherit from a Visual Basic class).An example of a restriction is that a managed class can only inherit from one base class.

1. **What Is .net And .net Framework?**

It is a Framework in which Windows applications may be developed and run. The Microsoft .NET Framework is a platform for building, deploying, and running Web Services and applications. It provides a highly productive, standards-based, multi-language environment for integrating existing investments with next-generation applications and services as well as the agility to solve the challenges of deployment and operation of Internet-scale applications. The .NET Framework consists of three main parts: the common language runtime, a hierarchical set of unified class libraries, and a componentized version of Active Server Pages called ASP.NET. The .NET Framework provides a new programming model and rich set of classes designed to simplify application development for Windows, the Web, and mobile devices. It provides full support for XML Web services, contains robust security features, and delivers new levels of programming power. The .NET Framework is used by all Microsoft languages including Visual C#, Visual J#, and Visual C++.

1. **What's A Windows Process?**

It’s an application that’s running and had been allocated memory

1. **What's Typical About A Windows Process In Regards To Memory Allocation?**

Each process is allocated its own block of available RAM space, no process can access another process code or data. If the process crashes, it dies alone without taking the entire OS or a bunch of other applications down.

1. **Why Do You Call It A Process? What's Different Between Process And Application In .net, Not Common Computer Usage, Terminology?**

A process is an instance of a running application. An application is an executable on the hard drive or network. There can be numerous processes launched of the same application (5 copies of Word running), but 1 process can run just 1 application.

1. **What Distributed Process Frameworks Outside .net Do You Know?**

Distributed Computing Environment/Remote Procedure Calls (DEC/RPC), Microsoft Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA), and Java Remote Method Invocation (RMI).

1. **What Are Possible Implementations Of Distributed Applications In .net?**

.NET Remoting and ASP.NET Web Services. If we talk about the Framework Class Library, noteworthy classes are in System.Runtime.Remoting and System.Web.Services.

1. **When Would You Use .net Remoting And When Web Services?**

Use remoting for more efficient exchange of information when you control both ends of the application. Use Web services for open-protocol-based information exchange when you are just a client or a server with the other end belonging to someone else.

1. **What's A Proxy Of The Server Object In .net Remoting?**

It’s a fake copy of the server object that resides on the client side and behaves as if it was the server. It handles the communication between real server object and the client object. This process is also known as marshaling.

1. **What Are Remotable Objects In .net Remoting?**

Remotable objects are the objects that can be marshaled across the application domains. You can marshal by value, where a deep copy of the object is created and then passed to the receiver. You can also marshal by reference, where just a reference to an existing object is passed.

1. **What Are Channels In .net Remoting?**

Channels represent the objects that transfer the other serialized objects from one application domain to another and from one computer to another, as well as one process to another on the same box. A channel must exist before an object can be transferred.

1. **What Security Measures Exist For .net Remoting In System.runtime.remoting?**

None. Security should be taken care of at the application level. Cryptography and other security techniques can be applied at application or server level.

1. **What Is A Formatter?**

A formatter is an object that is responsible for encoding and serializing data into messages on one end, and deserializing and decoding messages into data on the other end.

1. **Choosing Between Http And Tcp For Protocols And Binary And Soap For Formatters, What Are The Trade-offs?**

Binary over TCP is the most effiecient, SOAP over HTTP is the most interoperable.

1. **What's Singlecall Activation Mode Used For?**

If the server object is instantiated for responding to just one single request, the request should be made in SingleCall mode.

1. **What's Singleton Activation Mode?**

A single object is instantiated regardless of the number of clients accessing it. Lifetime of this object is determined by lifetime lease.

1. **How Do You Define The Lease Of The Object?**

By implementing ILease interface when writing the class code.

1. **Can You Configure A .net Remoting Object Via Xml File?**

Yes, via machine.config and application level .config file (or web.config in ASP.NET). Application-level XML settings take precedence over machine.config.

1. **How Can You Automatically Generate Interface For The Remotable Object In .net With Microsoft Tools?**

Use the Soapsuds tool.

1. **What Are The Consideration In Deciding To Use .net Remoting Or Asp.net Web Services?**

Remoting is a more efficient communication exchange when you can control both ends of the application involved in the communication process. Web Services provide an open-protocol-based exchange of informaion. Web Services are best when you need to communicate with an external organization or another (non-.NET) technology.

1. **What's Singlecall Activation Mode Used For?**

If the server object is instantiated for responding to just one single request, the request should be made in SingleCall mode.

1. **What Do You Mean By A Windows Process In Regards To Memory Allocation?**

Each process is allocated its own address space typically 4GB in which it can run. No other process can interfere in that address space.

If the process crashes, it dies alone without taking the entire OS or a bunch of other applications down.

1. **What Is The Relationship Between A Process, Application Domain, And Application?**

A process is an instance of a running application. An application is an executable in the computer. There can be numerous processes launched of the same application (5 copies of Word running), but 1 process can run just 1 application. Hence, Process is a running instance of the application.

1. **How To Decide Which To Use .net Remoting Or Asp.net Web Services?**

Remoting is a more efficient communication exchange when you can control both ends of the application involved in the communication process. Remoting is excellent in case of intranet applicaton becuase of the speed.

Web Services provide an open-protocol-based exchange of informaion. Web Services are best when you need to ommunicate with an external organization or another (non-.NET) technology applications.

1. **What Is The Proxy Of The Server Object In .net Remoting?**

Proxy is copy of the server object( a thin layer) that resides on the client side and behaves as if it was the server. It delegate calls to the real server object. This process is also known as marshaling.

1. **What Do Mean By Remotable Objects In .net Remoting?**

Remotable objects can be marshaled across the application domains. You can marshal by value, where a deep copy of the object is created and then passed to the receiver. You can also marshal by reference, where just a reference to an existing object is passed.

1. **Which Class Does The Remote Object Has To Inherit?**

All remote objects should inherit from System.MarshalbyRefObject.

1. **What Are Two Different Types Of Remote Object Creation Mode In .net?**

There are two different ways in which object can be created using Remoting:

-SAO (Server Activated Objects) also called as Well-Known call mode.  
-CAO (Client Activated Objects).  
SAO has two modes "Single Call" and "Singleton". With Single Call object the object is created with every method call thus making the object stateless. With Singleton the object is created only once and the object is shared with all clients.

CAO are stateful as compared to SAO. In CAO the creation request is sent from client side. Client holds a proxy to the server object created on server.

1. **What Are The Situations You Will Use Singleton Architecture In Remoting ?**

If all remoting clients have to share the same data singleton architecture will be used.

1. **What Is Fundamental Of Published Or Precreated Objects In Remoting?**

In scenarios of singleton or single call the objects are created dynamically. But in situations where you want to precreate object and publish it you will use published object scenarios.

Dim obj as new objRemote  
obj.Initvalue = 100  
RemotingServices.Marshal(obj,"RemoteObject")

As shown in above sample following changes will be needed on server side.

Remoting Configuration.Register WellKnown ServiceType is replaced by Remoting Services .Marshal (obj,"Remote Object")  where "obj" is the precreated objected on the server whose value is initialized to 100.

1. **What Are The Ways In Which Client Can Create Object On Server In Cao Model?**

There are two ways by which you can create Client objects on remoting server:

* + Activator.CreateInstance()
  + By Keyword "New".

1. **Are Cao Stateful In Nature?**

Yes. In CAO remoting model client creates a instance on server and instance variable set by client on server can be retrieved again with correct value.

1. **Is It A Good Design Practice To Distribute The Implementation To Remoting Client?**

It’s never advisable to distribute complete implementation at client, due to following reasons:  
-Any one can use ILDASM or similar utility and get access to your code.

-It’s a bad architecture move to have full implementation as client side as any changes in implementation on server side you have to redistribute it again.

So the best way is to have a interface or SOAPSUDS generated meta-data DLL at client side rather than having full implementation.

1. **Which Config File Has All The Supported Channels/protocol?**

Machine.config file has all the supported channels and formatter supported by .NET remoting.Machine.config file can be found at "C:\WINDOWS\Microsoft.NET\Framework\vXXXXX\CONFIG" path. Find element in the Machine. config file which has the channels and the formatters. Below is a figure shown which can give a clear idea of how the file looks like.

1. **Can Non-default Constructors Be Used With Single Call Sao?**

Non-Default constructors can not be used with single call objects as object is created with every method call, there is no way to define Non-default constructors in method calls. It’s possible to use Non-Default constructor with Client activated objects as both methods "NEW" keyword and "Activator.CreateInstance" provide a way to specify Non-Default constructors.

1. **How Can We Call Methods In Remoting Asynchronously?**

We can make Asynchronous method calls by using delegates.

1. **What Is Asynchronous One-way Calls?**

One-way calls are a different from asynchronous calls from execution angle that the .NET Framework does not guarantee their execution. In addition, the methods used in this kind of call cannot have return values or out parameters. One-way calls are defined by using [OneWay()] attribute in class.

1. **What Is Marshalling And What Are Different Kinds Of Marshalling?**

Marshaling is used when an object is converted so that it can be sent across the network or across application domains. Unmarshaling creates an object from the marshaled data. There are two ways to do marshalling:

-Marshal-by-value (MBV): In this the object is serialized into the channel, and a copy of the object is created on the other side of the network. The object to marshal is stored into a stream, and the stream is used to build a copy of the object on the other side with the unmarshalling sequence.

-Marshaling-by-reference (MBR): Here it creates a proxy on the client that is used to communicate with the remote object. The marshaling sequence of a remote object creates an ObjRef instance that itself can be serialized across the network.

1. **What Is Objref Object In Remoting?**

All Marshal() methods return ObjRef object.The ObjRef is serializable because it implements the interface ISerializable, and can be marshaled by value. The ObjRef knows about location of the remote object, host name, port number and object name.

1. **What Is A Web Service?**

Web Services are business logic components which provide functionality via the Internet using standard protocols such as HTTP. Web Services uses Simple Object Access Protocol (SOAP) in order to expose the business functionality.SOAP defines a standardized format in XML which can be exchanged between two entities over standard protocols such as HTTP. SOAP is platform independent so the consumer of a Web Service is therefore completely shielded from any implementation details about the platform exposing the Web Service. For the consumer it is simply a black box of send and receive XML over HTTP. So any web service hosted on Windows can also be consumed by UNIX and LINUX platform.

1. **What Is Uddi?**

Full form of UDDI is Universal Description, Discovery and Integration. It is a directory that can be used to publish and discover public Web Services.

1. **What Is Disco?**

DISCO is the abbreviated form of Discovery. It is basically used to club or group common services together on a server and provides links to the schema documents of the services it describes may require.

1. **What Is Wsdl?**

Web Service Description Language (WSDL)is a W3C specification which defines XML grammar for describing Web Services.XML grammar describes details such as where we can find the Web Service (its URI), what are the methods and properties that service supports, data type support and supported protocols.

Clients can consume this WSDL and build proxy objects that clients use to communicate with the Web Services.

1. **What The Different Phase/steps Of Acquiring A Proxy Object In Webservice?**

Following are the different steps needed to get a proxy object of a webservice at the client side:  
-Client communicates to UDDI node for webservice either through browser or UDDI’s public web service.  
-UDDI node responds with a list of webservices.  
-Every service listed by webservice has a URI pointing to DISCO or WSDL document.  
-After parsing the DISCO document, we follow the URI for the WSDL document related to the webservice which we need.  
-Client then parses the WSDL document and builds a proxy object which can communicate with Webservice.

1. **What Is File Extension Of Webservices?**

.ASMX is extension for Webservices.

1. **Which Attribute Is Used In Order That The Method Can Be Used As Webservice?**

WebMethod attribute has to be specified in order that the method and property can be treated as WebService.

1. **What Are The Steps To Create A Webservice And Consume It?**

In Visual Studio you can create a new project using "Asp.NET Web Service" template or just create asmx-file using any text editor. After webservice is implemented, you need either add a webreference to it in Visual Studio, or to create a proxy for the client manually using wsdl.exe utility and build a client program using created proxy class, so that using it client can consume the webservice.

1. **Why Do You Call It A Process? What's Different Between Process And Application In .net, Not Common Computer Usage, Terminology?**

A process is an instance of a running application. An application is an executable on the hard drive or network. There can be numerous processes launched of the same application (5 copies of Word running), but 1 process can run just 1 application.

1. **What's Singlecall Activation Mode Used For?**

If the server object is instantiated for responding to just one single request, the request should be made in SingleCall mode.

1. **What Is The Relation Between Classes And Objects?**

Class is a group of items, attributes of some entity. Object is any specific item that may or may not be a part of the class.When a class is created, objects for those classes are created.

1. **Explain Different Properties Of Object Oriented Systems.**

An object oriented system revolves around a Class and objects. A class is used to describe characteristics of any entity of the real world. An object is a pattern of the class. An actual object created at runtime is called as an instance. A class, apart from characteristics has some functions to perform called as methods. For.e.g A class named “Food” has attributes like ‘price’, ‘quantity’. “Food” class has methods like Serve\_food(), bill\_food().

Objects in Object Oriented Systems interact through messages.

* + Inheritance:- The main class or the root class is called as a Base Class. Any class which is expected to have ALL properties of the base class along with its own is called as a Derived class. The process of deriving such a class is Derived class. For the “Food” class, a Derived class can be “Class Chinesefood”.
  + Abstraction:- Abstraction is creating models or classes of some broad concept. Abstraction can be achieved through Inheritance or even Composition.
  + Encapsulation:- Encapsulation is a collection of functions of a class and object. The “Food” class is an encapsulated form. It is achieved by specifying which class can use which members (private, public, protected) of an object.
  + Polymorphism:- Polymorphism means existing in different forms. Inheritance is an example of Polymorphism. A base class exists in different forms as derived classes. Operator overloading is an example of Polymorphism in which an operator can be applied in different situations.

1. **What Is Difference Between Association, Aggregation And Inheritance Relationships?**

Association:  
An association describes a group of links with common structure and common semantics.

Example:  
A person works for a company. It is often appear verb in the problem statement. It is implemented as a pointer or reference to an object  
  
Aggregation:  
An aggregation is the "a part of" relationship in which objects represents the components in same assembly. Aggregation may be the special form of Association.

Example :  
A Paragraph consists of many sentences. Here Paragraph contains sentences.  
  
Inheritance:  
Inheritance is a relationship between a class and one or more redefined version of it. It is sometimes called "is a relationship.

1. **Explain The Features Of An Abstract Class In Net.**

An Abstract class is only used for inheritance. This means it acts as a base class for its derived classes. One cannot use it to create objects. When a class is declared as “Abstract”, it can only be inherited and not instantiated. MustInherit Keyword can be used to declare a class as abstract. Abstract classes can also specify abstract members.

1. **Difference Between Abstract Classes And Interfaces**

Abstract classes:

* + An abstract class can implement methods.
  + An abstract class can inherit from a class and one or more interfaces.
  + An abstract class can contain fields.
  + An abstract class can implement a property.
  + An abstract class can contain constructors or destructors.
  + An abstract class cannot be inherited from by structures.
  + An abstract class cannot support multiple inheritance.

interfaces:

* + An Interface cannot implement methods.
  + An Interface can only inherit from another Interface.
  + An Interface cannot contain fields.
  + An Interface can contain property definitions.
  + An Interface cannot contain constructors or destructors.
  + An Interface can be inherited from by structures.
  + An Interface can support multiple inheritance.

1. **Similarities And Difference Between Class And Structure In .net**

Similarities:

* + Both Class and Structures can have methods, variables and objects.
  + Both can have constructor.
  + Both are user defined types.

Differences:

* + Structure being value type, the variables cannot be assigned to NULL. On the other hand, classes being reference type, a class variable can be assigned to NULL.
  + Structure is allocated on a stack when instantiated, while Class is allocated on a heap.
  + When a method is passed to a class, pass by reference is used. Pass by value is used for struts.
  + A class can have a destructor.

1. **Features Of Static/shared Classes.**

Static classes are used when any change made to an object should not affect the data and functions of the class. They can be used to create data and functions without the need of an instance.  
  
Following are features of Static/Shared classes:-

* + They can not be instantiated. By default a object is created on the first method call to that object.
  + Static/Shared classes can not be inherited.
  + Static/Shared classes can have only static members.
  + Static/Shared classes can have only static constructor.

1. **What Is Operator Overloading In .net?**

Operator overloading is the most evident example of Polymorphism. In operator overloading, an operator is ‘overloaded’ or made to perform additional functions in addition to what it actually does. For e.g. we can overload the “+” operator to add two complex numbers or two imaginary numbers.  
  
X= a + b //+ is overloaded to add two integers, float etc  
Few operators like :: cannot be overloaded.

1. **What Is Finalize Method In .net?**

Object.Finalize method in .NET is typically used to clean and release unmanaged resources like OS files, window etc. Even though Garbage collector in .NET has the ability to determine the life time of such objects, however, with no knowledge how to clean them. The Finalize method allows an object to clean up such unmanaged resources when the garbage collector wishes to reclaim the memory. However, Finalize method should be avoided until necessary as it affects the performance because reclaiming the memory used by objects with Finalize methods requires at least two garbage collections.

1. **What Are The Concepts Of Dispose Method?**

Dispose method belongs to IDisposable interface. It is used to free unmanaged resources like files, network connection etc.

1. **Concurrency With Aop?**

Concurrency is the system's ability to act with several requests simultaneously, such a way that threads don't corrupt the state of objects when they gain access at the same time.

1. **Transparent Caching With Aop ?**

To get better results in terms of speed and resources used, it's suggested to use a cache. We can store in it the results corresponding to the methods invocations as key-value pairs: method and arguments as key and return object as value.

1. **Security With Aop?**

Security is one of the most important elements of an application. The word "security" covers two concepts: Authentication is the verifi cation's process of a principal's identity; a principal is typically a user. A principal in order to be authenticated provides a credential that is the password. Authorization, on the other hand, is the process of granting authorities, which are usually roles, to an authenticated user.

1. **Aspect-oriented Programming?**

Aspect Oriented Programming or AOP is an interesting concept that can be applied to many of the programming problems we solve everyday. In our Visual Studio team system code we have a lot of web-services and remoting code that essentially does the following

public void MyMethod(int parameter)

{

Trace.EnteredMethod("MyMethod", parameter);

SecurityCheck();

// Bunch of processing

Trace.ExitMethod("MyMethod");

}

1. **What Is Manifest In .net?**

Manifest is nothing but a simple text file used to store metadata information of different .NET assemblies. Manifest file can be saved as a stand alone file of type PE. It can also be stored as an exe or as a dll file containing intermediate language code.

1. **What Is Garbage Collection? How To Force Garbage Collector To Run?**

Garbage collection helps in releasing memory occupied by objects. CLR automatically releases these unused objects.

1. **Explain Boxing And Unboxing In .net.**

Boxing permits any value type to be implicitly converted to type object or to any interface type Implemented by value

1. **What Is Native Image Generator (ngen.exe)?**

Ngen.exe helps in improving performance of managed applications by creating native images and storing them on the.

1. **Explain Clr In Brief.**

CLR stands for Common Language Runtime.   
The CLR is a development platform. It provides a runtime, defines functionality in some libraries, and supports a set of programming languages. The CLR provides a runtime so that the softwares can utilize its services. The CLR Base Class Library allows interaction with the runtime. The CLR supports various programming languages, several standards and is itself been submitted as an open standard.

1. **Describe How A .net Application Is Compiled And Executed.**

From the source code, the compiler generates Microsoft Intermediate Language (MSIL) which is further used for the creation of an EXE or DLL. The CLR processes these at runtime. Thus, compiling is the process of generating this MSIL.  
  
The way you do it in .Net is as follows:  
Right-click and select Build / Ctrl-Shift-B / Build menu, Build command  
F5 - compile and run the application.   
Ctrl+F5 - compile and run the application without debugging.  
  
Compilation can be done with Debug or Release configuration. The difference between these two is that in the debug configuration, only an assembly is generated without optimization. However, in release complete optimization is performed without debug symbols.

1. **Describe The Parts Of Assembly.**

An assembly is a partially compiled code library. In .NET, an assembly is a portable executable and can be an EXE (process assembly) or a DLL (library assembly). An assembly can consist of one or more files or modules in various languages. It is used in deployment, versioning and security.

1. **Overview Of Clr Integration.**

The CLR (Common Language Runtime) integration is hosted in the Microsoft SQL Server 2005. With CLR integration, stored procedures, triggers, user- defined functions, user-defined types, and user-defined aggregates in managed code, etc can be written.

As managed code compiles to native code before executing, significant performance can be achieved.

The SQL Server acts as an operating system for the CLR when it is hosted inside SQL Server.  
  
Following are the steps to build a CLR stored procedure in SQL Server 2005

* + Enable CLR integration in SQL Server 2005
  + Create a CLR stored procedure Assembly
  + Deploy the Assembly in SQL Server 2005
  + Create and execute the CLR stored procedure in SQL Server 2005

1. **Clr Triggers?**

A CLR trigger could be a Date Definition or Date Manipulation Language trigger or could be an AFTER or INSTEAD OF trigger.  
  
Methods written in managed codes that are members of an assembly need to be executed provided the assembly is deployed in SQL 2005 using the CREATE assembly statement.  
  
The Microsoft.SqlServer.Server Namespace contains the required classes and enumerations for this objective.

1. **Steps For Creating Clr Trigger?**

Follow these steps to create a CLR trigger of DML (after) type to perform an insert action:   
• Create a .NET class of triggering action  
• Make an assembly (.DLL) from that Class  
• Enable CLR environment in that database.  
• Register the assembly in SQL Server   
• Create CLR Trigger using that assembly

1. **What Is A Clr (common Language Runtime)?**

Common Language Runtime - It is the implementation of CLI. The core runtime engine in the Microsoft .NET Framework for executing applications. The common language runtime supplies managed code with services such as cross-language integration, code access security, object lifetime management, resouce management, type safety, pre-emptive threading, metadata services (type reflection), and debugging and profiling support. The ASP.NET Framework and Internet Explorer are examples of hosting CLR.   
  
The CLR is a multi-language execution environment. There are currently over 15 compilers being built by Microsoft and other companies that produce code that will execute in the CLR.   
  
The CLR is described as the "execution engine" of .NET. It's this CLR that manages the execution of programs. It provides the environment within which the programs run. The software version of .NET is actually the CLR version.   
  
When the .NET program is compiled, the output of the compiler is not an executable file but a file that contains a special type of code called the Microsoft Intermediate Language (MSIL, now called CIL, Common Intermediate Language). This MSIL defines a set of portable instructions that are independent of any specific CPU. It's the job of the CLR to translate this Intermediate code into a executable code when the program is executed making the program to run in any environment for which the CLR is implemented. And that's how the .NET Framework achieves Portability. This MSIL is turned into executable code using a JIT (Just In Time) complier. The process goes like this, when .NET programs are executed, the CLR activates the JIT complier. The JIT complier converts MSIL into native code on a demand basis as each part of the program is needed. Thus the program executes as a native code even though it is compiled into MSIL making the program to run as fast as it would if it is compiled to native code but achieves the portability benefits of MSIL.

1. **Explain The Concepts Of Cts And Cls(common Language Specification).**

CTS  
When different languages are integrated and want to communicate, it is certain that the languages have their own data types and different declaration styles. CTS define how these different variables are declared and used in run time. E.g. VB offers an Integer data type while C++ offers long data type. Using CTS, these data types are converted to System32 which itself a data type of CLS.  
  
CLS  
Any language(s) that intend to use the Common Language Infrastructure needs to communicate with other CLS-Compliant language. This communication is based on set of rules laid by CLS. These rules define a subset of CTS.

1. **.net Framework**

This includes introduction of .Net framework, .Net framework architecture, role of assembly and GAC.

1. **How Do We Access Crystal Reports In .net?**

When crystal reports are integrated with .NET, data can be accessed using ODBC drivers, ADO drivers, database files like excel, xml etc

1. **What Are The Various Components In Crystal Reports?**

When .NET application uses crystal reports, the following components are required: Report files: .rpt or report files needs to be distributed which can either be compiled into the application (Embedded) or independently (non embedded) from the application.

1. **What Basic Steps Are Needed To Display A Simple Report In Crystal?**

Crystal reports offer a report designer. First, select specific rows and columns from a table. Using the designer, the data on the report can be rearranged and formatted.

1. **Asp.net 2.0 Themes**

One of the neat features of ASP.NET 2.0 is themes, which enable you to define the appearance of a set of controls once and apply the appearance to your entire web application.

1. **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.

1. **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.

1. **What Is A Satellite Assembly?**

When you write a multilingual or multi-cultural application in .NET, and want to distribute the core application separately from the localized modules, the localized assemblies that modify the core application are called satellite assemblies.

1. **.net Code Security**

This includes explanation of code security, Principal object, declarative and imperative security, role-based security, code access security and code group.

1. **.net Assembly**

This article explains .Net assembly, private and shared assembly, satellite assemblies, resource-only assembly, ResourceManager class, strong name, global assembly cache.

1. **.net Debug & Trace**

Here you can learn about break mode, options to step through code in .Net, Debug Vs Trace in .Net, trace class, listeners collection of Trace and Debug objects and Trace Switches.

1. **Who Benefits From Ajax?**

AJAX is employed to improve the user’s experience. A request is made for the initial page rendering. After that, asynchronous requests to the server are made. An asynchronous request is a background request to send or receive data in an entirely nonvisual manner.

1. **What Is Asp.net Ajax?**

ASP.NET AJAX, mostly called AJAX, is a set of extensions of ASP.NET. It is developed by Microsoft to implement AJAX functionalities in Web applications. ASP.NET AJAX provides a set of components that enable the developers to develop applications that can update only a specified portion of data without refreshing the entire page. The ASP.NET AJAX works with the AJAX Library that uses object-oriented programming (OOP) to develop rich Web applications that communicate with the server using asynchronous postback.

1. **What Is .net?**

.NET is a general-purpose software development platform, similar to Java. At its core is a virtual machine that turns intermediate language (IL) into machine code. High-level language compilers for C#, VB.NET and C++ are provided to turn source code into IL. C# is a new programming language, very similar to Java. An extensive class library is included, featuring all the functionality one might expect from a contempory development platform - windows GUI development (Windows Forms), database access (ADO.NET), web development (ASP.NET), web services, XML etc

1. **What Versions Of .net Are There?**
   * The final versions of the 1.0 SDK and runtime were made publicly available around &6pm PST on15-Jan-2002. At the same time, the final version of Visual Studio.NET was made available to MSDN subscribers.
   * .NET 1.1 was released in April 2003, and was mostly bug fixes for 1.0.
   * .NET 2.0 was released to MSDN subscribers in late October 2005, and was officially launched in early November
2. **What Operating Systems Does The .net Framework Run On?**

The runtime supports Windows Server 2003, Windows XP, Windows 2000, NT4 SP6a and Windows ME/98. Windows 95 is not supported. Some parts of the framework do not work on all platforms - for example, ASP.NET is only supported on XP and Windows 2000/2003. Windows 98/ME cannot be used for development

IIS is not supported on Windows XP Home Edition, and so cannot be used to host ASP.NET. However, the ASP.NET Web Matrix web server does run on XP Home.  
The .NET Compact Framework is a version of the .NET Framework for mobile devices, running Windows CE or Windows Mobile. The Mono project has a version of the .NET Framework that runs on Linux.

1. **What Tools Can I Use To Develop .net Applications?**

There are a number of tools, described here in ascending order of cost:

•The .NET Framework SDK is free and includes command-line compilers for C++, C#, and VB.NET and various other utilities to aid development.  
•SharpDevelop is a free IDE for C# and VB.NET.  
•Microsoft Visual Studio Express editions are cut-down versions of Visual Studio, for hobbyist or novice developers.There are different versions for C#, VB, web development etc. Originally the plan was to charge $49, but MS has decided to offer them as free downloads instead, at least until November 2006.  
•Microsoft Visual Studio Standard 2005 is around $300, or $200 for the upgrade.  
•Microsoft VIsual Studio Professional 2005 is around $800, or $550 for the upgrade  
•At the top end of the price range are the Microsoft Visual Studio Team Edition for Software Developers 2005 with MSDN Premium and Team Suite editions.

You can see the differences between the various Visual Studio versions here.

1. **Why Did They Call It .net?**

I don't know what they were thinking. They certainly weren't thinking of people using search tools. It's meaningless marketing nonsense.

1. **What Is The Cli? Is It The Same As The Clr?**

The CLI (Common Language Infrastructure) is the definiton of the fundamentals of the .NET framework - the Common Type System (CTS), metadata, the Virtual Execution Environment (VES) and its use of intermediate language (IL), and the support of multiple programming languages via the Common Language Specification (CLS). The CLI is documented through ECMA -

The CLR (Common Language Runtime) is Microsoft's primary implementation of the CLI. Microsoft also have a shared source implementation known as ROTOR, for educational purposes, as well as the .NET Compact Framework for mobile devices. Non-Microsoft CLI implementations includeMono and DotGNU Portable.NET.

1. **What Is C#?**

C# is a new language designed by Microsoft to work with the .NET framework. In their "Introduction to C#" whitepaper, Microsoft describe C# as follows:  
"C# is a simple, modern, object oriented, and type-safe programming language derived from C and C++. C# (pronounced “C sharp”) is firmly planted in the C and C++ family tree of languages, and will immediately be familiar to C and C++ programmers. C# aims to combine the high productivity of Visual Basic and the raw power of C++."  
Substitute 'Java' for 'C#' in the quote above, and you'll see that the statement still works pretty well.

1. **What Is The Difference Between A Private Assembly And A Shared Assembly?**

The terms 'private' and 'shared' refer to how an assembly is deployed, not any intrinsic attributes of the assembly.

A private assembly is normally used by a single application, and is stored in the application's directory, or a sub-directory beneath. A shared assembly is intended to be used by multiple applications, and is normally stored in the global assembly cache (GAC), which is a central repository for assemblies. (A shared assembly can also be stored outside the GAC, in which case each application must be pointed to its location via a codebase entry in the application's configuration file.) The main advantage of deploying assemblies to the GAC is that the GAC can support multiple versions of the same assembly side-by-side.

Assemblies deployed to the GAC must be strong-named. Outside the GAC, strong-naming is optional.

1. **How Can I Develop An Application That Automatically Updates Itself From The Web?**

For .NET 1.x, use the Updater Application Block. For .NET 2.x, use ClickOnce.

1. **Can I Write My Own .net Host?**

Yes. For an example of how to do this, take a look at the source for the dm.net moniker developed by Jason Whittington and Don Box. There is also a code sample in the .NET SDK called CorHost.

1. **Is It True That Objects Don't Always Get Destroyed Immediately When The Last Reference Goes Away?**

Yes. The garbage collector offers no guarantees about the time when an object will be destroyed and its memory reclaimed.& There was an interesting thread on the DOTNET list, started by Chris Sells, about the implications of non-deterministic destruction of objects in C#. In October 2000, Microsoft's Brian Harry posted a lengthy analysis of the problem. Chris Sells' response to Brian's posting is here.

1. **Should I Implement Finalize On My Class? Should I Implement Idisposable?**

This issue is a little more complex than it first appears. There are really two categories of class that require deterministic destruction - the first category manipulate unmanaged types directly, whereas the second category manipulate managed types that require deterministic destruction. An example of the first category is a class with an IntPtr member representing an OS file handle. An example of the second category is a class with a System.IO.FileStream member.

For the first category, it makes sense to implement IDisposable and override Finalize. This allows the object user to 'do the right thing' by calling Dispose, but also provides a fallback of freeing the unmanaged resource in the Finalizer, should the calling code fail in its duty. However this logic does not apply to the second category of class, with only managed resources. In this case implementing Finalize is pointless, as managed member objects cannot be accessed in the Finalizer. This is because there is no guarantee about the ordering of Finalizer execution. So only the Dispose method should be implemented. (If you think about it, it doesn't really make sense to call Dispose on member objects from a Finalizer anyway, as the member object's Finalizer will do the required cleanup.)

For classes that need to implement IDisposable and override Finalize, see Microsoft's documented pattern.

Note that some developers argue that implementing a Finalizer is always a bad idea, as it hides a bug in your code (i.e. the lack of a Dispose call). A less radical approach is to implement Finalize but include a Debug.Assert at the start, thus signalling the problem in developer builds but allowing the cleanup to occur in release builds.

1. **Do I Have Any Control Over The Garbage Collection Algorithm?**

A little. For example the System.GC class exposes a Collect method, which forces the garbage collector to collect all unreferenced objects immediately.

Also there is a gcConcurrent setting that can be specified via the application configuration file. This specifies whether or not the garbage collector performs some of its collection activities on a separate thread. The setting only applies on multi-processor machines, and defaults to true.

1. **How Can I Find Out What The Garbage Collector Is Doing?**

Lots of interesting statistics are exported from the .NET runtime via the '.NET CLR xxx' performance counters. Use Performance Monitor to view them.

1. **What Is The Lapsed Listener Problem?**

The lapsed listener problem is one of the primary causes of leaks in .NET applications. It occurs when a subscriber (or 'listener') signs up for a publisher's event, but fails to unsubscribe. The failure to unsubscribe means that the publisher maintains a reference to the subscriber as long as the publisher is alive. For some publishers, this may be the duration of the application.

This situation causes two problems. The obvious problem is the leakage of the subscriber object. The other problem is the performance degredation due to the publisher sending redundant notifications to 'zombie' subscribers.

There are at least a couple of solutions to the problem. The simplest is to make sure the subscriber is unsubscribed from the publisher, typically by adding an Unsubscribe() method to the subscriber. Another solution, documented here by Shawn Van Ness, is to change the publisher to use weak references in its subscriber list.

1. **I Want To Serialize Instances Of My Class. Should I Use Xmlserializer, Soapformatter Or Binaryformatter?**

It depends. XmlSerializer has severe limitations such as the requirement that the target class has a parameterless constructor, and only public read/write properties and fields can be serialized. However, on the plus side, XmlSerializer has good support for customising the XML document that is produced or consumed. XmlSerializer's features mean that it is most suitable for cross-platform work, or for constructing objects from existing XML documents.

SoapFormatter and BinaryFormatter have fewer limitations than XmlSerializer. They can serialize private fields, for example. However they both require that the target class be marked with the [Serializable] attribute, so like XmlSerializer the class needs to be written with serialization in mind. Also there are some quirks to watch out for - for example on deserialization the constructor of the new object is not invoked.

The choice between SoapFormatter and BinaryFormatter depends on the application.

BinaryFormatter makes sense where both serialization and deserialization will be performed on the .NET platform and where performance is important. SoapFormatter generally makes more sense in all other cases, for ease of debugging if nothing else.

1. **Xmlserializer Is Throwing A Generic "there Was An Error Reflecting Myclass" Error. How Do I Find Out What The Problem Is?**

Look at the InnerException property of the exception that is thrown to get a more specific error message.

1. **What Is Code Access Security (cas)?**

CAS is the part of the .NET security model that determines whether or not 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 your hard disk.

1. **How Does Cas Work?**

The CAS security policy revolves around two key concepts - code groups and permissions. Each .NET assembly is a member of a particular code group, and each code group is granted the permissions specified in a named permission set.

For example, using the default security policy, a control downloaded from a web site belongs to the 'Zone - Internet' code group, which adheres to the permissions defined by the 'Internet' named permission set. (Naturally the 'Internet' named permission set represents a very restrictive range of permissions.)

1. **I'm Having Some Trouble With Cas. How Can I Troubleshoot The Problem?**

Caspol has a couple of options that might help. First, you can ask caspol to tell you what code group an assembly belongs to, using caspol -rsg. Similarly, you can ask what permissions are being applied to a particular assembly using caspol -rsp

1. **I Can't Be Bothered With Cas. Can I Turn It Off?**

Yes, as long as you are an administrator. Just run:  
caspol -s off

1. **Can I Look At The Il For An Assembly?**

Yes. MS supply a tool called Ildasm that can be used to view the metadata and IL for an assembly.

1. **Can Source Code Be Reverse-engineered From Il?**

Yes, it is often relatively straightforward to regenerate high-level source from IL. Lutz Roeder'sReflector does a very good job of turning IL into C# or VB.NET.

1. **How Can I Stop My Code Being Reverse-engineered From Il?**

You can buy an IL obfuscation tool. These tools work by 'optimising' the IL in such a way that reverse-engineering becomes much more difficult  
Of course if you are writing web services then reverse-engineering is not a problem as clients do not have access to your IL.

1. **Can I Do Things In Il That I Can't Do In C#?**

Yes. A couple of simple examples are that you can throw exceptions that are not derived from System.Exception, and you can have non-zero-based arrays.

1. **Does .net Replace Com?**

 .NET has its own mechanisms for type interaction, and they don't use COM. No IUnknown, no IDL, no typelibs, no registry-based activation. This is mostly good, as a lot of COM was ugly. Generally speaking, .NET allows you to package and use components in a similar way to COM, but makes the whole thing a bit easier.

1. **Is Dcom Dead?**

Pretty much, for .NET developers. The .NET Framework has a new remoting model which is not based on DCOM. DCOM was pretty much dead anyway, once firewalls became widespread and Microsoft got SOAP fever. Of course DCOM will still be used in interop scenarios.

1. **Is Com+ Dead?**

Not immediately. The approach for .NET 1.0 was to provide access to the existing COM+ services (through an interop layer) rather than replace the services with native .NET ones. Various tools and attributes were provided to make this as painless as possible. Over time it is expected that interop will become more seamless - this may mean that some services become a core part of the CLR, and/or it may mean that some services will be rewritten as managed code which runs on top of the CLR.

1. **Can I Use Com Components From .net Programs?**

Yes. COM components are accessed from the .NET runtime via a Runtime Callable Wrapper (RCW). This wrapper turns the COM interfaces exposed by the COM component into .NET-compatible interfaces. For oleautomation interfaces, the RCW can be generated automatically from a type library. For non-oleautomation interfaces, it may be necessary to develop a custom RCW which manually maps the types exposed by the COM interface to .NET-compatible types.

1. **Can I Use .net Components From Com Programs?**

Yes. .NET components are accessed from COM via a COM Callable Wrapper (CCW). This is similar to a RCW (see previous question), but works in the opposite direction. Again, if the wrapper cannot be automatically generated by the .NET development tools, or if the automatic behaviour is not desirable, a custom CCW can be developed. Also, for COM to 'see' the .NET component, the .NET component must be registered in the registry.

1. **Is Atl Redundant In The .net World?**

Yes. ATL will continue to be valuable for writing COM components for some time, but it has no place in the .NET world.

1. **How Do I Stop A Thread?**

There are several options. First, you can use your own communication mechanism to tell the ThreadStart method to finish. Alternatively the Thread class has in-built support for instructing the thread to stop. The two principle methods are Thread.Interrupt() and Thread.Abort(). The former will cause a Thread Interrupted Exception to be thrown on the thread when it next goes into a WaitJoinSleep state. In other words, Thread.Interrupt is a polite way of asking the thread to stop when it is no longer doing any useful work. In contrast, Thread.Abort() throws a ThreadAbortException regardless of what the thread is doing. Furthermore, the ThreadAbortException cannot normally be caught (though the ThreadStart's finally method will be executed). Thread.Abort() is a heavy-handed mechanism which should not normally be required.

1. **How Do I Know When My Thread Pool Work Item Has Completed?**

There is no way to query the thread pool for this information. You must put code into the WaitCallback method to signal that it has completed. Events are useful for this.

1. **Should I Use Readerwriterlock Instead Of Monitor.enter/exit?**

Maybe, but be careful. ReaderWriterLock is used to allow multiple threads to read from a data source, while still granting exclusive access to a single writer thread. This makes sense for data access that is mostly read-only, but there are some caveats. First, ReaderWriterLock is relatively poor performing compared to Monitor. Enter/Exit, which offsets some of the benefits. Second, you need to be very sure that the data structures you are accessing fully support multithreaded read access. Finally, there is apparently a bug in the v1.1 Reader Writer Lock that can cause starvation for writers when there are a large number of readers. Ian Griffiths has some interesting discussion on Reader Writer Lock here and here.

1. **Tracing . Is There Built-in Support For Tracing/logging?**

Yes, in the System.Diagnostics namespace. There are two main classes that deal with tracing - Debug and Trace. They both work in a similar way - the difference is that tracing from the Debug class only works in builds that have the DEBUG symbol defined, whereas tracing from the Trace class only works in builds that have the TRACE symbol defined. Typically this means that you should use System.Diagnostics.Trace.WriteLine for tracing that you want to work in debug and release builds, and System. Diagnostics. Debug. WriteLine for tracing that you want to work only in debug builds.

1. **Can I Redirect Tracing To A File?**

Yes. The Debug and Trace classes both have a Listeners property, which is a collection of sinks that receive the tracing that you send via Debug.WriteLine and Trace.WriteLine respectively. By default the Listeners collection contains a single sink, which is an instance of the DefaultTraceListener class. This sends output to the Win32 OutputDebugString() function and also the System.Diagnostics.Debugger.Log() method. This is useful when debugging, but if you're trying to trace a problem at a customer site, redirecting the output to a file is more appropriate. Fortunately, the TextWriterTraceListener class is provided for this purpose.Here's how to use the Text Writer Trace Listener class to redirect Trace output to a file:

Trace.Listeners.Clear();  
FileStream fs = new FileStream( @"c:\log.txt", FileMode. Create, FileAccess.Write);  
Trace.Listeners.Add( new TextWriterTraceListener( fs ));  
Trace.WriteLine( @"This will be writen to c:\log.txt!" );  
Trace.Flush();

Note the use of Trace.Listeners.Clear() to remove the default listener. If you don't do this, the output will go to the file and OutputDebugString(). Typically this is not what you want, because OutputDebugString() imposes a big performance hit.

1. **Are There Any Third Party Logging Components Available?**

Log4net is a port of the established log4j Java logging component.

1. **Miscellaneous. How Does .net Remoting Work?**

.NET remoting involves sending messages along channels. Two of the standard channels are HTTP and TCP. TCP is intended for LANs only-HTTP can be used for LANs or WANs (internet). Support is provided for multiple message serializarion formats. Examples are SOAP (XML-based) and binary. By default, the HTTP channel uses SOAP (via the .NET runtime Serialization SOAP Formatter), and the TCP channel uses binary (via the .NET runtime Serialization Binary Formatter). But either channel can use either serialization format.

There are a number of styles of remote access:

•SingleCall. Each incoming request from a client is serviced by a new object. The object is thrown away when the request has finished.  
•Singleton. All incoming requests from clients are processed by a single server object.  
•Client-activated object. This is the old stateful (D)COM model whereby the client receives a reference to the remote object and holds that reference (thus keeping the remote object alive) until it is finished with it.

Distributed garbage collection of objects is managed by a system called 'leased based lifetime'. Each object has a lease time, and when that time expires the object is disconnected from the .NET runtime remoting infrastructure. Objects have a default renew time - the lease is renewed when a successful call is made from the client to the object. The client can also explicitly renew the lease.If you're interested in using XML-RPC as an alternative to SOAP, take a look at Charles Cook's XML-RPC.Net.

1. **When Should You Call The Garbage Collector In .net?**

As a good rule, you should not call the garbage collector. However, you could call the garbage collector when you are done using a large object (or set of objects) to force the garbage collector to dispose of those very large objects from memory. However, this is usually not a good practice.

1. **What Is The Difference Between An Event And A Delegate?**

An event is just a wrapper for a multicast delegate. Adding a public event to a class is almost the same as adding a public multicast delegate field. In both cases, subscriber objects can register for notifications, and in both cases the publisher object can send notifications to the subscribers. However, a public multicast delegate has the undesirable property that external objects can invoke the delegate, something we'd normally want to restrict to the publisher. Hence events - an event adds public methods to the containing class to add and remove receivers, but does not make the invocation mechanism public. See this post by Julien Couvreur for more discussion.

1. **What Size Is A .net Object?**

Each instance of a reference type has two fields maintained by the runtime - a method table pointer and a sync block. These are 4 bytes each on a 32-bit system, making a total of 8 bytes per object overhead. Obviously the instance data for the type must be added to this to get the overall size of the object. So, for example, instances of the following class are 12 bytes each:  
class MyInt  
{  
...  
private int x;  
}  
However, note that with the current implementation of the CLR there seems to be a minimum object size of 12 bytes, even for classes with no data (e.g. System.Object). Values types have no equivalent overhead.

1. **Will My .net App Run On 64-bit Windows?**

64-bit (x64) versions of Windows support both 32-bit and 64-bit processes, and corresponding 32-bit and 64-bit versions of .NET 2.0. (.NET 1.1 is 32-bit only)..NET 1.x apps automatically run as 32-bit processes on 64-bit Windows..NET 2.0 apps can either run as 32-bit processes or as 64-bit processes. The OS decides which to use based on the PE header of the executable. The flags in the PE header are controlled via the compiler /platform switch, which allows the target of the app to be specified as 'x86', 'x64' or 'any cpu'. Normally you specify 'any cpu', and your app will run as 32-bit on 32-bit Windows and 64-bit on 64-bit Windows. However if you have some 32-bit native code in your app (loaded via COM interop, for example), you will need to specify 'x86', which will force 64-bit Windows to load your app in a 32-bit process. You can also tweak the 32-bit flag in the PE header using the SDK corflags utility.

1. **.net 2.0 What Are The New Features Of .net 2.0?**

Generics, anonymous methods, partial classes, iterators, property visibility (separate visibility for get and set) and static classes.

1. **What's New In The .net 2.0 Class Library?**

Here is a selection of new features in the .NET 2.0 class library:  
•Generic collections in the System.Collections.Generic namespace.  
•The System.Nullable type. (Note that C# has special syntax for this type, e.g. int? is equivalent to Nullable)  
•The GZipStream and DeflateStream classes in the System.IO.Compression namespace.  
•The Semaphore class in the System.Threading namespace.  
•Wrappers for DPAPI in the form of the ProtectedData and Protected Memory classes in the System. Security. Cryptography namespace.  
•The IPC remoting channel in the System. Runtime. Remoting. Channels.Ipc namespace, for optimised intra-machine communication.

1. **What Is Ajax?**

Ajax stands for Asynchronous Javascript & XML. It is a web technology through which a postback from a client (browser) to the server goes partially, which means that instead of a complete postback, a partial postback is triggered by the Javascript XmlHttpRequest object. In such a scenario, web-application users won't be able to view the complete postback progress bar shown by the browser. In an AJAX environment, it is Javascript that starts the communication with the web server.

Ajax technology in a website may be implemented by using plain Javascript and XML. Code in such a scenario may tend to look little complex, for which the AJAX Framework in .NET can be embedded in ASP.NET web applications.In addition to XML & Javascript, AJAX is also based on DOM-the Document Object Model technology of browsers through which objects of the browser can be accessed through the memory heap using their address. JSON-Javascript Object Notation is also one of the formats used in AJAX, besides XML.

So basically, in an AJAX-based web application, the complete page does not need to reload, and only the objects in context of ajaxification are reloaded. Ajax technology avoids the browser flickering

1. **Can Ajax Be Implemented In Browsers That Do Not Support The Xmlhttprequest Object?**

Yes. This is possible using remote scripts.

1. **Can Ajax Technology Work On Web Servers Other Than Iis?**

Yes, AJAX is a technology independent of web server the web application is hosted on. Ajax is a client (browser) technology.

1. **Which Browsers Support The Xmlhttprequest Object?**

Internet Explorer 5.0+, Safari 1.2, Mozilla 1.0/Firefox, Opera 8.0 +, Netscape 7

1. **What Are Ajax Extensions?**

The ASP.NET Ajax Extensions are set of Ajax-based controls that work in ASP.NET 2 (or above) based applications.

Ofcourse,they also need the Ajax runtime which is actually the Ajax Framework 1.0.

ASP.NET Ajax Extensions 1.0 have to be downloaded to run with ASP.NET 2.0

The new ASP.NET 3.5 Framework comes with the Ajax Library 3.5 (containing the Ajax Extensions 3.5). So in order to use the latest Ajax, simply download .NET 3.5 Framework.

 Summary:  
ASP.NET Ajax Extensions 1.0 -> For ASP.NET 2.0  
ASP.NET Ajax Extensions 3.5 -> For ASP.NET 3.5

1. **What Is Dojo?**

Dojo is a third-party javascript toolkit for creating rich featured applications. Dojo is an Open Source DHTML toolkit written in JavaScript. It builds on several contributed code bases (nWidgets, Burstlib, f(m)), which is why we refer to it sometimes as a "unified" toolkit. Dojo aims to solve some long-standing historical problems with DHTML which prevented mass adoption of dynamic web application development.

1. **How To Handle Multiple Or Concurrent Requests In Ajax?**

For concurrent requests, declare separate XmlHttpRequest objects for each request. For example, for request to get data from an SQL table1, use something like this...  
xmlHttpObject1.Onreadystatechange = functionfromTable1();  
and to get data from another table (say table2) at the same time, use  
xmlHttpObject2.Onreadystatechange = functionfromTable2();  
Ofcourse, the XmlHttpObject needs to be opened & parameters passed too, like as shown below...  
xmlHTTPObject1.open("GET","http://"localhost// " + "Website1/ Default1.aspx" true);  
Note that the last parameter "true" used above means that processing shall carry on without waiting for any response from the web server. If it is false, the function shall wait for a response.

1. **What Is The Role Of Scriptmanager In Ajax?**

ScriptManager class is the heart of ASP.NET Ajax. Before elaborating more on ScriptManager, note that ScriptManager is class and a control (both) in Ajax.

The ScriptManager class in ASP.NET manages Ajax Script Libraries, partial page rendering functionality and client proxy class generation for web applications and services. By saying client proxy class, this means an instance of the Ajax runtime is created on the browser.

This class is defined in the System.Web.Extensions.dll. You will find this DLL in your system's Global Assembly Cache at C:\Windows\Assembly (For XP)

The ScriptManager control (that we may drag on a web form) is actually an instance of the ScriptManager class that we put on a web page. The ScriptManager manages all the ASP.NET Ajax controls on a web page. Following tasks are taken care by the ScriptManager class:  
1 - Managing all resources (all objects/controls) on a web page  
2 - Managing partial page updates  
3 - Download Ajax Script Library to the client (means to the browser). This needs to happen so that Ajax engine is accessible to the browsers javascript code.  
4 - Interacting with UpdatePanel Control, UpdateProgress Control.  
5 - Register script (using RegisterClientScriptBlock)  
6 - Information whether Release OR Debug script is sent to the browser  
7 - Providing access to Web service methods from the script by registering Web services with the ScriptManager control  
8 - Providing access to ASP.NET authentication, role, and profile application services from client script after registering these services with the ScriptManager control  
9 - Enable culture specific display of clientside script.  
10 - Register server controls that implement IExtenderControl and IScriptControl interfaces.

ScriptManager class' EnablePartialRendering property is true by default.

1. **Can We Override The Enablepartialrendering Property Of The Scriptmanager Class?**

Yes. But this has to be done before the init event of the page (or during runtime after the page has already loaded). Otherwise an InvalidOperationException will be thrown.

1. **How To Use Multiple Scriptmanager Controls In A Web Page?**

No. It is not possible to use multiple ScriptManager control in a web page. In fact, any such requirement never comes in because a single ScriptManager control is enough to handle the objects of a web page.

1. **Whats The Difference Between Registerclientscriptblock, Registerclientscriptinclude And Registerclientscriptresource?**

For all three, a script element is rendered after the opening form tag. Following are the differences:  
1-RegisterClientScriptBlock - The script is specified as a string parameter.  
2-RegisterClientScriptInclude - The script content is specified by setting the src attribute to a URL that points to a script file.  
3-RegisterClientScriptResource - The script content is specified with a resource name in an assembly. The src attribute is automatically populated with a URL by a call to an HTTP handler that retrieves the named script from the assembly.

1. **What Are Type/key Pairs In Client Script Registration? Can There Be 2 Scripts With The Same Type/key Pair Name?**

When a script is registered by the ScriptManager class, a type/key pair is created to uniquely identify the script.

For identification purposes, the type/key pair name is always unique for dentifying a script. Hence, there may be no duplication in type/key pair names.

1. **What Are The Modes Of Updation In An Updatepanel? What Are Triggers Of An Updatepanel?**

An UpdatePanel has a property called UpdateMode. There are two possible values for this property:  
1)Always  
2)Conditional

If the UpdateMode property is set to "Always", the UpdatePanel control’s content is updated on each postback that starts from anywhere on the webpage. This also includes asynchronous postbacks from controls that are inside other UpdatePanel controls, and postbacks from controls which are not inside UpdatePanel controls.

If the UpdateMode property is set to Conditional, the UpdatePanel control’s content is updated when one of the following is true:  
1-When the postback is caused by a trigger for that UpdatePanel control.  
2-When you explicitly call the UpdatePanel control's Update() method.  
3-When the UpdatePanel control is nested inside another UpdatePanel control and the parent panel is updated.

When the ChildrenAsTriggers property is set to true and any child control of the UpdatePanel control causes a postback. Child controls of nested UpdatePanel controls do not cause an update to the outer UpdatePanel control unless they are explicitly defined as triggers for the parent panel.

Controls defined inside a node have the capability to update the contents of an UpdatePanel.

If the ChildrenAsTriggers property is set to false and the UpdateMode property is set to Always, an exception is thrown. The ChildrenAsTriggers property is intended to be used only when the UpdateMode property is set to Conditional.

1. **How To Control How Long An Ajax Request May Last?**

Use the ScriptManager's AsyncPostBackTimeout Property.

For example, if you want to debug a web page but you get an error that the page request has timed out, you may set <asp:scriptmanager id="ScriptManager1 " runat=" server" asyncpost backtimeout="9000 "></asp: script manager>

where the value specified is in seconds.

1. **What Is Asp.net Futures?**

ASP.NET AJAX Futures

The new release includes support for managing browser history (Back button support), selecting elements by CSS selectors or classes, and information on accessing “Astoria” Web data services. The Futures (July 2007) release adds:

History support for the Safari browser, inclusion of “titles”, encoding and encrypting of server-side history state and the ability to handle history in the client without a server requirement.

CSS Selectors APIs have been modified to be applicable to W3C recommendations.

A script resource extraction tool that allows you to create script files on disk that originate from embedded resources in assemblies. Important: this version of the browser history feature is now outdated and should not be used. Instead, please download the ASP.NET 3.5 Extensions Preview, which contains the new version.

1. **What Are Limitations Of Ajax?**

1)An Ajax Web Application tends to confused end users if the network bandwidth is slow, because there is no full postback running. However, this confusion may be eliminated by using an UpdateProgress control in tandem.  
2)Distributed applications running Ajax will need a central mechanism for communicating with each other.

1. **How To Trigger A Postback On An Updatepanel From Javascript?**

Call the \_\_doPostBack function. ASP.NET runtime always creates a javascript function named \_\_doPostBack(eventTarget, eventArgument) when the web page is rendered. A control ID may be passed here to specifically invoke updation of the UpdatePanel.

1. **Which Request Is Better With Ajax, Get Or Post?**

AJAX requests should use an HTTP GET request while retrieving data where the data does not change for a given URL requested. An HTTP POST should be used when state is updated on the server. This is in line with HTTP idempotency recommendations and is highly recommended for a consistent web application architecture.

1. **What Is The Asp.net Ajax Framework? What Versions Have Been Released So Far?**

ASP.NET AJAX is a free framework to implement Ajax in asp.net web applications, for quickly creating efficient and interactive Web applications that work across all popular browsers.

The Ajax Framework is powered with

1-Reusable Ajax Controls  
2-Support for all modern browsers  
3-Access remote services and data from the browser without tons of complicated script.

Versions of Ajax release

1-ASP.NET Ajax Framework 1.0 (earlier release to this was called the Atlas)  
2-ASP.NET Ajax Framework 1.0 was available as a separate download for ASP.NET 2.0

1. **What Is The Asp.net Control Toolkit?**

Besides the Ajax Framework (which is the Ajax engine) and Ajax Extensions (which contain the default Ajax controls), there is a toolkit called the Ajax Control Toolkit available for use & download (for free). This is a collection of rich featured, highly interactive controls, created as a joint venture between Microsoft & the Developer Community.

1. **How To Create An Ajax Website Using Visual Studio?**

Using Visual Studio Web Developer Express 2005 & versions above it, Ajax based applications may easily be created. Note that the Ajax Framework & Ajax Extensions should be installed (In case of VS 2005). If using Visual Studio 2008 Web Developer Express or above, Ajax comes along with it (so no need of a separate installation).

1. **How To Make Sure That Contents Of An Updatepanel Update Only When A Partial Postback Takes Place (and Not On A Full Postback)?**

Make use of ScriptManager.IsInAsyncPostBack property (returns a boolean value)

1. **What Are The Requirements To Run Asp.net Ajax Applications On A Server?**

You would need to install ‘ASP.NET AJAX Extensions’ on your server. If you are using the ASP.NET AJAX Control toolkit, then you would also need to add the AjaxControlToolkit.dll in the /Bin folder.

1. **Explain The Updatepanel?**

The UpdatePanel enables you to add AJAX functionality to existing ASP.NET applications. It can be used to update content in a page by using Partial-page rendering. By using Partial-page rendering, you can refresh only a selected part of the page instead of refreshing the whole page with a postback.

1. **Can I Use Asp.net Ajax With Any Other Technology Apart From Asp.net?**

To answer this question, check out this example of using ASP.NET AJAX with PHP, to demonstrate running ASP.NET AJAX outside of ASP.NET. Client-Side ASP.NET AJAX framework can be used with PHP and Coldfusion.

1. **Difference Between Server-side Ajax Framework And Client-side Ajax Framework?**

ASP.NET AJAX contains both a server-side Ajax framework and a client-side Ajax framework. The server-side framework provides developers with an easy way to implement Ajax functionality, without having to possess much knowledge of JavaScript. The framework includes server controls and components and the drag and drop functionality. This framework is usually preferred when you need to quickly ajaxify an asp.net application. The disadvantage is that you still need a round trip to the server to perform a client-side action.

The Client-Side Framework allows you to build web applications with rich user-interactivity as that of a desktop application. It contains a set of JavaScript libraries, which is independent from ASP.NET. The library is getting rich in functionality with every new build released.

1. **How Can You Debug Asp.net Ajax Applications?**

Explain about two tools useful for debugging: Fiddler for IE and Firebug for Mozilla.

1. **Can We Call Server-side Code (c# Or Vb.net Code) From Javascript?**

Yes. You can do so using PageMethods in ASP.NET AJAX or using webservices.

1. **Can You Nest Updatepanel Within Each Other?**

Yes, you can do that. You would want to nest update panels to basically have more control over the Page Refresh.

1. **How Can You To Add Javascript To A Page When Performing An Asynchronous Postback?**

Use the ScriptManager class. This class contains several methods like the RegisterStartupScript(), Register Client ScriptBlock(), Register Client Script Include(), Register Array Declaration(), Register Client Script Resource(), Register Expand oAttribute(), Register On SubmitStatement() which helps to add javascript while performing an asynchronous postback.

1. **Explain Differences Between The Page Execution Lifecycle Of An Asp.net Page And An Asp.net Ajax Page?**

In an asynchronous model, all the server side events occur, as they do in a synchronous model. The Microsoft AJAX Library also raises client side events. However when the page is rendered, asynchronous postback renders only the contents of the update panel, where as in a synchronous postback, the entire page is recreated and sent back to the browser.

1. **Is The Asp.net Ajax Control Toolkit(ajaxcontroltoolkit.dll) Installed In The Global Assembly Cache?**

No. You must copy the AjaxControlToolkit.dll assembly to the /Bin folder in your application.

1. **What Role "#&&" Plays In A Querysting?**

# works as a fragment delimeter in a querysting. With which you can delimit history state.  
While && preceeds, state information in query string.

1. **Is There Any Property Names "isnavigating"?**

Yes, it is available when you are managing browser history. With this property of “IsNavigating”, you can determine if postback is occurred for navigation purpose or for some other. Its set to true if its navigation call.

1. **Tell Name Of All The Control Of Ajax?**

There are 5 controls.  
1.ScriptManager  
2.UpdatePanel  
3.UpdateProgress  
4.Timer  
5.ScriptManageProxy

1. **If There Are Multiple Update Panels On The Page Say Upd1 And Upd2. There Is A Button Placed In Upd1. How Can You Stop Upd2 To Update When Button Placed In Upd1 Is Clicked?**

There is a property called UpdateMode which takes two values  
1.Always : Default value of this property.  
2.Conditional

When set to conditional, then that updatepanel's content only gets updated when control placed in that update panel does a postback. Control placed in other update panel will not affect this updatepanel.

1. **How Many Types Of Triggers Are There In Update Panel?**

There are 2 types of triggers.

1.PostBackTrigger :It does a full postback. This is useful when any such control which placed within updatePanel but it cannot work asynchronously. Like File Upload Control.  
2.AsyncPostBackTrigger :- It does partial post back asynchronously.

1. **What Is The Displayafter Property In Updateprogress Control?**

Displayafter property specifies after how many seconds the loading image needs to be displayed in ajax postback. It takes values in seconds.

1. **Is It Compulsory To Have Script Manager On The Page When You Are Using Any Control Of Ajax Control Tool Kit?**

Yes. Page needs to have a script manager for ajax control tool kit controls to work.

1. **Which Property Needs To Be Set For Script Manager Control To Extend The Time Before Throwing Time Out Expection If No Response Is Received From The Server?**

AsyncPost BackTimeout Property needs to set which gets or sets a value that indicates the time, in seconds, before asynchronous postback time out if no response is received from the server.

<asp :script manager id = "script Manager1" runat = "server" async = "" postback = "" error = "" message = "We can not serve your request at this moment.Please try later." asyncpost back timeout = "36000" > < /asp :script manager >

The default value of this property is 90 second. We can also set the user defined error message using asyncpostbackerrormessage property (as shown in above code) for time out.

1. **To Create Browser History Point Using Client Script, We Make Call Method.....**

Sys.Application.addHistoryPoint. This method is used while adding history point to browser stack.

1. **Default Value Of Enablehistory Attribute In Scriptmanager Is,**

False

1. **What Is A Design Pattern?**

Design Pattern is a re-usable, high quality solution to a given requirement, task or recurring problem. Further, it does not comprise of a complete solution that may be instantly converted to a code component, rather it provides a framework for how to solve a problem.

In 1994, the release of the book Design Patterns, Elements of Reusable Object Oriented Software made design patterns popular.

Because design patterns consist of proven reusable architectural concepts, they are reliable and they speed up software development process.

Design Patterns are in a continious phase of evolution, which means that they keep on getting better & better as they are tested against time, reliability and subjected to continious improvements. Further, design patterns have evolved towards targeting specific domains. For example, windows-based banking applications are usually based on singleton patterns, e-commerce web applications are based on the MVC (Model-View-Controller) pattern.

Design Patterns are categorized into 3 types:

* + Creational Patterns
  + Structural Patterns
  + Behavioral Patterns

1. **What Are Creational Design Patterns?**

The Creational Design Patterns focus on how objects are created and utilized in an application. They tackle the aspects of when and how objects are created, keeping in mind whats the best way these objects should be created.

Listed below are some of the commonly known Creational Design Patterns:  
>>>Abstract Factory Pattern  
>>>Factory Pattern  
>>>Builder Pattern  
>>>Lazy Pattern  
>>>Prototype Pattern  
>>>Singleton Pattern

1. **Whats The Difference Between Abstract Factory Pattern And Factory Pattern?**

In an abstract factory design, a framework is provided for creating sub-components that inherit from a common component. In .NET, this is achieved by creating classes that implement a common interface or a set of interfaces, where the interface comprises of the generic method declarations that are passed on to the sub-components. TNote that not just interfaces, but even abstract classes can provide the platform of creating an application based on the abstract factory pattern.

Example, say a class called CentralGovernmentRules is the abstract factory class, comprised of methods like Should Have Police() and Should Have Courts(). There may be several sub-classes like State1Rules, State2Rules etc. created that inheriting the class CentralGovernmentRules, and thus deriving its methods as well.

Note that the term "Factory" refers to the location in the code where the code is created.

A Factory Pattern is again an Object creation pattern. Here objects are created without knowing the class of the object. Sounds strange? Well, actually this means that the object is created by a method of the class, and not by the class's constructor. So basically the Factory Pattern is used wherever sub classes are given the priviledge of instantiating a method that can create an object.

1. **Describe The Builder Design Pattern**

In a builder design pattern, an object creation process is separated from the object design construct. This is useful becuase the same method that deals with construction of the object, can be used to construct different design constructs.

1. **What Is The Lazy Design Pattern?**

The approach of the Lazy Design Pattern is not to create objects until a specific requirement matches, and when it matches, object creation is triggered. A simple example of this pattern is a Job Portal application. Say you register yourself in that site thus filling up the registration table, only when the registration table is filled, the other objects are created and invoked, that prompt you to fill in other details too, which will be saved in other tables.

1. **What Is The Prototype Design Pattern?**

A prototype design pattern relies on creation of clones rather than objects. Here, we avoid using the keyword 'new' to prevent overheads.

1. **What Is The Singleton Design Pattern?**

The Singleton design pattern is based on the concept of restricting the instantiation of a class to one object. Say one object needs to perform the role of a coordinator between various instances of the application that depend on a common object, we may design an application using a Singleton. Usage of Singleton patterns is common in Banking, Financial and Travel based applications where the singleton object consists of the network related information.

A singleton class may be used to instantiate an object of it, only if that object does not already exist. In case the object exists, a reference to the existing object is given. A singleton object has one global point of access to it.

An ASP.NET Web Farm is also based on the Singleton pattern. In a Web Farm, the web application resides on several web servers. The session state is handled by a Singleton object in the form of the aspnet\_state.exe, that interacts with the ASP.NET worker process running on each web server. Note that the worker process is the aspnet\_wp.exe process. Imagine one of the web servers shutting down, the singleton object aspnet\_state.exe still maintains the session state information across all web servers in the web farm.

In .NET, in order to create a singleton, a class is created with a private constructor, and a "static readonly" variable as the member that behaves as the instance.

1. **What Are Structural Design Patterns?**

A structural design pattern establishes a relationship between entities. Thus making it easier for different components of an application to interact with each other. Following are some of the commonly known structural patterns:

>>> Adapter Pattern-Interfaces of classes vary depending on the requirement.  
>>> Bridge Pattern-Class level abstraction is separated from its implementation.  
>>> Composite Pattern-Individual objects & a group of objects are treated similarly in this approach.  
>>> Decorator Pattern-Functionality is assigned to an object.  
>>> Facade Pattern-A common interface is created for a group of interfaces sharing a similarity.  
>>> Flyweight Pattern-The concept of sharing a group of small sized objects.  
>>> Proxy Pattern-When an object is complex and needs to be shared, its copies are made. These copies are called the proxy objects.

1. **What Are The Different Types Of Proxy Patterns?**

1-Remote Proxy - A reference is given to a different object in a different memory location. This may be on a different or a same machine.  
2-Virtual Proxy - This kind of object is created only & only when really required because of its memory usage.  
3-Cache Proxy - An object that behaves as a temporary storage so that multiple applications may use it. For example, in ASP.NET when a page or a user control contains the OutputCache directive, that page/control is cached for some time on the ASP.NET web server.

1. **What Is A Behavioral Design Pattern?**

Behaviorial design patterns focus on improving the communication between different objects. Following are different types of behavioral patterns:  
>>> Chain Or Responsibilities Pattern-In this pattern, objects communicate with each other depending on logical decisions made by a class.  
>>> Command Pattern-In this pattern, objects encapsulate methods and the parameters passed to them.  
>>> Observer Pattern-Objects are created depending on an events results, for which there are event handlers created.

1. **What Is The Mvc Pattern (model View Controller Pattern)?**

The MVC Pattern (Model View Controller Pattern) is based on the concept of designing an application by dividing its functionalities into 3 layers. Its like a triad of components. The Model component contains the business logic, or the other set of re-usable classes like classes pertaining to data access, custom control classes, application configuration classes etc. The Controller component interacts with the Model whenever required. The control contains events and methods inside it, which are raised from the UI which is the View component.

Consider an ASP.NET web application. Here, all aspx, ascx, master pages represent the View.

he code behind files (like aspx.cs, master.cs, ascx.cs) represent the Controller.

The classes contained in the App\_Code folder, or rather any other class project being referenced from this application represent the Model component.

Advantages:  
\*Business logic can be easily modified, without affecting or any need to make changes in the UI.  
\*Any cosmetic change in the UI does not affect any other component.

1. **What Is The Gang Of Four Design Pattern?**

The history of all design patterns used in modern day applications derive from the Gang of Four (GoF) Pattern. Gang of Four patterns are categorized into 3 types:  
1-Creational  
2-Structural  
3-Behavioral

The term "Gang of Four"(or "GoF" in acronym) is used to refer to the four authors of the book Design Patterns: Elements of Reusable Object-Oriented Software. The authors are Erich Gamma, Ralph Johnson, Richard Helm and John Vlissides.

1. **When Should Design Patterns Be Used?**

While developing software applications, sound knowledge of industry proven design patterns make the development journey easy and successful. Whenever a requirement is recurring, a suitable design pattern should be identified. Usage of optimal design patterns enhance performance of the application. Though there are some caveats. Make sure that there are no overheads imposed on a simple requirement, which means that design patterns should not be unnecessarily be used.

1. **How Many Design Patterns Can Be Created In .net?**

As many as one can think. Design patterns are not technology specific, rather their foundation relies on the concept of reusability, object creation and communication. Design patterns can be created in any language.

1. **Describe The Ajax Design Pattern.**

In an Ajax Design Pattern, partial postbacks are triggered asyncronously to a web server for getting live data. A web application would not flicker here, and the web site user would not even come to know that a request is being sent to the web server for live data.

Such a design pattern is used in applications like Stock Market Websites to get live quotes, News Websites for live news, Sports websites for live scores etc.

1. **Explain The .net Framework.**

.NET framework is a foundation calss on which you can build robust applications .This framework comprises of web forms,window forms andconsole applications..NET framework is basically a collection of services and classes.This exists as a layer between .NET applications and underlying operating systems.  
  
.NET Framework is an Integrity of Windows Component which is used to develop Web Applications, Windows, Console Application and Webservices.

1. **Describe The .net Framework Architecture.**

.Net framework has two components:  
  
1. .Net framework class library  
2. Common language runtime.

1. **What Are The Components Of The .net Framework.**

.NET Framework provides enormous advantages to software developers in comparison to the advantages provided by other platforms. Microsoft has united various modern as well as existing technologies of software development in .NET Framework. These technologies are used by developers to develop highly efficient applications for modern as well as future business needs. The following are the key components of .NET Framework:  
  
\* .NET Framework Class Library  
\* Common Language Runtime  
\* Dynamic Language Runtimes (DLR)  
\* Application Domains  
\* Runtime Host  
\* Common Type System  
\* Metadata and Self-Describing Components  
\* Cross-Language Interoperability  
\* .NET Framework Security  
\* Profiling  
\* Side-by-Side Execution

1. **Explain The Role Of Assembly In The .net Framework.**

.Net Framework keeps executable code or DLL in the form of assembly. .Net Framework maintains multiple versions of the application in the system through assembly. The assemblies have MSIL code and manifest that contains metadata. The metadata contains version information of the assembly.

1. **Describe The Gac In The .net Framework.**

.Net Framework provides Global Assembly cache, a machine-wide cache. It stores shared assemblies that can be accessed by multiple languages.

1. **What Is The Advantage Of Packaging Over Xcopy In .net?**

Define connected and disconnected data access in ADO.NET| Describe CommandType property of a SQL Command in ADO.NET.|Define Dataview component of ADO.NET.|What are the ways to create connection in ADO.NET?|Access database at runtime using ADO.NET

1. **.net Code Security Interview Questions With Answers**

What is code security? What are the types? | Define Principal object. | Define declarative and imperative security. | Define role-based security. | Explain code access security. | What is Code group? | Define the use of Caspol.exe.

1. **.net Debug & Trace Interview Questions With Answers**

What is break mode? What are the options to step through code? | Debug Vs Trace. | Define trace class. Define Listeners collection of Trace and Debug objects. | Define Trace Switches.

1. **What Is Asp.net 2.0 Ajax?**

ASP.NET 2.0 AJAX is an AJAX-oriented .NET library that runs on .NET 2.0. Though ASP.NET 2.0 AJAX is an AJAX library and can be used to perform AJAX operations, it is really much more. ASP.NET 2.0 AJAX offers many of the same types of features of the server-side ASP.NET.

1. **The Components In The Asp.net 2.0 Ajax Packaging?**

The packaging of ASP.NET 2.0 AJAX can be fairly confusing. The basics of the packaging are.

* + ASP.NET 2.0 AJAX Extensions 1.0—The ASP.NET 2.0 AJAX Extensions 1.0, also referred to as the RTM/Core code, is an independent download. This contains the functionality that will receive support from Microsoft in the initial release of the product. The source code is available.
  + ASP.NET AJAX Futures Community Technology Preview (CTP) — The ASP.NET 2.0 AJAX framework contains a set of functionality that is experimental in nature. This functionality will eventually become integrated with the RTM/Core code. During the initial release, the Futures CTP functionality will be a separate download from the RTM/Core bits. This will not receive specific support from Microsoft beyond community-based support. The CTP bits require that the RTM/Core bits already be installed for the CTP bits to be installed. The CTP is also referred to as Value-Added Bits.
  + Microsoft AJAX Library—The Microsoft AJAX Library is a set of JavaScript client libraries that make up the standard download to a web browser and provide much of the support for AJAX in the client. These libraries will work with a non-IIS server and are available as a separate download. This library is included in the ASP.NET 2.0 AJAX Extensions 1.0 download as well as being available as a separate download.
  + ASP.NET AJAX Control Toolkit — The AJAX Control Toolkit is a separate download that provides a set of client-side GUI widgets that integrate with the ASP.NET 2.0 AJAX framework. The toolkit is licensed separately from the framework and includes the source code for developers who would like to review the source.

1. **What Are The Benefits Of Ajax?**

Ajax stands for "Asynchronous JavaScript and XML." It is a term used to refer to several related Web development tools. Ajax is one of several technologies comprising Web 2.0 and is designed to create efficient, interactive Web content and dynamic online user interfaces without requiring a separate desktop download. For both users, designers and businesses, Ajax provides a number of benefits over traditional Web design systems.

1. **What's The .net Collection Class That Allows An Element To Be Accessed Using A Unique Key?**

HashTable.

1. **Balancing Client And Server Programming With Asp.net Ajax**

Without the advanced use of JavaScript running in the browser, web applications have their logic running on the server. This means a lot of page refreshes for potentially small updates to the user’s view. With AJAX, much of the logic surrounding user interactions can be moved to the client. This presents its own set of challenges. Some examples of AJAX use include streaming large datasets to the browser that are managed entirely in JavaScript. While JavaScript is a powerful language, the debugging facilities and options for error handling are very limited. Putting complex application logic on the client can take a lot of time, effort, and patience. ASP.NET AJAX allows you to naturally migrate some parts of the application processing to the client while leveraging partial page rendering to let the server control some aspects of the page view.

1. **Asp.net 2.0 Features?**

ASP.NET 2.0 was designed to make web development easier and quicker.  
  
Design goals for ASP.NET 2.0:

* + Increase productivity by removing 70% of the code
  + Use the same controls for all types of devices
  + Provide a faster and better web server platform
  + Simplify compilation and installation
  + Simplify the administration of web applications

1. **What Is The Smallest Unit Of Execution In .net?**

An Assembly.

1. **Will The Finally Block Get Executed If An Exception Has Not Occurred?**

Yes.

1. **Asp.net Application Life Cycle**

This topic describes the application life cycle for ASP.NET applications that are running in IIS 7.0 in Integrated mode and with the .NET Framework 3.0

1. **What's Difference Between "optimistic" And "pessimistic" Locking?**
   * In pessimistic locking when user wants to update data it locks the record and till then no one can update data. Other user’s can only view the data when there is pessimistic locking.
   * In optimistic locking multiple users can open the same record for updating, thus increase maximum concurrency. Record is only locked when updating the record. This is the most preferred way of locking practically. Now a days in browser based application it is very common and having pessimistic locking is not a practical solution.
2. **Differences Between "dataset" And "datareader".**

Dataset:

* + DataSet object can contain multiple rowsets from the same data source as well as from the relationships between them.
  + Dataset is a disconnected architecture
  + Dataset can persist data.
  + A DataSet is well suited for data that needs to be retrieved from multiple tables.
  + Due to overhead DatsSet is slower than DataReader.

Datareader

* + DataReader provides forward-only and read-only access to data.
  + Datareader is connected architecture. It has live connection while reading data
  + Datareader can not persist data.
  + Speed performance is better in DataReader.

1. **What Are The Steps Involved To Fill A Dataset?**

a. Create a connection object.  
b. Create an adapter by passing the string query and the connection object as parameters.  
c. Create a new object of dataset.  
d. Call the Fill method of the adapter and pass the dataset object.   
  
Example:  
VB.NET Code:-  
Dim strSQL as String  
strSQL = "SELECT \* from tbl"  
Dim sqlCmd As New SqlCommand(strSQL, sqlConn)  
Dim sda As New SqlDataAdapter(sqlCmd)  
Dim ds As New DataSet  
sda.Fill(ds)

1. **Describe The .net Base Class Library.**

The .NET Framework class library escribe is a library of classes, interfaces, and value types.  
This library system functionalities and is the foundation of .NET Framework applications, components, and controls.

1. **What Is The Difference Between Value Types And Reference Types?**

Value types:  
\* Value types can be created at compile time.  
\* Stored in stack memory.  
\* Garbage collector can't access the stack  
\* value types holds the data directly  
\* No default values will be stored in value types  
\* Examples for value types: Predefined datatypes,structures,enums   
  
Reference types:  
\* Reference types can be created at run time.  
\* Stored in heap memory  
\* Garbage collector can access heap  
\* Reference types holds the data indiredtly  
\* Reference types holds default value  
\* Examples for reference types: Classes,objects,Arrays,Indexers,Interfaces

1. **Can You Store Multiple Data Types In System.array?**

No.

1. **Explain An Object, Class And Method.**

An object is an entity that keeps together state and behaviors. For instance, a car encapsulates state such as red color, 900 cc etc and behaviors as 'Start', 'Stop' etc., so does an object.  
  
An object is an instance of a class. If you consider “Dog” as a class, it will contain all possible dog traits, while object “German Shepherd” contains characteristics of specific type of dog.  
  
A class represents description of objects that share same attributes and actions. It defines the characteristics of the objects such as attributes and actions or behaviors. It is the blue print that describes objects.   
  
Method is an object’s behavior. If you consider “Dog” as an object then its behaviors are bark, walk, run etc.

1. **What Is The Difference Between Localization And Globalization?**

Globalization is process of identifying how many resources needs to be localized to adopt a multiple culture support, while Localization is actual process of translating resource to a specific culture. So Localization is the part of Globalization.

1. **What Is Unicode?**

Unicode (UCS-2 ISO 10646) is a 16-bit character encoding that contains all of the characters (216 = 65,536 different characters total) in common use in the world's major languages, including Vietnamese. The Universal Character Set provides an unambiguous representation of text across a range of scripts, languages and platforms. It provides a unique number, called a code point (or scalar value), for every character, no matter what the platform, no matter what the program, no matter what the language. The Unicode standard is modeled on the ASCII character set. Since ASCII's 7-bit character size is inadequate to handle multilingual text, the Unicode Consortium adopted a 16-bit architecture which extends the benefits of ASCII to multilingual text.   
  
Unicode characters are consistently 16 bits wide, regardless of language, so no escape sequence or control code is required to specify any character in any language. Unicode character encoding treats symbols, alphabetic characters, and ideographic characters identically, so that they can be used simultaneously and with equal facility. Computer programs that use Unicode character encoding to represent characters but do not display or print text can (for the most part) remain unaltered when new scripts or characters are introduced.   
  
The Unicode Standard has been adopted by such industry leaders as Apple, HP, IBM, Microsoft, Oracle, SAP, Sun, Sybase, Unisys, and many others. Unicode is required by modern standards such as XML, Java, .NET, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc., and is the official way to implement ISO/IEC 10646. It is supported in many operating systems, all modern browsers, and many other products. The emergence of the Unicode Standard, and the availability of tools supporting it, offers significant cost savings over the use of legacy character sets. It allows data to be transported through many different systems without corruption.

1. **What Are Resource File And How Do We Generate Resource File?**

Resource files are used to separate the implementation of the application from its User Interface. Thus, it eliminates the need to change a various sections of code to add a different language.

1. **Describe How To Implement Globalization And Localization In The Use Interface In .net.**

Globalization is the process of making an application that supports multiple cultures without mixing up the business logic and the culture related information of that application.   
  
Localization involves adapting a global application and applying culture specific alterations to it.   
  
The classes and the interfaces provided by the System.Resources allow storing culture specific resources.   
  
The ResourceManager class performs:

* + A look up for culture-specific resources
  + Provision of resource fallback when a localized resource does not exist
  + Supports for resource serialization

1. **Explain How To Prepare Culture-specific Formatting In .net.**

The CutureInfo class can be used for this purpose. It represents information about a specific culture including the culture names, the writing system, and the calendar. It also provides an access to objects that provide information for common operations like date formatting and string sorting.

1. **Explain The Use Of Resource Manager Class In .net.**

The ResourceManager class performs:

* + A look up for culture-specific resources
  + Provision of resource fallback when a localized resource does not exist
  + Supports for resource serialization

1. **Asp.net Globalization-localization**

Here you have description of globalization, localization and their approaches in ASP.NET. It also describes resource files and satellite assemblies.

1. **Asp.net Overview**

This article includes brief about ASP.NET, advantages of ASP.NET, navigation sequence of ASP.NET web form, web Form components, .NET framework, event handlers in ASP.NET, web form events, server control events in ASP.NET, and server controls vs. HTML controls, validation controls, navigation, and steps to store cookies, ways to authenticate and authorize users in ASP.NET etc.

1. **What's The Top .net Class That Everything Is Derived From?**

System.Object.

1. **What Kinds Of Fonts Are Supported With Silverlight?**

Beyond standard and western fonts, Silverlight also supports East Asian characters, double-byte characters, and can work with any East Asian font or Middle Eastern font by using the glyphs element and a supporting TrueType font file that supports the requested glyph.

1. **How Much Is The Pay-for-use Service If I Chose Not To Use Microsoft-sponsored Advertising?**

We’re not prepared to discuss the final pricing of the nonadvertising-based product at this time except to say that it will be extremely cost competitive. The advertising-based product will continue to be free in perpetuity.

1. **What Is The Difference Between Silverlighttm Streaming And Other Video Sharing Services?**

SilverlightTM Streaming is focused on developers who want to build their own media-rich applications or Web sites. Unlike other video sharing services, there are no third-party branding requirements for the use of SilverlightTM Streaming, and the developer is in full control over their rich media experience within the context of their Web site.

1. **Why Is This Service Branded With Windows Livetm?**

This service is part of the Windows LiveTM Platform.

1. **It's Free-what's The Catch?**

There is no catch. This is a new offering designed to accelerate the development of the next generation of media rich applications.

1. **Can I Tap Into Other Windows Livetm Services?**

Yes, customers are able to use Windows Live IDTM and other Windows Live APIs today and in the future. Silverlight provides a great platform to consume these services.

1. **Do You Support Digital Rights Management To Protect My Videos?**

In the future, SilverlightTM Streaming will provide support for DRM-encoded video as an optional paid turnkey offering.

1. **How Does The Service Stream Content?**

Content is streamed progressively using a progressive download mechanism today. Active streaming support using Windows Media Services is being considered based on customer feedback in the future.

1. **Can I Stream Live Content/events?**

No, the service only supports on-demand content today. Customers requiring additional capabilities are encouraged to contact a Windows Media Streaming Hosting Provider.

1. **Do I Need To Have The Latest Version Of Windows Media Player Installed?**

No. Silverlight is completely independent and when installed is less than 2 MB in size.

1. **What Audio Or Video Formats Are Supported In Silverlight?**

Silverlight supports Windows Media Audio and Video (WMA, WMV7–9) and VC-1, as well as MP3 audio. Additional formats may be available by the final release based on customer feedback.

1. **Will Silverlight Support All The Codecs Windows Media Player Supports?**

Since Silverlight is a lightweight cross-platform technology, it only carries the most common codecs that are needed for Web playback. However, we are gathering information from customers about the needed codecs and can update Silverlight when necessary.

1. **Will Silverlight Support Live Streaming Events As Well As Downloading Media?**

Yes, in the final release. The February CTP is optimized for progressive "download and play" scenarios to test the platform.

1. **Does Silverlight Support Mpeg4 And H.264 Video, Or Advanced Audio Coding (aac) Audio, Or Flash Video?**

No. However, content from many of these formats can be transcoded into formats that are supported by Silverlight, such as by an automated server function (many available third-party solutions support this workflow), and then incorporated into a Silverlight-based application.

1. **Explain Silverlight Architecture.**

Silver light is a plug-in used for different platforms and browsers. It is designed for offering media experiences based on .net platform.

1. **Difference Between Wpf And Silverlight**

In terms of features and functionality, Silver light is a sub set of Windows Presentation Foundation.

1. **What Are The Limitations Of Using External Fonts In Silverlight?**

One of the major challenges is to use a downloader and some of the SetFontSource methods on a TextBlock to perform it.

1. **Describe How To Perform Event Handling In Silver Light**

Event handling is performed in two event cases – Input events and Non-input events.

1. **Explain How To Add The Reference Of A Class Library Project In Silverlight Application Project**

The following is the process for adding the reference library project: After defining business object classes in another project.

1. **What Is Silverlight.js File? Explain With An Example.**

Silverlight.js file is a Java Script helper file. It supports for adding a Silverlight application to a web page through Java Script. It has a number of methods defined to help with, most importantly the create Object and createObjectEx. The following are certain notable functions and event handlers :   
-getSilverlight, isBrowserRestartRequired, isInstalled, onGetSilverlight, onSilverlightInstalled, WaitForInstallCompletion.

The Silverlight.js file is installed along with Silverlight 2 SDK.

Using Visual Studio 2008, one can create a quick sample project by selecting File->New Project and selecting Silverlight Application. After clicking OK, select “Automatically generate a test page to host Silverlight at build time”. Click on OK. Right click on the new project and add an HTML page.

1. **What Is A .xap File? Explain With An Example.**

A .xap file is an application package based on Silverlight which will be generated when the Silverlight project is built. This file helpful in creating heavily client based Silverlight applications.

1. **Explain How Can Silverlight Use Asx Files.**

An ASX file is an XML file in which media files are specified in the playlist. Using ASX files in silver is pretty simple. Assign the ‘source’ property of a MediaElement object to the ASX file name.

1. **Explain Silverlight Application Life-cycle**

The entry point of Silverlight applications is Silverlight Application class. It provides various services which is commonly needed by Silverlight application.

1. **What Is The Role Of Silverlight Plugin In The Silverlight Application Life-cycle?**

The Silverlight plug-in loads the core services of Silverlight followed by Silverlight Common Language Runtime. CLR creates domains for applications.

1. **Explain The Purpose Of Storyboard.targetproperty.**

Using Storyboard.TargetProperty, the properties of an object can be assigned with values. The objects are referred by Storyboard. TargetName attribute. The following snippet illustrates the change of width and color of a rectangle object.

1. **Why Is Xap Important?**

Using XAP, Silverlight applications which are heavily client based can be created by managing code. The managed code, benefits of using the tools such as Visual Studio 2008 with Silverlight Tools Beta version 2 , are utilized by using XAP.

1. **How Does Xap Work?**

The .xap file is used for transferring and containing the assemblies and resources of an application with managed code. This code must be written within the Silverlight browser plugin.   
  
Once the .xap file is created, the Silverlight 2 plug-in will download the file and executes in a separate work space

1. **Explain The Use Of Clientbin Folder In Silverlight.**

The ClientBin folder is used for placing .xap file of a Silverlight application. This folder can be kept anywhere in the web application.

1. **What Is Clipping In Silverlight?**

Clipping is a modification to a given image / object based on the geometry type–like a line, rectangle, ellipse or even a group geometry objects.

1. **What Is The Parent Xaml Tag Of Silverlight Page? Explain Its Purposes.**

The’UserConrol’ is the parent xaml tag of a Silverlight page. All other tags are authored under UserControl tag. Developers are given a facility to implement new custom controls and create re-usable user controls.

1. **Explain With Example How To Change The Default Page Of The Silverlight Application.**

The RootVisual property of Application\_Startup event in App.xaml file needs to be set to change the default Silverlight application page. The following code snippet sets the property.

1. **How Many Xaml Files Are Created When You Create A New Project In Silverlight Through Visual Studio And What Are The Uses Of Those Files?**

There are two xaml files are created when a new project in Silverlight is created through Visual Studio.

1. **What Are The Programming Language That Can Be Used To Write The Backend Of The Silverlight Application?**

Visual Basic or Visual C# can be used for authoring code for the backend of the Silverlight application.

1. **Explain How To Set Silverlight Contents Width As 100%.**

Usually the UserConrol will be spread full screen. The contents width and height can also be set by using width and height attributes.

1. **Can You Provide A List Of Layout Management Panels And When You Will Use Them?**

The following are the list of Layout Management Panels: 1. Canvas Panel: Simple layouts use canvas panel and when there is no requirement of resizing the panel. The controls will overlap each other at the time of resizing the panel.

1. **Explain How To Apply Style Elements In A Silverlight Project?**

Application resources utilize the style elements for supporting the forms. The App.xaml file could be used to contain an application resource XML construct. Each style’s target type is set to the control that needs the style.

1. **What Are The Main Features And Benefits Of Silverlight?**

The following are the features of SilverLight: 1. Built in CLR engine is available for delivering a super high performance execution environment for the browser.

1. **When Would One Use Silverlight Instead Of Asp.net Ajax?**

Silverlight media experiences and Rich Internet Applications can be enhanced by the existing ASP.NET AJAX applications. Web applications and ASP.NET AJAX technologies are integrated in Silverlight.

1. **True Or False: A Web Service Can Only Be Written In .net?**

False

1. **Does Silverlight Have A System.console Class? Why?**

Yes. Silverlight have System.Console class. It is cocooned in the SecurityCritical attribute. It is so for using for internal uses and making it useless for remote usage.

1. **What Are The Properties That Have To Be Initialized For Creating A Silverlight Control Using Createsilverlight()?**

The properties ‘source’ and ‘elementID’ are to be initialized. The ‘source’ attribute can be a ‘.xaml’ file or an ‘.aspx’ file.

1. **Explain The Path Instructions In Xaml**

The <path> instruction of XAML allows to draw and fill a path. Various points in the path represents are represented by the Data attribute. The attribute includes M which means to move to command, moves to a coordinate and C represents the absolute path. The H represents line to command.   
  
The following is the code snippet to draw a path:   
<Path Data="M 200,40 C 50,90 200,250 200,75 H 480" Stroke="Black" StrokeThickness="4"/>

1. **Explain The Resemblance Between Css And Silverlight, Regarding Overlapping Objects.**

Silverlight content is hosted on the tag. CSS of DIV can be changed as it is for other HTML documents. The changes can be for absolute positioning, z-indexing, top and left properties etc.

1. **What Kind Of Brushed Does Silverlight Support?**

Silverlight brush objects supports for painting with solid colors, linear gradients, radical gradients and images.   
  
SolidColorBrush is used to paint a closed object such as rectangle   
  
LinearGradientBrush - used to paint a linear gradient like glass, water and other smooth surfaces.   
  
RadialGradientBrush – used to paint a radial gradient which is a blend together along an axis.

1. **Explain The Mouse Events That Silverlight Currently Supports.**

The mouse events that supports silverlight are   
  
LostMouseCapture - occurs when an UI element lost mouse capture  
MouseMove - occurs when the mouse position changes  
MouseEnter - occurs when the mouse pointer enters into the bounding area of an object  
MouseLeave - occurs when the mouse pointer leaves the bounding area of an object  
MouseLeftButtonDown - occurs when the left mouse button is down  
MouseLeftButtonU - occurs when the left mouse button is up followed by MouseLeftButtonDown

1. **Difference Between Mouseleftbuttondown And Mouseleftbuttonup In Silverlight.**

The difference is the action of the button. Once mouse button is pressed MouseLeftButtonDown event is invoked and MouseLeftButtonUp event is invoked after pressing left mouse button down and release it.

1. **What Is The Function Used For Removing An Event Listener?**

The method removeEventListener() is used for deleting an event listener. The parameters for this method are eventName and listener.

1. **How Would You Implement Drag-and-drop In Silverlight?**

Drag and drop in Silverlight can be implemented by using Drag Drop Manager by using DragSource and DropTarget controls. DragSource makes any control to be draggable and DropTarget defines a location in the application to drop the DragSources.

1. **What Are The Properties Of The Eventargs Argument When Capturing Keyboard Events?**

The properties of eventArgs are types of keys like shift, ctrl and mouse actions, mouse pointer coordinates, x coordinate value, y coordinate value.

1. **What Is The Function Used To Get A Reference To An Object Inside The Silverlight Control?**

The method findName() is used to refer an object inside the Silverlight control. The container reference is made in the Container.xaml and the corresponding Java Script file is called as Container.js. Now to access the object use the following function:

Container[1].Element.findName(“MyImageControl”).Source = “myPersonalImage.jpg”

1. **What Objects Support Tranformations? What Are The Transformations That Silverlight Supports For The Elements?**

The objects Ellipse and Rectangle are supported for transformations. The transformations that Silverlight supports for elements are rotations, scales, skews, and translations.

1. **Explain The Steps Needed To Be Performed In Order To Create An Animation In Xaml**

Animation is performed by using Storyboard.TargetName property. For example, to animate an object, the following are the steps:

- Write <BeginStoryboard> tag.  
- Embed <Storyboard> tag  
- Write <DoubleAnimationUsingKeyFrames> along with begin time and Storyboard.TargetName, Storyboard.TargetProperty attributes.   
- Use <SplineDoubleKeyFrame> tag along with KeyTime and value attributes   
- Close all the corresponding tags.

The following is the snippet of XAML file to animate an object  
<BeginStoryboard>  
        <Storyboard>  
            <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"   
                   Storyboard.TargetName= "border" Storyboard.TargetProperty="(Border.Background).  
                   (GradientBrush.GradientStops)[0].(GradientStop.Offset) ">  
             <SplineDoubleKeyFrame KeyTime="00:00:0" Value="0"/>  
            <SplineDoubleKeyFrame KeyTime="00:00:1" Value="1"/>  
           </DoubleAnimationUsingKeyFrames>  
         </Storyboard>  
</BeginStoryboard>

1. **What Are The Animation Types Supported By Silverlight?**

Silverlight supports 2 types of animations:

**From/To/By animation:**

Used for animating between a starting and ending value:

- From is used to set the starting value  
- To is used to set the ending value  
- By is used for setting ending value relative to the starting value of the animation

**Key-frame animation**

Key frame objects animates between a series of values. They are powerful than From/To/By animations, as there is flexibility to specify any number of target values and even interpolation method can also be controlled.

1. **Explain The Concept Of Keyframe. What Is The Difference Between Silverlight And Flash Regarding Animations?**

The value of the target property is animated by a key-frame animation. A transition among its target values over its duration is created. However the targeted values can by any number.   
  
The differences are:   
  
Flash:   
- Has no notion of animation other than matrices transformation.  
- Binary shape records are used to store the shapes.  
- Supports multiple video formats.  
- Works in IE, Firefox, Safari and Chrome  
  
Silverlight:  
- Supports the WPF animation model.  
- Uses XAML to output a simple XML object.  
- Implements industry standard VC-1 codec for video and supports for WMV and WMA.  
- Works after installing silverlight plug-in in IE  
- Works after plug-in with warnings about the execution, was installed and works  
- Not supported by Safari  
- The animated box will be shown but will not animate.

1. **How Many Classes Can A Single .net Dll Contain?**

It can contain many classes.

1. **How Could You Determine The Media Position When Playing A Video In Silverlight?**

The elements mePlayer.width and mePlayer.height retrieves the positions of the media in silver light. The source of the media player is set by mePlayer.Source , using the uri resource.

1. **Name Two Properties Common In Every Validation Control?**

ControlToValidate property and Text property.

1. **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.

1. **How Could You Modify Xaml Content From Javascript?**

Place xaml in Java Script using createFromXaml function. Use   
  
- object.SetFontSource(downloader) for adding the font files in the downloaded content.   
- object.SetSource(downloader, part) for setting the value of an Image object  
- object.SetValue(propertyName, value) for setting the value .

1. **What Are The Necessary Step That Need To Be Performed In Order To Download Content From Within Silverlight?**

A collection independent files that contain XAML content, media assets, and other application information can be downloaded from within Silverlight.   
  
To download the content, the System.net.WebClient class is used.   
- Request the content. The WebClient class empowers you to request the content.  
- Specify how to load the content.   
  
To download string element content use DownloadStringAsync method. This method starts downloading the process which runs asynchronously. Requesting Binary Content   
  
To download binary element content, use OpenReadAsync method. The contents includes compressed files, application modules and media files.

1. **What Asp.net Control Can Embed Xaml Into Asp.net Pages?**

The asp:Silverlight is used to embed XAML into ASP.Net pages.The following code snippet illustrates this:

<asp:Silverlight ID="Xaml" runat="server" Source="../ClientBin/sample.xap" Width="100%" Height="100%" />

1. **Does Silverlight Supports Database Connectivity? Explain**

Silverlight supports database connectivity. It is done through LINQ to SQL classes.   
  
- Add a new Web to the solution for hosting the control" radio button is checked. And click on OK.  
- Add a new LINQ to SQL class in App\_Code folder  
- Use server explorer and drag and drop the required table in DataClass  
- Add a new Silverlight-enabled WCF Service in Web Project.  
          - Write code on Service.cs to retrieve data from tables.  
          - Use DataClassesDataContetxt class to create objects.  
          - Use a variable to store the data from a table

1. **What Is The Codec Silverlight Supports For Hd Streaming?**

The following formats are supported for Video streaming:  
- WMV1: Windows Media Video 7   
- WMV2: Windows Media Video 8   
- WMV3: Windows Media Video 9   
- WMVA: Windows Media Video Advanced Profile, non-VC-1   
- WMVC1: Windows Media Video Advanced Profile, VC-1   
  
The following formats are supported for Audio streaming:  
- WMA 7: Windows Media Audio 7   
- WMA 8: Windows Media Audio 8   
- WMA 9: Windows Media Audio 9   
- MP3: ISO/MPEG Layer-3

1. **How Can Iis Improve Silverlight Streaming?**

IIS smooth streaming enables for delivering HD streams smoothly on any device. IIS smooth streaming is an extension of IIS7 Media Services 3.0. It enables adaptive live streaming through HTTProtocols.

1. **Whats An Assembly?**

Assemblies are the building blocks of the .NET framework. Overview of assemblies from MSDN

1. **What Is Smpte Vc-1?**

VC-1 is an industry-standard video format, recognized by the Society of Motion Picture and Television Engineers (SMPTE), and most notably ships in all HD-DVD and Blu-ray Disc?certified electronics, hardware, and tools. Windows Media Video 9 (WMV-9) is the Microsoft implementation of the SMPTE VC-1 standard video codec. Microsoft initiated development of the standard with the release of WMV-9 to SMPTE.

1. **What Is Xaml? Are Xaml File Compiled Or Built On Runtime?**

XAML stands for Extensible Application Markup Language. It is an XML based markup language. XAML is the language for visual presentation of the application development in MS-Expression Blend. Expression Blend application creation is writing XAML code in design view of Expression Blend.   
  
Usually XAML files are compiled and also support parsing during runtime. At runtime XAML file is not seen. When the XAML based project is build, a “g.cs” file will be created in obi\Debug folder.

1. **What Are Dependency Properties In Silverlight?**

Dependency properties are exposed as CLR properties. Dependency properties purpose is to provide a way of computing the value of a property that is based on the value of other inputs.

1. **Will Silverlight Support Digital Rights Management?**

For content providers, Silverlight will support digital rights management (DRM) built on the recently announced Microsoft PlayReady content access technology on Windows-based computers and Macintosh computers

1. **Explain The Relationship Between Silverlight And Windows Media Technologies?**

Silverlight is built on top of Windows Media enabling developers and designers to collaborate in building media experiences and RIAs. Silverlight is released by the Server and Tools Division at Microsoft as a part of the .NET Framework.

1. **What Is The Relationship Between Silverlight And Windows Media Player?**

The Silverlight browser plug-in is a separate component, independent of Windows Media Player. Silverlight is designed for delivery of cross-platform, cross-browser media experiences and rich interactive applications (RIAs) inside a Web browser combining audio, video, animation, overlays, and more. Windows Media Player delivers a breadth of local playback and user focused experiences, while also offering support for application and Web page embedding.

1. **Will Silverlight-based Applications And Content Run On Any Web Server? What Are The Benefits To Running It On Servers Running Windows?**

Silverlight works with any web server just like HTML. Video and audio content can also be downloaded and played back from any Web server platform.The main advantages of Windows server-based distribution of Silverlight applications include Windows Media Services with Fast Stream and Fast reconnect technologies, lower distribution costs and tap into the full Windows server ecosystem of platform components and partner solutions.

1. **What Features Are Missing From Silverlight Presentation Markup That Will Be Supported In Windows Presentation Foundation?**

High-end Windows specific features of WPF, such as real 3D, hardware-based video acceleration, and full document support, will not be supported in Silverlight. This is by design in order to serve Silverlight’s cross-browser, cross-platform reach scenario that demands a light weight plug-in.

1. **Is Silverlight Supported On Various Locales?**

Silverlight installs on localized versions of Macintosh computers and Windows. At this time, the installation is available in an international English format. Final releases will render international text (using double-byte characters) and support the full 64K Unicode character set. Silverlight uses simple input mechanism that treats all the languages in the same way.

1. **What Are The Different Ways To Display Text With Silverlight?**

Silverlight supports displaying static preformatted text that is comprised out of glyph elements and also dynamic text that uses TextBlock. With glyphs, one needs to position the characters individually while TextBlock supports simple layout.

1. **What Is Xaml ?**

Extensible Application Markup Language (XAML,pronounced zammel) is a declarative XML-based language created by Microsoft which is used to initialize structured values and objects.

1. **What Is The Difference Between Wpf And Silverlight?**

Silverlight uses a particular implementation of a XAML parser, with that parser being part of the Silverlight core install. In some cases, the parsing behavior differs from the parsing behavior in Windows Presentation Foundation (WPF), which also has a particular implementation.

1. **Can You Name Built-in Layout Panels You Have Been Using With Silverlight?**

You are looking for Canvas, StackPanel and Grid.

1. **How Can I Switch To Expression Blend From Visual Studio?**

Expression blend provide great extensibility for XAML files.To switch to Expression Blend, right-click on the XAML file and select Open in Expression Blend.

1. **How Can You Set Image Source Dynamically From C# Application To"test.png" File?**

Surprisingly it is not as straight forward as it might sound, but anyone who seriously worked with Silverlight should be easily answer it. One of the ways is: img.Source = new BitmapImage(new Uri("test.png", UriKind. Relative));

1. **How Can I Create Image Pieces/sub Image?**

In straight way you create a subimage from an existing image. Here you just clip an image, Clipping is just different from Cropping. In clipping, first you have to dictate which part of the images to draw and later you have to remove all but the desired part of the image. Silverlight does not support cropping.

1. **How Does Silverlight 2 Differ From Adobe Flash?**

As I am from .Net background so in my views you can get C# / Vb.net compiled code but in Flash there is only action script.

1. **Can You Elaborate How To Start A Silverlight Application With Visual Studio?**

In the following step(s) I am giving the ideas all about:  
1.Create a project:Here you just start your visual studio, Select your programming language [C#/VB], Choose Silverligh Template give the name and save it.  
2. Adding SIlverlight COntrols:One thing is happened here, controls cannot dragged onto the designer, you can draw/drag the controls on XAML page.  
(a) Naming to control:In this step just give the name to your silverligt control like for Button you can give name as:btnmy SilverligtButton  
(b)Adding event handlers to Silverlight controls:Here you can give the event handlers like for click etc.  
(c)Testing Silverlight applications in Visual Studio:Now just press F5 and test your application.

1. **What Is Silverlight Tool Kit?**

To create an application or game you need to design, code and give some extra feature to your output. To do the above, you need some controls, IDE etc.

Silverlight Tool kit is nothing but is a collection of Silverlight Tools, Components etc. It includes source code describing the all you need to develop an application.

1. **What Happened When I Press F5 Within Visual Studio To Run Silverlight Application?**

When you run the Silverlight application within Visual Studio, a new folder created in the web-site project for silverlight solution and it happened only first time. The folder name is ClientBin and having package with XAP extension which contains compiled project.

1. **What Is Storyboard?**

Storyboard is a Silverlight class with controls animations with a timeline, and provides object and property targeting information for its child animations.

1. **Can I Add More Than One .xaml Pages In Silverlight Application?**

Yes, you can have multiple .xaml files in a single project.In the App.xaml, in the method Application\_Startup you can choose, which page you want to initially display.

1. **What Is The Best Place To Start Silverlight Application?**

There is no hard and fast rule to start Silverlight application. Every developer can start as per his/her experience. like as per my case I always prefer Visual Studio. SO, in my view Visual Studio is the best place to start with Silverlight2 applications.Microsoft provides templates for creating Silverlight applications and libraries in C# and Visual Basic.

1. **Is Silverlight The Official Name For "wpf/e"?**

Yes. Silverlight was formerly code-named "WPF/E."

1. **Does Silverlight Web Application Work With All Browsers ?**

Yes,A web application developed by silverlight technology can work with any browser.

1. **Is Silverlight A New Media Player?**

No. Silverlight is a cross-browser, cross-platform plug-in for delivering media experiences and RIAs.It is not a desktop application or stand-alone media player.

1. **How Is My Content Secured From Unauthorized Access?**

You will have to be signed into the SilverlightTM Streaming service to manage your account and your Silverlight applications. Your SilverlightTM Streaming ID and secret key, associated to your Windows Live ID, will authenticate you as the unique and legitimate owner of the applications and content you upload to the service. You will also need this information to manage your Silverlight applications using the API. The SilverlightTM Streaming ID is public. However, the secret key should be kept confidential.

1. **Explain Form Level Validation And Field Level Validation**

Field-level validation provides immediate validation of the input given by the user. The events associated with field-level validation are KeyDown, KeyPress, textchange, etc.  
In form-level validation the validation step is done after the filling up of the form is done. It’s usually when the user submits the forms.

1. **Overview Of Ado.net Architecture.**

Data Provider provides objects through which functionalities like opening and closing connection, retrieving and updating data can be availed.   
  
It also provides access to data source like SQL Server, Access, and Oracle).   
  
Some of the data provider objects are:

* + Command object which is used to store procedures.
  + Data Adapter which is a bridge between datastore and dataset.
  + Datareader which reads data from data store in forward only mode.
  + A dataset object is not in directly connected to any data store. It represents disconnected and cached data. The dataset communicates with Data adapter that fills up the dataset. Dataset can have one or more Datatable and relations.
  + DataView object is used to sort and filter data in Datatable.

1. **Asp.net State Management**

This article describes state management in ASP.NET. It explains client-side state management and server-side state management.

1. **Asp.net Caching**

This includes caching mechanism in ASP.NET, its advantages and types.

1. **Define Xslt.**

XSLT is a language for transforming XML documents into XHTML documents or to other XML documents.   
  
XPath is a language for navigating in XML documents.

1. **What Is Xpath?**

XPath is a language that is used to navigate through XML documents.

1. **Define Xmlreader Class.**

The XMLReader Class (Assembly: System.Xml.dll) represents a reader that provides fast, non-cached, forward-only access to XML data.

1. **Define Xmlvalidatingreader Class.**

The XMLValidatingReader class (Assembly: System.Xml.dll) represents a reader that provides:  
- Document type definition (DTD),  
- XML-Data Reduced (XDR) schema, and   
- XML Schema definition language (XSD) validation

1. **How Can You Sort The Elements Of The Array In Descending Order?**

By calling Sort() and then Reverse() methods.

1. **What's The .net Collection Class That Allows An Element To Be Accessed Using A Unique Key?**

HashTable.

1. **Can You Prevent Your Class From Being Inherited By Another Class?**

Yes. The keyword “sealed” will prevent the class from being inherited.

1. **How Is Method Overriding Different From Method Overloading?**

When overriding a method, you change the behavior of the method for the derived class. Overloading a method simply involves having another method with the same name within the class.

1. **What Is A Satellite Assembly?**

When you write a multilingual or multi-cultural application in .NET, and want to distribute the core application separately from the localized modules, the localized assemblies that modify the core application are called satellite assemblies.

1. **What Are Advantages And Disadvantages Of Microsoft-provided Data Provider Classes In Ado.net?**

SQLServer.NET data provider is high-speed and robust, but requires SQL Server license purchased from Microsoft. OLE-DB.NET is universal for accessing other sources, like Oracle, DB2, Microsoft Access and Informix. OLE-DB.NET is a .NET layer on top of the OLE layer, so it’s not as fastest and efficient as SqlServer.NET.

1. **What Is .net Framework?**

The .NET Framework has two main components: the common language runtime and the .NET Framework class library.

You can think of the runtime as an agent that manages code at execution time,providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness.

The class library, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

1. **Is .net A Runtime Service Or A Development Platform?**

It's both and actually a lot more. Microsoft .NET includes a new way of delivering software and services to businesses and consumers. A part of Microsoft.NET is the .NET Frameworks. The .NET frameworks SDK consists of two parts: the .NET common language runtime and the .NET class library. In addition, the SDK also includes command-line compilers for C#, C++, JScript, and VB. You use these compilers to build applications and components. These components require the runtime to execute so this is a development platform.

1. **What Are The New Features Of Framework 1.1 ?**

1. Native Support for Developing Mobile Web Applications

2. Enable Execution of Windows Forms Assemblies Originating from the Internet Assemblies originating from the Internet zone for example, Microsoft Windows Forms controls embedded in an Internet-based Web page or Windows Forms assemblies hosted on an Internet Web server and loaded either through the Web browser or programmatically using the System.Reflection.Assembly.LoadFrom() method now receive sufficient permission to execute in a semi-trusted manner. Default security policy has been changed so that assemblies assigned by the common language runtime (CLR) to the Internet zone code group now receive the constrained permissions associated with the Internet permission set. In the .NET Framework 1.0 Service Pack 1 and Service Pack 2, such applications received the permissions associated with the Nothing permission set and could not execute.

3. Enable Code Access Security for ASP.NET Applications Systems administrators can now use code access security to further lock down the permissions granted to ASP.NET Web applications and Web services. Although the operating system account under which an application runs imposes security restrictions on the application, the code access security system of the CLR can enforce additional restrictions on selected application resources based on policies specified by systems administrators. You can use this feature in a shared server environment (such as an Internet service provider (ISP) hosting multiple Web applications on one server) to isolate separate applications from one another, as well as with stand-alone servers where you want applications to run with the minimum necessary privileges.

4. Native Support for Communicating with ODBC and Oracle Databases

5. Unified Programming Model for Smart Client Application Development The Microsoft .NET Compact Framework brings the CLR, Windows Forms controls, and other .NET Framework features to small devices. The .NET Compact Framework supports a large subset of the .NET Framework class library optimized for small devices.

6. Support for IPv6  
The .NET Framework 1.1 supports the emerging update to the Internet Protocol, commonly referred to as IP version 6, or simply IPv6. This protocol is designed to significantly increase the address space used to identify communication endpoints in the Internet to accommodate its ongoing growth.

1. **How Do I Define My Own Code Group?**

Use caspol. For example, suppose you trust code from www.mydomain.com and you want it have full access to your system, but you want to keep the default restrictions for all other internet sites. To achieve this, you would add a new code group as a subgroup of the 'Zone - Internet' group, like this:  
 Now if you run caspol -lg you will see that the new group has been added as group

1. **How Do I Change The Permission Set For A Code Group?**

Use caspol. If you are the machine administrator, you can operate at the 'machine' level - which means not only that the changes you make become the default for the machine, but also that users cannot change the permissions to be more permissive. If you are a normal (non-admin) user you can still modify the permissions, but only to make them more restrictive. For example, to allow intranet code to do what it likes you might do this:

Note that because this is more permissive than the default policy (on a standard system), you should only do this at the machine level - doing it at the user level will have no effect.

1. **Can I Create My Own Permission Set?**

Yes. Use caspol -ap, specifying an XML file containing the permissions in the permission set. To save you some time, here is a sample file corresponding to the 'Everything' permission set - just edit to suit your needs. When you have edited the sample, add it to the range of available permission sets like this:

caspol -ap samplepermset.xml

Then, to apply the permission set to a code group, do something like this:

caspol -cg 1.3 SamplePermSet (By default, 1.3 is the 'Internet' code group)

1. **I'm Having Some Trouble With Cas. How Can I Diagnose My Problem?**

Caspol has a couple of options that might help. First, you can ask caspol to tell you what code group an assembly belongs to, using caspol -rsg. Similarly, you can ask what permissions are being applied to a particular assembly using caspol -rsp.

1. **I Can't Be Bothered With All This Cas Stuff. Can I Turn It Off?**

Yes, as long as you are an administrator. Just run:  
caspol -s off

1. **What Is Msil, Il?**

When compiling to managed code, the compiler translates your source code into Microsoft intermediate language (MSIL), which is a CPU-independent set of instructions that can be efficiently converted to native code. MSIL includes instructions for loading, storing, initializing, and calling methods on objects, as well as instructions for arithmetic and logical operations, control flow, direct memory access, exception handling, and other operations. Microsoft intermediate language (MSIL) is a language used as the output of a number of compilers and as the input to a just-in-time (JIT) compiler. The common language runtime includes a JIT compiler for converting MSIL to native code.

1. **Can I Write Il Programs Directly?**

Yes. Peter Drayton posted this simple example to the DOTNET mailing list:  
.assembly MyAssembly {}  
.class MyApp {  
.method static void Main() {  
.entrypoint  
ldstr "Hello, IL!"  
call void System.Console::WriteLine(class System.Object)  
ret  
}  
}  
Just put this into a file called hello.il, and then run ilasm hello.il. An exe assembly will be generated.

1. **What Is Jit (just In Time)? How It Works?**

Before Microsoft intermediate language (MSIL) can be executed, it must be converted by a .NET Framework just-in-time (JIT) compiler to native code, which is CPU-specific code that runs on the same computer architecture as the JIT compiler.

Rather than using time and memory to convert all the MSIL in a portable executable (PE) file to native code, it converts the MSIL as it is needed during execution and stores the resulting native code so that it is accessible for subsequent calls.

The runtime supplies another mode of compilation called install-time code generation. The install-time code generation mode converts MSIL to native code just as the regular JIT compiler does, but it converts larger units of code at a time, storing the resulting native code for use when the assembly is subsequently loaded and executed.

As part of compiling MSIL to native code, code must pass a verification process unless an administrator has established a security policy that allows code to bypass verification. Verification examines MSIL and metadata to find out whether the code can be determined to be type safe, which means that it is known to access only the memory locations it is authorized to access.

1. **Which Namespace Is The Base Class For .net Class Library?**

system.object

1. **What Is Event - Delegate? Clear Syntax For Writing A Event Delegate**

The event keyword lets you specify a delegate that will be called upon the occurrence of some "event" in your code. The delegate can have one or more associated methods that will be called when your code indicates that the event has occurred. An event in one program can be made available to other programs that target the .NET Framework Common Language Runtime.

// keyword\_delegate.cs  
// delegate declaration  
delegate void MyDelegate(int i);  
class Program  
{  
public static void Main()  
{  
TakesADelegate(new MyDelegate(DelegateFunction));  
}  
public static void TakesADelegate(MyDelegate SomeFunction)  
{  
SomeFunction(21);  
}  
public static void DelegateFunction(int i)  
{  
System.Console.WriteLine("Called by delegate with number: {0}.", i);  
}  
}

1. **What Are Object Pooling And Connection Pooling And Difference? Where Do We Set The Min And Max Pool Size For Connection Pooling?**

Object pooling is a COM+ service that enables you to reduce the overhead of creating each object from scratch. When an object is activated, it is pulled from the pool. When the object is deactivated, it is placed back into the pool to await the next request. You can configure object pooling by applying the ObjectPoolingAttribute attribute to a class that derives from the System.EnterpriseServices.ServicedComponent class.

Object pooling lets you control the number of connections you use, as opposed to connection pooling, where you control the maximum number reached. Following are important differences between object pooling and connection pooling:

Creation. When using connection pooling, creation is on the same thread, so if there is nothing in the pool, a connection is created on your behalf. With object pooling, the pool might decide to create a new object. However, if you have already reached your maximum, it instead gives you the next available object. This is crucial behavior when it takes a long time to create an object, but you do not use it for very long.

Enforcement of minimums and maximums. This is not done in connection pooling. The maximum value in object pooling is very important when trying to scale your application. You might need to multiplex thousands of requests to just a few objects. (TPC/C benchmarks rely on this.)

COM+ object pooling is identical to what is used in .NET Framework managed SQL Client connection pooling. For example, creation is on a different thread and minimums and maximums are enforced.

1. **Interop Services?**

The common language runtime provides two mechanisms for interoperating with unmanaged code:

Platform invoke, which enables managed code to call functions exported from an unmanaged library.  
COM interop, which enables managed code to interact with COM objects through interfaces.

Both platform invoke and COM interop use interop marshaling to accurately move method arguments between caller and callee and back, if required.

1. **What Are Server Controls?**

ASP.NET server controls are components that run on the server and encapsulate user-interface and other related functionality. They are used in ASP.NET pages and in ASP.NET code-behind classes.

1. **What Is Exception Handling?**

When an exception occurs, the system searches for the nearest catch clause that can handle the exception, as determined by the run-time type of the exception. First, the current method is searched for a lexically enclosing try statement, and the associated catch clauses of the try statement are considered in order. If that fails, the method that called the current method is searched for a lexically enclosing try statement that encloses the point of the call to the current method. This search continues until a catch clause is found that can handle the current exception, by naming an exception class that is of the same class, or a base class, of the run-time type of the exception being thrown. A catch clause that doesn't name an exception class can handle any exception.

Once a matching catch clause is found, the system prepares to transfer control to the first statement of the catch clause. Before execution of the catch clause begins, the system first executes, in order, any finally clauses that were associated with try statements more nested that than the one that caught the exception.

Exceptions that occur during destructor execution are worth special mention. If an exception occurs during destructor execution, and that exception is not caught, then the execution of that destructor is terminated and the destructor of the base class (if any) is called. If there is no base class (as in the case of the object type) or if there is no base class destructor, then the exception is discarded.

1. **What Are The Different Types Of Assemblies?**

Private, Public/Shared, Satellite

1. **What Are Satellite Assemblies? How You Will Create This? How Will You Get The Different Language Strings?**

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.

(For example, English and Japanese editions of the .NET Framework version 1.1 share the same core files. The Japanese .NET Framework version 1.1 adds satellite assemblies with localized resources in a \ja subdirectory. An application that supports the .NET Framework version 1.1, regardless of its language, always uses the same core runtime files.)

1. **What Is Jagged Arrays?**

A jagged array is an array whose elements are arrays. The elements of a jagged array can be of different dimensions and sizes. A jagged array is sometimes called an "array-of-arrays."

1. **What Is Assembly Manifest? What All Details The Assembly Manifest Will Contain?**

Every assembly, whether static or dynamic, contains a collection of data that describes how the elements in the assembly relate to each other. The assembly manifest contains this assembly metadata. An assembly manifest contains all the metadata needed to specify the assembly's version requirements and security identity, and all metadata needed to define the scope of the assembly and resolve references to resources and classes. The assembly manifest can be stored in either a PE file (an .exe or .dll) with Microsoft intermediate language (MSIL) code or in a standalone PE file that contains only assembly manifest information.

It contains Assembly name, Version number, Culture, Strong name information, List of all files in the assembly, Type reference information, Information on referenced assemblies.

1. **Difference Between Assembly Manifest & Metadata?**

assembly self-describing. The assembly manifest contains the assembly's metadata. The manifest establishes the assembly identity, specifies the files that make up the assembly implementation, specifies the types and resources that make up the assembly, itemizes the compile-time dependencies on other assemblies, and specifies the set of permissions required for the assembly to run properly. This information is used at run time to resolve references, enforce version binding policy, and validate the integrity of loaded assemblies. The self-describing nature of assemblies also helps makes zero-impact install and XCOPY deployment feasible.

metadata - Information that describes every element managed by the common language runtime: an assembly, loadable file, type, method, and so on. This can include information required for debugging and garbage collection, as well as security attributes, marshaling data, extended class and member definitions, version binding, and other information required by the runtime.

1. **What Is Global Assembly Cache (gac) And What Is The Purpose Of It? (how To Make An Assembly To Public? Steps) How More Than One Version Of An Assembly Can Keep In Same Place?**

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.  
Steps  
- Create a strong name using sn.exe tool  
eg: sn -k keyPair.snk  
- with in AssemblyInfo.cs add the generated file name  
eg: [assembly: AssemblyKeyFile("abc.snk")]  
- recompile project, then install it to GAC by either  
drag & drop it to assembly folder (C:\WINDOWS\assembly OR  
C:\WINNT\assembly) (shfusion.dll tool)  
or  
gacutil -i abc.dll

1. **How Do I Know When My Thread Pool Work Item Has Completed?**

There is no way to query the thread pool for this information. You must put code into the WaitCallback method to signal that it has completed. Events are useful for this.

1. **How To Find Methods Of A Assembly File (not Using Ildasm)**

Reflection

1. **What Is Garbage Collection In .net? Garbage Collection Process?**

The process of transitively tracing through all pointers to actively used objects in order to locate all objects that can be referenced, and then arranging to reuse any heap memory that was not found during this trace. The common language runtime garbage collector also compacts the memory that is in use to reduce the working space needed for the heap.

1. **Readonly Vs. Const?**

A const field can only be initialized at the declaration of the field. A readonly field can be initialized either at the declaration or in a constructor.Therefore,readonly fields can have different values depending on the constructor used. Also, while a const field is a compile-time constant, the readonly field can be used for runtime constants, as in the following example:

public static readonly uint l1 = (uint) DateTime.Now.Ticks;

1. **What Is Reflection In .net? Namespace? How Will You Load An Assembly Which Is Not Referenced By Current Assembly?**

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. Using reflection to access .NET metadata is very similar to using ITypeLib/ITypeInfo to access type library data in COM, and it is used for similar purposes - e.g. determining data type sizes for marshaling data across context/process/machine boundaries.

Reflection can also be used to dynamically invoke methods (see System.Type.InvokeMember), or even create types dynamically at run-time (see System.Reflection.Emit.TypeBuilder).

1. **What Is Custom Attribute? How To Create? If I'm Having Custom Attribute In An Assembly, How To Say That Name In The Code?**

The primary steps to properly design custom attribute classes are as follows:  
Applying the AttributeUsageAttribute  
([AttributeUsage(AttributeTargets.All,  
Inherited = false, AllowMultiple = true)])  
Declaring the attribute. (class public class MyAttribute : System.Attribute { // .  
. . })  
Declaring constructors (public MyAttribute(bool myvalue) { this.myvalue =  
myvalue; })  
Declaring properties  
public bool MyProperty  
{  
get {return this.myvalue;}  
set {this.myvalue = value;}  
}

The following example demonstrates the basic way of using reflection to get access to custom attributes.  
class MainClass  
{  
public static void Main()  
{  
System.Reflection.MemberInfo info = typeof(MyClass);  
object[] attributes = info.GetCustomAttributes();  
for (int i = 0; i < attributes.Length; i ++)  
{  
System.Console.WriteLine(attributes[i]);  
}  
}  
}

1. **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.

1. **How Do You Create Threading In .net? What Is The Namespace For That?**

\*\*  
System.Threading.Thread

1. **Using Directive Vs Using Statement**

You create an instance in a using statement to ensure that Dispose is called on the object when the using statement is exited. A using statement can be exited either when the end of the using statement is reached or if, for example, an exception is thrown and control leaves the statement block before the end of the statement.

The using directive has two uses:

Create an alias for a namespace (a using alias).

Permit the use of types in a namespace, such that, you do not have to qualify the use of a type in that namespace (a using directive).

1. **Describe The Managed Execution Process?**

The managed execution process includes the following steps:  
Choosing a compiler.

To obtain the benefits provided by the common language runtime, you must use one or more language compilers that target the runtime.

Compiling your code to Microsoft intermediate language (MSIL).

Compiling translates your source code into MSIL and generates the required metadata.  
Compiling MSIL to native code.

At execution time, a just-in-time (JIT) compiler translates the MSIL into native code. During this compilation, code must pass a verification process that examines the MSIL and metadata to find out whether the code can be determined to be type safe.

Executing your code.

The common language runtime provides the infrastructure that enables execution to take place as well as a variety of services that can be used during execution.

1. **What Is Active Directory? What Is The Namespace Used To Access The Microsoft Active Directories? What Are Adsi Directories?**

Active Directory Service Interfaces (ADSI) is a programmatic interface for Microsoft Windows Active Directory. It enables your applications to interact with diverse directories on a network, using a single interface. Visual Studio .NET and the .NET Framework make it easy to add ADSI functionality with the DirectoryEntry and DirectorySearcher components. Using ADSI, you can create applications that perform common administrative tasks, such as backing up databases, accessing printers, and administering user accounts. ADSI makes it possible for you to:

Log on once to work with diverse directories. The Directory Entry component class provides username and password properties that can be entered at runtime and communicated to the Active Directory object you are binding to. Use a single application programming interface (API) to perform tasks on multiple directory systems by offering the user a variety of protocols to use. The DirectoryServices namespace provides the classes to perform most administrative functions.

Perform "rich querying" on directory systems. ADSI technology allows for searching for an object by specifying two query dialects: SQL and LDAP. Access and use a single, hierarchical structure for administering and maintaining diverse and complicated network configurations by accessing an Active Directory tree.

Integrate directory information with databases such as SQL Server. The DirectoryEntry path may be used as an ADO.NET connection string provided that it is using the LDAP provider using System.DirectoryServices;

1. **How Garbage Collector (gc) Works?**

The methods in this class influence when an object is garbage collected and when resources allocated by an object are released. Properties in this class provide information about the total amount of memory available in the system and the age category, or generation, of memory allocated to an object. Periodically, the garbage collector performs garbage collection to reclaim memory allocated to objects for which there are no valid references. Garbage collection happens automatically when a request for memory cannot be satisfied using available free memory. Alternatively, an application can force garbage collection using the Collect method.

Garbage collection consists of the following steps:  
The garbage collector searches for managed objects that are referenced in managed code.  
The garbage collector attempts to finalize objects that are not referenced.  
The garbage collector frees objects that are not referenced and reclaims their memory.

1. **Why Do We Need To Call Cg.supressfinalize?**

Requests that the system not call the finalizer method for the specified object.  
public static void SuppressFinalize( object obj );

The method removes obj from the set of objects that require finalization.  
The obj parameter is required to be the caller of this method.

Objects that implement the IDisposable interface can call this method from the IDisposable.Dispose method to prevent the garbage collector from calling Object.Finalize on an object that does not require it.

1. **What Is Nmake Tool?**

The Nmake tool (Nmake.exe) is a 32-bit tool that you use to build projects based on commands contained in a .mak file.  
usage : nmake -a all

1. **What Are Namespaces?**

The namespace keyword is used to declare a scope. This namespace scope lets you organize code and gives you a way to create globally-unique types. Even if you do not explicitly declare one, a default namespace is created. This unnamed namespace, sometimes called the global namespace,is present in every file.Any identifier in the global namespace is available for use in a named namespace. Namespaces implicitly have public access and this is not modifiable.

1. **What Is Rcw (runtime Callable Wrappers)?**

The common language runtime exposes COM objects through a proxy called the runtime callable wrapper (RCW). Although the RCW appears to be an ordinary object to .NET clients, its primary function is to marshal calls between a .NET client and a COM object.

1. **What Is Ccw (com Callable Wrapper)**

A proxy object generated by the common language runtime so that existing COM applications can use managed classes, including .NET Framework classes, transparently.

1. **How Will You Register Com+ Services?**

The .NET Framework SDK provides the .NET Framework Services Installation Tool (Regsvcs.exe - a command-line tool) to manually register an assembly containing serviced components. You can also access these registration features programmatically with the System.EnterpriseServices RegistrationHelper class by creating an instance of class RegistrationHelper and using the method InstallAssembly.

1. **What Is Use Of Contextutil Class?**

ContextUtil is the preferred class to use for obtaining COM+ context information.

1. **What Is Pinvoke?**

Platform invoke is a service that enables managed code to call unmanaged functions implemented in dynamic-link libraries (DLLs), such as those in the Win32 API. It locates and invokes an exported function and marshals its arguments (integers, strings, arrays, structures, and so on) across the interoperation boundary as needed.

1. **Is It True That Com Objects No Longer Need To Be Registered On The Server?**

Yes and No. Legacy COM objects still need to be registered on the server before they can be used. COM developed using the new .NET Framework will not need to be registered. Developers will be able to autoregister these objects just by placing them in the 'bin' folder of the application.

1. **Can .net Framework Components Use The Features Of Component Services?**

Yes, you can use the features and functions of Component Services from a .NET Framework component.

1. **What Are The Oops Concepts?**

1) Encapsulation: It is the mechanism that binds together code and data in manipulates, and keeps both safe from outside interference and misuse. In short it isolates a particular code and data from all other codes and data. A well-defined interface controls the access to that particular code and data.

2) Inheritance: It is the process by which one object acquires the properties of another object. This supports the hierarchical classification. Without the use of hierarchies, each object would need to define all its characteristics explicitly. However, by use of inheritance, an object need only define those qualities that make it unique within its class. It can inherit its general attributes from its parent. A new sub-class inherits all of the attributes of all of its ancestors.

3) Polymorphism: It is a feature that allows one interface to be used for general class of actions. The specific action is determined by the exact nature of the situation. In general polymorphism means "one interface, multiple methods", This means that it is possible to design a generic interface to a group of related activities. This helps reduce complexity by allowing the same interface to be used to specify a general class of action. It is the compiler's job to select the specific action (that is, method) as it applies to each situation.

1. **What Is The Difference Between A Struct And A Class?**

The struct type is suitable for representing lightweight objects such as Point, Rectangle, and Color. Although it is possible to represent a point as a class, a struct is more efficient in some scenarios. For example, if you declare an array of 1000 Point objects, you will allocate additional memory for referencing each object. In this case, the struct is less expensive.

When you create a struct object using the new operator, it gets created and the appropriate constructor is called. Unlike classes, structs can be instantiated without using the new operator. If you do not use new, the fields will remain unassigned and the object cannot be used until all of the fields are initialized. It is an error to declare a default (parameterless) constructor for a struct. A default constructor is always provided to initialize the struct members to their default values.

It is an error to initialize an instance field in a struct.

There is no inheritance for structs as there is for classes. A struct cannot inherit from another struct or class, and it cannot be the base of a class. Structs, however, inherit from the base class Object. A struct can implement interfaces, and it does that exactly as classes do.

A struct is a value type, while a class is a reference type.

1. **What Is Method Overloading?**

Method overloading occurs when a class contains two methods with the same name, but different signatures.

1. **What Is Method Overriding? How To Override A Function In C#?**

Use the override modifier to modify a method, a property, an indexer, or an event. An override method provides a new implementation of a member inherited from a base class. The method overridden by an override declaration is known as the overridden base method. The overridden base method must have the same signature as the override method.

You cannot override a non-virtual or static method. The overridden base method must be virtual, abstract, or override.

1. **Can We Call A Base Class Method Without Creating Instance?**

Its possible If its a static method.  
Its possible by inheriting from that class also.  
Its possible from derived classes using base keyword.

1. **You Have One Base Class Virtual Function How Will Call That Function From Derived Class?**

class a  
{  
public virtual int m()  
{  
return 1;  
}  
}  
class b:a  
{  
public int j()  
{  
return m();  
}  
}

1. **In Which Cases You Use Override And New Base?**

Use the new modifier to explicitly hide a member inherited from a base class. To hide an inherited member, declare it in the derived class using the same name, and modify it with the new modifier.

1. **What Are Sealed Classes In C#?**

The sealed modifier is used to prevent derivation from a class. A compile-time error occurs if a sealed class is specified as the base class of another class. (A sealed class cannot also be an abstract class)

1. **What Is Polymorphism? How Does Vb.net/c# Achieve Polymorphism?**

class Token  
{  
public string Display()  
{  
//Implementation goes here  
return "base";  
}  
}  
class IdentifierToken:Token  
{  
public new string Display() //What is the use of new keyword  
{  
//Implementation goes here  
return "derive";  
}  
}  
static void Method(Token t)  
{  
Console.Write(t.Display());  
}  
public static void Main()  
{  
IdentifierToken Variable=new IdentifierToken();  
Method(Variable); //Which Class Method is called here  
Console.ReadLine();  
}  
For the above code What is the "new" keyword and Which Class Method is called here  
A: it will call base class Display method  
class Token  
{  
public virtual string Display()  
{  
//Implementation goes here  
return "base";  
}  
}  
class IdentifierToken:Token  
{  
public override string Display() //What is the use of new keyword  
{  
//Implementation goes here  
return "derive";  
}  
}  
static void Method(Token t)  
{  
Console.Write(t.Display());  
}  
public static void Main()  
{  
IdentifierToken Variable=new IdentifierToken();  
Method(Variable); //Which Class Method is called here  
Console.ReadLine();  
}  
A: Derive

1. **Explain About Protected And Protected Internal, "internal" Accessspecifier?**

protected - Access is limited to the containing class or types derived from the containing class.  
internal - Access is limited to the current assembly.  
protected internal - Access is limited to the current assembly or types derived from the containing class.

1. **Difference Between Type Constructor And Instance Constructor? What Is Static Constructor, When It Will Be Fired? And What Is Its Use?**

(Class constructor method is also known as type constructor or type initializer) Instance constructor is executed when a new instance of type is created and the class constructor is executed after the type is loaded and before any one of the type members is accessed. (It will get executed only 1st time, when we call any static methods/fields in the same class.) Class constructors are used for static field initialization. Only one class constructor per type is permitted, and it cannot use the vararg (variable argument) calling convention.

A static constructor is used to initialize a class. It is called automatically to initialize the class before the first instance is created or any static members are referenced.

1. **What Is Private Constructor? And It's Use? Can You Create Instance Of A Class Which Has Private Constructor?**

When a class declares only private instance constructors, it is not possible for classes outside the program to derive from the class or to directly create instances of it. (Except Nested classes) Make a constructor private if:

- You want it to be available only to the class itself. For example, you might have a special constructor used only in the implementation of your class' Clone method.  
- You do not want instances of your component to be created. For example, you may have a class containing nothing but Shared utility functions, and no instance data. Creating instances of the class would waste memory.

1. **I Have 3 Overloaded Constructors In My Class. In Order To Avoid Making Instance Of The Class Do I Need To Make All Constructors To Private?**

Yes

1. **Overloaded Constructor Will Call Default Constructor Internally?**

NO

1. **What Is The Difference Between Finalize And Dispose (garbage Collection)**

Class instances often encapsulate control over resources that are not managed by the runtime, such as window handles (HWND), database connections, and so on. Therefore, you should provide both an explicit and an implicit way to free those resources. Provide implicit control by implementing the protected Finalize Method on an object (destructor syntax in C# and the Managed Extensions for C++). The garbage collector calls this method at some point after there are no longer any valid references to the object.

In some cases, you might want to provide programmers using an object with the ability to explicitly release these external resources before the garbage collector frees the object. If an external resource is scarce or expensive, better performance can be achieved if the programmer explicitly releases resources when they are no longer being used. To provide explicit control, implement the Dispose method provided by the IDisposable Interface. The consumer of the object should call this method when it is done using the object. Dispose can be called even if other references to the object are alive.

1. **Is Goto Statement Supported In C#? How About Java?**

Gotos are supported in C# to the fullest. In Java goto is a reserved keyword that provides absolutely no functionality.

1. **What's Different About Switch Statements In C#?**

No fall-throughs allowed. Unlike the C++ switch statement, C# does not support an explicit fall through from one case label to another. If you want, you can use goto a switch-case, or goto default.  
case 1:  
cost += 25;  
break;  
case 2:  
cost += 25;  
goto case 1;

1. **Advantage Of Ado.net?**
   * ADO.NET Does Not Depend On Continuously Live Connections
   * Database Interactions Are Performed Using Data Commands
   * Data Can Be Cached in Datasets
   * Datasets Are Independent of Data Sources
   * Data Is Persisted as XML
   * Schemas Define Data Structures
2. **How Would You Connect To Database Using .net?**

SqlConnection nwindConn = new SqlConnection("Data Source=localhost;  
Integrated Security=SSPI;" +"Initial Catalog=northwind");  
nwindConn.Open();

1. **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.

1. **What Are The Different Namespaces Used In The Project To Connect The Database? What Data Providers Available In .net To Connect To Database?**

System.Data.OleDb – classes that make up the .NET Framework Data Provider for OLE DB-compatible data sources. These classes allow you to connect to an OLE DB data source, execute commands against the source, and read the results.

System.Data.SqlClient – classes that make up the .NET Framework Data Provider for SQL Server, which allows you to connect to SQL Server 7.0, execute commands, and read results. The System.Data. SqlClient namespace is similar to the System.Data.OleDb namespace, but is optimized for access to SQL Server 7.0 and later.

System.Data.Odbc - classes that make up the .NET Framework Data Provider for ODBC. These classes allow you to access ODBC data source in the managed space.

System.Data.OracleClient - classes that make up the .NET Framework Data Provider for Oracle. These classes allow you to access an Oracle data source in the managed space.

1. **Which Method Do You Invoke On The Dataadapter Control To Load Your Generated Dataset With Data?**

Fill()

1. **Explain Different Methods And Properties Of Datareader Which You Have Used In Your Project?**
   * Read
   * GetString
   * GetInt32
   * while (myReader.Read())
   * Console.WriteLine("\t{0}\t{1}", myReader.GetInt32(0),
   * myReader.GetString(1));
   * myReader.Close();
2. **What Happens When We Issue Dataset.readxml Command?**

Reads XML schema and data into the DataSet.

1. **In How Many Ways We Can Retrieve Table Records Count? How To Find The Count Of Records In A Dataset?**

foreach(DataTable thisTable in myDataSet.Tables){

// For each row, print the values of each column.

foreach(DataRow myRow in thisTable.Rows){

1. **How To Check If A Datareader Is Closed Or Opened?**

IsClosed()

1. **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.

1. **What Is Method To Get Xml And Schema From Dataset?**

getXML () and getSchema ()

1. **Difference Between Application Events And Session Events**

The ASP.NET page framework provides ways for you to work with events that can be raised when your application starts or stops or when an individual user's session starts or stops:

**Application Events** are raised for all requests to an application. For example, Application\_BeginRequest is raised when any Web Forms page or XML Web service in your application is requested. This event allows you to initialize resources that will be used for each request to the application. A corresponding event, Application\_ EndRequest, provides you with an opportunity to close or otherwise dispose of resources used for the request.

**Session Events** are similar to application events (there is a Session\_OnStart and a Session\_OnEnd event), but are raised with each unique session within the application. A session begins when a user requests a page for the first time from your application and ends either when your application explicitly closes the session or when the session times out.

1. **Difference Between Asp Session And Asp.net Session?**

In Asp, the session is Process dependent, whereas in  
Asp.Net the session is Process independent.

1. **What Is Cookie Less Session? How It Works?**

By default, ASP.NET will store the session state in the same process that processes the request, just as ASP does. If cookies are not available, a session can be tracked by adding a session identifier to the URL. This can be enabled by setting the following:

sessionstate cookieless="true"

1. **What Method Do You Use To Explicitly Kill A Users Session?**

Abandon()

1. **What Are The Different Ways You Would Consider Sending Data Across Pages In Asp (i.e Between 1.asp To 2.asp)?**
   * Session
   * public properties
2. **What Is State Management In .net And How Many Ways Are There To Maintain A State In .net? What Is View State?**

Web pages are recreated each time the page is posted to the server. In traditional Web programming, this would ordinarily mean that all information associated with the page and the controls on the page would be lost with each round trip.

To overcome this inherent limitation of traditional Web programming, the ASP.NET page framework includes various options to help you preserve changes — that is, for managing state. The page framework includes a facility called view state that automatically preserves property values of the page and all the controls on it between round trips.

However, you will probably also have application-specific values that you want to preserve. To do so, you can use one of the state management options.

Client-Based State Management Options:

* + View State
  + Hidden Form Fields
  + Cookies
  + Query Strings

Server-Based State Management Options

* + Application State
  + Session State
  + Database Support

1. **What Are The Disadvantages Of View State / What Are The Benefits?**

Automatic view-state management is a feature of server controls that enables them to repopulate their property values on a round trip (without you having to write any code). This feature does impact performance, however, since a server control's view state is passed to and from the server in a hidden form field. You should be aware of when view state helps you and when it hinders your page's performance.

1. **When Maintaining Session Through Sql Server, What Is The Impact Of Read And Write Operation On Session Objects?**

Maintaining state using database technology is a common practice when storing user-specific information where the information store is large. Database storage is particularly useful for maintaining long-term state or state that must be preserved even if the server must be restarted.

1. **Explain The Differences Between Server-side And Client-side Code?**

Server side code will process at server side and it will send the result to client. Client side code (javascript) will execute only at client side.

1. **Which Asp.net Configuration Options Are Supported In The Asp.net Implementation On The Shared Web Hosting Platform?**

Many of the ASP.NET configuration options are not configurable at the site, application or subdirectory level on the shared hosting platform. Certain options can affect the security, performance and stability of the server and, therefore cannot be changed. The following settings are the only ones that can be changed in your site’s web.config file (s):

* + browserCaps
  + clientTarget
  + pages
  + customErrors
  + globalization
  + authorization
  + authentication
  + webControls
  + webServices

1. **What Is Role-based Security?**

A role is a named set of principals that have the same privileges with respect to security (such as a teller or a manager). A principal can be a member of one or more roles. Therefore, applications can use role membership to determine whether a principal is authorized to perform a requested action.

1. **How Do You Specify Whether Your Data Should Be Passed As Query String And Forms (mainly About Post And Get)**

Through Attribute tag of Form tag.

1. **Which Two Properties Are There On Every Validation Control?**

ControlToValidate, ErrorMessage

1. **How Do You Use Css In Asp.net?**

Within the section of an HTML document that will use these styles, add a link to this external CSS style sheet that follows this form:

<link rel="STYLESHEET" type="text/css" href="MyStyles.css" />

MyStyles.css is the name of your external CSS style sheet.

1. **How Do You Implement Postback With A Text Box?**

Make AutoPostBack property to true

1. **What Is Sql Injection?**

An SQL injection attack "injects" or manipulates SQL code by adding unexpected SQL to a query. Many web pages take parameters from web user, and make SQL query to the database. Take for instance when a user login, web page that user name and password and make SQL query to the database to check if a user has valid name and password.

Username: ' or 1=1 ---  
Password: [Empty]

This would execute the following query against the users table:

select count(\*) from users where userName='' or 1=1 --' and userPass=''

1. **Asp.net - How To Find Last Error Which Occurred?**

Server.GetLastError();  
[C#]  
Exception LastError;  
String ErrMessage;  
LastError = Server.GetLastError();  
if (LastError != null)  
ErrMessage = LastError.Message;  
else  
ErrMessage = "No Errors";  
Response.Write("Last Error = " + ErrMessage);

1. **What Is The Use Of sessionstate Tag In The Web.config File?**

Configuring session state: Session state features can be configured via the <sessionstate> section in a web.config file. To double the default timeout of 20 minutes, you can add the following to the web.config file of an application:

<sessionState timeout="40"/>

1. **What Are The Different Modes For The Sessionstates In The Web.config File?**

Off                Indicates that session state is not enabled.  
Inproc           Indicates that session state is stored locally.  
StateServer  Indicates that session state is stored on a remote server.  
SQLServer    Indicates that session state is stored on the SQL Server.

1. **Is It Possible For Me To Change My .aspx File Extension To Some Other Name?**

Yes.

* + Open IIS->Default Website -> Properties
  + Select HomeDirectory tab
  + Click on configuration button
  + Click on add. Enter aspnet\_isapi details
  + (C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\aspnet\_isapi.dll
  + GET,HEAD,POST,DEBUG)

Open  
machine.config(C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CONFIG)  
& add new extension under tag  
<add verb="\*" path="\*.santhosh" type="System.Web.UI.PageHandlerFactory"/>

1. **What Is A Webservice And What Is The Underlying Protocol Used In It?**

Web Services are applications delivered as a service on the Web. Web services allow for programmatic access of business logic over the Web. Web services typically rely on XML- based protocols, messages, and interface descriptions for communication and access. Web services are designed to be used by other programs or applications rather than directly by end user. Programs invoking a Web service are called clients. SOAP over HTTP is the most commonly used protocol for invoking Web services.

1. **In A Webservice, Need To Display 10 Rows From A Table. Which Is The Best Choice Among Datareader Or Dataset?**

WebService will support only DataSet.

1. **Are Web Services A Replacement For Other Distributed Computing Platforms?**

No. Web Services is just a new way of looking at existing implementation platforms.

1. **How To Generate Proxy Class Other Than .net App And Wsdl Tool?**

To access an XML Web service from a client application, you first add a Web reference, which is a reference to an XML Web service. When you create a Web reference, Visual Studio creates an XML Web service proxy class automatically and adds it to your project. This proxy class exposes the methods of the XML Web service and handles the marshalling of appropriate arguments back and forth between the XML Web service and your application. Visual Studio uses the Web Services Description Language (WSDL) to create the proxy. To generate an XML Web service proxy class:

From a command prompt, use Wsdl.exe to create a proxy class, specifying (at a minimum) the URL to an XML Web service or a service description, or the path to a saved service description.

Wsdl /language:language /protocol:protocol

/namespace:myNameSpace /out:filename

/username:username /password:password /domain:domain <url or path>

1. **What Is A Proxy In Web Service? How Do I Use A Proxy Server When Invoking A Web Service?**

If you are using the SOAP Toolkit, you need to set some connector properties to use a proxy server:

Dim soap As SoapClient Set soap=New SoapClient

soap.ConnectorProperty("ProxyServer") = ?proxyservername?

soap.ConnectorProperty("ProxyPort") = ?8080?

soap.ConnectorProperty("UseProxy") = True

While with .NET , you just need to create a System.Net.WebProxy object and use it to set the Proxy property Dim

webs As localhost.MyService() webs.Proxy=New

System.Net.WebProxy(?http://proxyserver:8080?)

1. **How You Will Protect / Secure A Web Service?**

For the most part, things that you do to secure a Web site can be used to secure a Web Service. If you need to encrypt the data exchange, you use Secure Sockets Layer (SSL) or a Virtual Private Network to keep the bits secure. For authentication, use HTTP Basic or Digest authentication with Microsoft® Windows® integration to figure out who the caller is. these items cannot:  
Parse a SOAP request for valid values  
Authenticate access at the Web Method level (they can authenticate at the Web Service level)  
Stop reading a request as soon as it is recognized as invalid.

1. **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.

1. **Cao And Sao.**

Client Activated objects are those remote objects whose Lifetime is directly Controlled by the client. This is in direct contrast to SAO. Where the server, not the client has complete control over the lifetime of the objects.  
Client activated objects are instantiated on the server as soon as the client request the object to be created. Unlike as SAO a CAO doesn’t delay the object creation until the first method is called on the object. (In SAO the object is instantiated when the client calls the method on the object)

1. **Difference Between Singleton And Singlecall.**

Singleton types never have more than one instance at any one time. If an instance exists, all client requests are serviced by that instance.

Single Call types always have one instance per client request. The next method invocation will be serviced by a different server instance, even if the previous instance has not yet been recycled by the system.

1. **In Which Process Does Iis Runs (was Asking About The Exe File)**

inetinfo.exe is the Microsoft 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 to the actual worker process aspnet\_wp.exe.

1. **Where Are The Iis Log Files Stored?**

C:\WINDOWS\system32\Logfiles\W3SVC1  
OR  
c:\winnt\system32\LogFiles\W3SVC1

1. **How Do I Send An Email Message From My Asp.net Page?**

You can use the System.Web.Mail.MailMessage and the System.Web.Mail.SmtpMail class to send email in your ASPX pages. Below is a simple example of using this class to send mail in C# and VB.NET. In order to send mail through our mail server, you would want to make sure to set the static SmtpServer property of the SmtpMail class to mail-fwd.

C#

<%@ Import Namespace="System" %>

<%@ Import Namespace="System.Web" %>

<%@ Import Namespace="System.Web.Mail" %>

<HTML>

<HEAD>

<title>Mail Test</title>

</HEAD>

<script language="C#" runat="server">

private void Page\_Load(Object sender, EventArgs e)

{

try

{

MailMessage mailObj = new MailMessage();

mailObj.From = "sales@joeswidgets.com";

mailObj.To = "ringleader@forexample-domain.com";

mailObj.Subject = "Your Widget Order";

mailObj.Body = "Your order was processed.";

mailObj.BodyFormat = MailFormat.Text;

SmtpMail.SmtpServer = "mail-fwd";

SmtpMail.Send(mailObj);

Response.Write("Mail sent successfully");

}

catch (Exception x)

{

Response.Write("Your message was not sent: " + x.Message);

}

}

</script>

<body>

<form id="mail\_test" method="post" runat="server">

</form>

</body>

</HTML>

1. **What Is An Il?**

Intermediate Language is also known as MSIL (Microsoft Intermediate Language) or CIL (Common Intermediate Language). All .NET source code is compiled to IL. IL is then converted to machine code at the point where the software is installed, or at run-time by a Just-In-Time (JIT) compiler.

1. **What Is Difference Between Namespace And Assembly?**

Assembly is physical grouping of logical units, Namespace, logically groups classes.  
Namespace can span multiple assembly.

1. **What Is The Use Of Errorprovider Control In .net?**

The ErrorProvider control is used to indicate invalid data on a data entry form. Using this control, you can attach error messages that display next to the control when the data is invalid, as seen in the following image. A red circle with an exclamation point blinks, and when the user mouses over the icon, the error message is displayed as a tooltip.

1. **Can Any Object Be Stored In A Viewstate In .net?**

An object that either is serializable or has a TypeConverter defined for it can be persisted in ViewState.

1. **Difference Between Class And Interface In .net?**
   * Class is logical representation of object. It is collection of data and related sub procedures with definition.
   * Interface is also a class containing methods which is not having any definitions.
   * Class does not support multiple inheritance. But interface can support.
2. **What Is Shadowing?**

Shadowing is either through scope or through inheritance. Shadowing through inheritance is hiding a method of a base class and providing a new implementation for the same. This is the default when a derived class writes an implementation of a method of base class which is not declared as overridden in the base class. This also serves the purpose of protecting an implementation of a new method against subsequent addition of a method with the same name in the base class.’shadows’ keyword is recommended although not necessary since it is the default.

1. **Differences Between Dll And Exe?**

**.exe**

* + These are outbound file.
  + Only one .exe file exists per application.
  + .Exe cannot be shared with other applications.

**.dll**

* + These are inbound file .
  + Many .dll files may exists in one application.
  + .dll can be shared with other applications.

1. **What Do You Mean By Authentication And Authorization?**

Authentication is the process of validating a user on the credentials(username and password) and authorization performs after authentication. After Authentication a user will be verified for performing the various tasks, It access is limited it is known as authorization.

1. **Where Do You Add An Event Handler?**

It's the Attributesproperty, the Add function inside that property. e.g.btnSubmit.Attributes.Add("onMouseOver","someClientCode();")

1. **Differentiate Between Client-side And Server-side Validations In Web Pages?**

Client-side validations take place at the client end with the help of JavaScript and VBScript before the Web page is sent to the server. On the other hand, server-side validations take place at the server end.

1. **What Does The Orientation Property Do In A Menu Control?**

Orientation property of the Menu control sets the horizontal or vertical display of a menu on a Web page. By default, the orientation is vertical.

1. **Which Method Is Used To Force All The Validation Controls To Run?**

The Page.Validate() method is used to force all the validation controls to run and to perform validation.

1. **What Is Viewstate?**

The ViewState is a feature used by ASP.NET Web page to store the value of a page and its controls just before posting the page. Once the page is posted, the first task by the page processing is to restore the ViewState to get the values of the controls.

1. **Differentiate Globalization And Localization?**

The globalization is a technique to identify the specific part of a Web application that is different for different languages and make separate that portion from the core of the Web application. The localization is a procedure of configuring a Web application to be supported for a specific language or locale.

1. **How Can You Register A Custom Server Control To A Web Page?**

You can register a custom server control to a Web page using the @Register directive.

1. **What Is Iis? Why Is It Used?**

Internet Information Services (IIS) is created by Microsoft to provide Internet-based services to ASP.NET Web applications. It makes your computer to work as a Web server and provides the functionality to develop and deploy Web applications on the server. IIS handles the request and response cycle on the Web server. It also offers the services of SMTP and FrontPage server extensions. The SMTP is used to send emails and use FrontPage server extensions to get the dynamic features of IIS, such as form handler.

1. **Define A Multilingual Website?**

A multilingual Website serves content in a number of languages. It contains multiple copies for its content and other resources, such as date and time, in different languages.

1. **What Are The Advantages Of The Code-behind Feature?**
   * Makes code easy to understand and debug by separating application logic from HTML tags.
   * Provides the isolation of effort between graphic designers and software engineers.
   * Removes the problems of browser incompatibility by providing code files to exist on the Web server and supporting Web pages to be compiled on demand.
2. **Which Is The Parent Class Of The Web Server Control?**

The System.Web.Ul.Control class is the parent class for all Web server controls.

1. **How Can We Identify That The Page Is Post Back?**

Page object has an "IsPostBack" property, which can be checked to know that is the page posted back.

1. **In Which Event Are The Controls Fully Loaded?**

Page load event guarantees that all controls are fully loaded. Controls are also accessed in Page\_Init events but you will see that view state is not fully loaded during this event

1. **Which Properties Are Used To Bind A Datagridview Control?**

The DataSource property and the DataMember property are used to bind a DataGridView control.

1. **What Is Autopostback?**

If you want a control to postback automatically when an event is raised, you need to set the AutoPostBack property of the control to True.

1. **What Are The Parameters That Control Most Of Connection Pooling Behaviours?**
   * Connect Timeout
   * Max Pool Size
   * Min Pool Size
   * Pooling
2. **What Are Different Types Of Authentication Techniques That Are Used In Connection Strings To Connect .net Applications With Microsoft Sql Server?**
   * The Windows Authentication option.
   * The SQL Server Authentication option.
3. **Which Adapter Should You Use, If You Want To Get The Data From An Access Database?**

OleDbDataAdapter is used to get the data from an Access database.

1. **What Are The Pre-requisites For Connection Pooling?**

There must be multiple processes to share the same connection describing the same parameters and security settings. The connection string must be identical.

1. **Name The Method That Needs To Be Invoked On The Dataadapter Control To Fill The Generated Dataset With Data?**

The Fill() method is used to fill the dataset with data.

1. **Which Property Is Used To Check Whether A Datareader Is Closed Or Opened?**

The IsClosed property is used to check whether a DataReader is closed or opened. This property returns a true value if a Data Reader is closed, otherwise a false value is returned.

1. **What Is The Role Of The Dataset Object In Ado.net?**

One of the major component of ADO.NET is the DataSet object, which always remains disconnected from the database and reduces the load on the database.

1. **Which Architecture Does Datasets Follow?**

Datasets follow the disconnected data architecture.

1. **Mention The Namespace That Is Used To Include .net Data Provider For Sql Server In .net Code?**

The System.Data.SqlClient namespace.

1. **What Is The Meaning Of Object Pooling?**

Object pooling is a concept of storing a pool (group) of objects in memory that can be reused later as needed. Whenever, a new object is required to create, an object from the pool can be allocated for this request; thereby, minimizing the object creation. A pool can also refer to a group of connections and threads. Pooling, therefore, helps in minimizing the use of system resources, improves system scalability, and performance.

1. **What Are Tuples?**

Tuple  is a fixed-size collection that can have elements of either same or different data types. Similar to arrays, a user must have to specify the size of a tuple at the time of declaration. Tuples are allowed to hold up from 1 to 8 elements and if there are more than 8 elements, then the 8th element can be defined as another tuple. Tuples can be specified as parameter or return type of a method.

1. **What Is The Difference Between Int And Int32?**

There is no difference between int and int32. System.Int32 is a .NET Class and int is an alias name for System.Int32.

1. **Which Method Do You Use To Enforce Garbage Collection In .net?**

The System.GC.Collect() method.

1. **What Is Managed Extensibility Framework?**

Managed extensibility framework (MEF) is a new library that is introduced as a part of .NET 4.0 and Silverlight 4. It helps in extending your application by providing greater reuse of applications and components. MEF provides a way for host application to consume external extensions without any configuration requirement.

1. **What Is Microsoft Intermediate Language (msil)?**

The .NET Framework is shipped with compilers of all .NET programming languages to develop programs. There are separate compilers for the Visual Basic, C#, and Visual C++ programming languages in .NET Framework. Each .NET compiler produces an intermediate code after compiling the source code. The intermediate code is common for all languages and is understandable only to .NET environment. This intermediate code is known as MSIL.

1. **Mention The Execution Process For Managed Code?**
   * Choosing a language compiler
   * Compiling the code to MSIL
   * Compiling MSIL to native code
   * Executing the code.
2. **What Are The Benefits Of .net Framework?**

.NET Framework offers many benefits to application developers. Some of these benefits are as follows:

* + **Consistent programming model** —.NET Framework provides a consistent object-oriented programming model across various languages. You on we this model to create programs for performing different tasks, such as connecting to and retrieving data from databases and reading from and writing to files.
  + **Language interoperability** —Language interoperability is a feature that enables a piece of code written in one language to be used in another language. This facilities the reuse of code and therefore improves the efficiency of the development process.
  + **Automatic management of resources** —Wink developing .NET. applications, you may use various resources, such as files, memory, and database connecters. With MET Framework, you do not need to manually free these resources when they are no longer required.
  + **Ease of deployment**- .NET framework makes the deployment of applications easier. To install an application that is not based on NET Framework you need to copy It and its components on target computers. However, with MET Framework, you can quickly Install or deploy the applications such that Installation of new applications or components does not affect the existing applications.

1. **Mention The Core Components Of .net Framework?**

The two core components of .NET Framework are:

* + Common Language Runtime
  + .NET Framework Class Library.[sociallocker]

1. **Briefly Describe The Roles Of Clr In .net Framework?**
   * CLR provides an environment to execute MET applications on target machines.
   * CLR-is also a common runtime environment for all MET code irrespective of their programming language, because the compilers of MET Framework convert every source code into a common language known as MSIL .
   * CLR also provides various services to execute processes, such as memory management service and security services..CLR performs various tasks to manage the execution process of MET applications.

The responsibilities of as are listed below:

* + Automatic memory management–CLR invokes venous built-in functions of MET Framework to allocate and de-allocate the memory of MET objects. Therefore, programmers need not write the code to explicitly allocate and de-allocate memory to programs.
  + Garbage Collection —Garbage collection Is the major role of CUR, with prevents memory leaks during execution of programs. The Garbage collector of CLR automatically determines the best time to free the memory, which is reserved by an object for execution.
  + Code Access Security —Code Access Security (CAS) model is used in MET Framework to impose restrictions and security during execution of programs. CLR uses security objects to manage access to code during execution of MET applications. as allows an executing code to perform only those tasks for which it has permission.

1. **Why Is String Called Immutable Data Type?**

A sting represents text and stores a sequential collection of characters in the memory. A string object is said to be immutable (read only), because a value once assigned to a string object cannot be changed after the acting object has been created. When the value In the string object is modified, a new string object is created with a new value assigned to the string object therefore, keeping the old string In memory for garbage collector to be disposed.

1. **What Is Side-by-side Execution? Can Two Applications, One Using A Private Assembly And Other Using A Shared Assembly, Be Stated As Side-by-side Executables?**

Side-by-side execution Is the ability to run multiple versions of an application or component on the same computer. You can have multiple versions of the CLR and multiple versions of applications and components Mat use a version of the runtime on the same computer at the same time. As versioning is only applied to shared assemblies and not to private assemblies, two applications, one using a private assembly and other using a shared assembly, cannot be stated as side-by-side executables.

1. **Explain The Differences Between Managed And Unmanaged Code?**

 Managed code is the code that is executed chatty by the QR Instead of the operating system. Unmanaged code is the code that is executed directly by the operating system outside the OR environment. In managed code, since the execution of the code Is governed by CLR, the runtime provides different services  such as garbage collection, type checking, exception handling, and safety  and support These sat help provide uniformly in platform and language-Independent behavior of managed code applications. In unmanaged code, the allocation of memory, type safety, and security  is required to be taken are of by the developer. If the unmanaged code is not Properly handles, may result In memory leak. Examples of unmanaged code are ActiveX components and Wm32 AM that execute beyond the scope of native CLR.

1. **What Is The Maximum Number Of Classes That Can Be Contained In One Dll File?**

There is no limit to the number of classes that can be curtained In a DLL file.

1. **Can One Dll File Contains The Compiled Code Of More Than One .net Language?**

No, a DLL file can contain the complied code of only one programming language.

1. **Explain The Different Types Of Assemblies?**

Assemblies are of two types, private and shared assemblies. A private assembly is used by the clients of the same application directory structure as the assembly. A shared assembly is stored in the global assembly cache (GAC), which is a repository of assemblies maintained by the runtime. A Shared assembly can be referenced by more than one application.

C##

1. **What's The Advantage Of Using System.text.stringbuilder Over System.string?**

StringBuilder is more efficient in the cases, where a lot of manipulation is done to the text. Strings are im mutable, so each time it’s being operated on, a new instance is created.

1. **Can You Store Multiple Data Types In System.array?**

No.

1. **What's The Difference Between The System.array.copyto() And System.array.clone()?**

The first one performs a deep copy of the array, the second one is shallow.

1. **How Can You Sort The Elements Of The Array In Descending Order?**

By calling Sort() and then Reverse() methods.

1. **What's The .net Datatype That Allows The Retrieval Of Data By A Unique Key?**

HashTable.

1. **What's Class Sortedlist Underneath?**

A sorted HashTable.

1. **Will Finally Block Get Executed If The Exception Had Not Occurred?**

Yes.

1. **What's The C# Equivalent Of C++ Catch (....), Which Was A Catch-all Statement For Any Possible Exception?**

A catch block that catches the exception of type System.Exception. You can also omit the parameter data type in this case and just write catch {}.

1. **Can Multiple Catch Blocks Be Executed?**

No, once the proper catch code fires off, the control is transferred to the finally block (if there are any), and then whatever follows the finally block.

1. **Why Is It A Bad Idea To Throw Your Own Exceptions?**

Well, if at that point you know that an error has occurred, then why not write the proper code to handle that error instead of passing a new Exception object to the catch block? Throwing your own exceptions signifies some design flaws in the project.

1. **What's A Delegate?**

A delegate object encapsulates a reference to a method. In C++ they were referred to as function pointers.

1. **What's A Multicast Delegate?**

It’s a delegate that points to and eventually fires off several methods.

1. **How's The Dll Hell Problem Solved In .net?**

Assembly versioning allows the application to specify not only the library it needs to run (which was available under Win32), but also the version of the assembly.

1. **What Are The Ways To Deploy An Assembly?**

An MSI installer, a CAB archive, and XCOPY command.

1. **What's A Satellite Assembly?**

When you write a multilingual or multi-cultural application in .NET, and want to distribute the core application separately from the localized modules, the localized assemblies that modify the core application are called satellite assemblies.

1. **What Namespaces Are Necessary To Create A Localized Application?**

System.Globalization, System.Resources.

1. **What's The Difference Between // Comments, /\* \*/ Comments And /// Comments?**

Single-line, multi-line and XML documentation comments.

1. **How Do You Generate Documentation From The C# File Commented Properly With A Command-line Compiler?**

Compile it with a /doc switch.

1. **What's The Difference Between <c> And <code> Xml Documentation Tag?**

Single line code example and multiple-line code example.

1. **Is Xml Case-sensitive?**

Yes, so <Student> and <student> are different elements.

1. **What Debugging Tools Come With The .net Sdk?**

CorDBG – command-line debugger, and DbgCLR – graphic debugger. Visual Studio .NET uses the DbgCLR. To use CorDbg, you must compile the original C# file using the /debug switch.

1. **What Does The This Window Show In The Debugger?**

It points to the object that’s pointed to by this reference. Object’s instance data is shown.

1. **What Does Assert() Do?**

In debug compilation, assert takes in a Boolean condition as a parameter, and shows the error dialog if the condition is false. The program proceeds without any interruption if the condition is true.

1. **What's The Difference Between The Debug Class And Trace Class? Documentation Looks The Same.**

Use Debug class for debug builds, use Trace class for both debug and release builds.

1. **Why Are There Five Tracing Levels In System.diagnostics.traceswitcher?**

The tracing dumps can be quite verbose and for some applications that are constantly running you run the risk of overloading the machine and the hard drive there. Five levels range from None to Verbose, allowing to fine-tune the tracing activities.

1. **Where Is The Output Of Textwritertracelistener Redirected?**

To the Console or a text file depending on the parameter passed to the constructor.

1. **How Do You Debug An Asp.net Web Application?**

Attach the aspnet\_wp.exe process to the DbgClr debugger.

1. **What Are Three Test Cases You Should Go Through In Unit Testing?**

Positive test cases (correct data, correct output), negative test cases (broken or missing data, proper handling), exception test cases (exceptions are thrown and caught properly).

1. **Can You Change The Value Of A Variable While Debugging A C# Application?**

Yes, if you are debugging via Visual Studio.NET, just go to Immediate window.

1. **Explain The Three Services Model (three-tier Application).**

Presentation (UI), business (logic and underlying code) and data (from storage or other sources).

1. **What Are Advantages And Disadvantages Of Microsoft-provided Data Provider Classes In Ado.net?**

SQLServer.NET data provider is high-speed and robust, but requires SQL Server license purchased from Microsoft. OLE-DB.NET is universal for accessing other sources, like Oracle, DB2, Microsoft Access and Informix, but it’s a .NET layer on top of OLE layer, so not the fastest thing in the world. ODBC.NET is a deprecated layer provided for backward compatibility to ODBC engines.

1. **What's The Role Of The Datareader Class In Ado.net Connections?**

It returns a read-only dataset from the data source when the command is executed.

1. **What Is The Wildcard Character In Sql? Let's Say You Want To Query Database With Like For All Employees Whose Name Starts With La.**

The wildcard character is %, the proper query with LIKE would involve ‘La%’.

1. **Explain Acid Rule Of Thumb For Transactions.**

Transaction must be Atomic (it is one unit of work and does not dependent on previous and following transactions), Consistent (data is either committed or roll back, no “in-between” case where something has been updated and something hasn’t), Isolated (no transaction sees the intermediate results of the current transaction), Durable (the values persist if the data had been committed even if the system crashes right after).

1. **What Connections Does Microsoft Sql Server Support?**

Windows Authentication (via Active Directory) and SQL Server authentication (via Microsoft SQL Server user name and passwords).

1. **Which One Is Trusted And Which One Is Untrusted?**

Windows Authentication is trusted because the username and password are checked with the Active Directory, the SQL Server authentication is untrusted, since SQL Server is the only verifier participating in the transaction.

1. **Why Would You Use Untrusted Verificaion?**

Web Services might use it, as well as non-Windows applications.

1. **What Does The Parameter Initial Catalog Define Inside Connection String?**

The database name to connect to.

1. **What's The Data Provider Name To Connect To Access Database?**

Microsoft.Access.

1. **What Does Dispose Method Do With The Connection Object?**

Deletes it from the memory.

1. **What Is A Pre-requisite For Connection Pooling?**

Multiple processes must agree that they will share the same connection, where every parameter is the same, including the security settings.

1. **What Is C#?**

C# is a programming language designed by Microsoft. It is loosely based on C/C++, and bears a striking similarity to Java. Microsoft describe C# as follows:

"C# is a simple, modern, object oriented, and type-safe programming language derived from C and C++. C# (pronounced 'C sharp') is firmly planted in the C and C++ family tree of languages, and will immediately be familiar to C and C++ programmers. C# aims to combine the high productivity of Visual Basic and the raw power of C++."

1. **How Do I Develop C# Apps?**

The (free) .NET SDK contains the C# command-line compiler (csc.exe). Visual Studio has fully integrated support for C# development. On Linux you can use Mono.

1. **Does C# Replace C++?**

There are three options open to the Windows developer from a C++ background:

•Stick with standard C++. Don't use .NET at all.  
•Use C++ with .NET. Microsoft supply a .NET C++ compiler that produces IL rather than machine code. However to make full use of the .NET environment (e.g. garbage collection), a set of extensions are required to standard C++. In .NET 1.x this extended language is called Managed Extensions for C++. In .NET 2.0 ME C++ has been completely redesigned under the stewardship of Stan Lippman, and renamed C++/CLI.  
•Forget C++ and use C#.

Each of these options has merits, depending on the developer and the application. For my own part, I intend to use C# where possible, falling back to C++ only where necessary. ME C++ (soon to be C++/CLI) is very useful for interop between new .NET code and old C++ code - simply write a managed wrapper class using ME C++, then use the managed class from C#. From experience, this works well.

1. **Does C# Have Its Own Class Library?**

Not exactly. The .NET Framework has a comprehensive class library, which C# can make use of. C# does not have its own class library.

1. **What Standard Types Does C# Use?**

C# supports a very similar range of basic types to C++, including int, long, float, double, char, string, arrays, structs and classes. However, don't assume too much. The names may be familiar, but many of the details are different. For example, a long is 64 bits in C#, whereas in C++ the size of a long depends on the platform (typically 32 bits on a 32-bit platform, 64 bits on a 64-bit platform). Also classes and structs are almost the same in C++ - this is not true for C#. Finally, chars and strings in .NET are 16-bit (Unicode/UTF-16), not 8-bit like C++.

1. **Is It True That All C# Types Derive From A Common Base Class?**

Yes and no. All types can be treated as if they derive from object (System.Object), but in order to treat an instance of a value type (e.g. int, float) as object-derived, the instance must be converted to a reference type using a process called 'boxing'. In theory a developer can forget about this and let the run-time worry about when the conversion is necessary, but in reality this implicit conversion can have side-effects that may trip up the unwary.

1. **What Are The Fundamental Differences Between Value Types And Reference Types?**

C# divides types into two categories - value types and reference types. Most of the intrinsic types (e.g. int, char) are value types. Structs are also value types. Reference types include classes, arrays and strings. The basic idea is straightforward - an instance of a value type represents the actual data, whereas an instance of a reference type represents a pointer or reference to the data.

The most confusing aspect of this for C++ developers is that C# has predetermined which types are represented as values, and which are represented as references. A C++ developer expects to take responsibility for this decision.

For example, in C++ we can do this:  
int x1 = 3; // x1 is a value on the stack  
int \*x2 = new int(3) // x2 is a pointer to a value on the heap  
but in C# there is no control:  
int x1 = 3; // x1 is a value on the stack  
int x2 = new int();  
x2 = 3; // x2 is also a value on the stack!

1. **Okay, So An Int Is A Value Type, And A Class Is A Reference Type. How Can Int Be Derived From Object?**

It isn't, really. When an int is being used as an int, it is a value. However, when it is being used as an object, it is a reference to an integer value (on the managed heap). In other words, when you treat an int as an object, the runtime automatically converts the int value to an object reference. This process is called boxing. The conversion involves copying the int to the heap, and creating an object instance which refers to it. Unboxing is the reverse process - the object is converted back to a value.

int x = 3; // new int value 3 on the stack  
object objx = x; // new int on heap, set to value 3 - still have x=3 on stack  
int y = (int)objx; // new value 3 on stack, still got x=3 on stack and objx=3 on heap

1. **Are C# References The Same As C++ References?**

Not quite. The basic idea is the same, but one significant difference is that C# references can be null . So you cannot rely on a C# reference pointing to a valid object. In that respect a C# reference is more like a C++ pointer than a C++ reference. If you try to use a null reference, a NullReferenceException is thrown.

For example, look at the following method:  
void displayStringLength( string s )  
{  
Console.WriteLine( "String is length {0}", s.Length );  
}  
The problem with this method is that it will throw a NullReferenceException if called like this:  
string s = null;  
displayStringLength( s );

Of course for some situations you may deem a NullReferenceException to be a perfectly acceptable outcome, but in this case it might be better to re-write the method like this:  
void displayStringLength( string s )  
{  
if( s == null )  
Console.WriteLine( "String is null" );  
else  
Console.WriteLine( "String is length {0}", s.Length );  
}

1. **Structs Are Largely Redundant In C++. Why Does C# Have Them?**

In C++, a struct and a class are pretty much the same thing. The only difference is the default visibility level (public for structs, private for classes). However, in C# structs and classes are very different. In C#, structs are value types (instances stored directly on the stack, or inline within heap-based objects), whereas classes are reference types (instances stored on the heap, accessed indirectly via a reference). Also structs cannot inherit from structs or classes, though they can implement interfaces. Structs cannot have destructors. A C# struct is much more like a C struct than a C++ struct.

1. **Does C# Support Multiple Inheritance (mi)?**

No, though it does support implementation of multiple interfaces on a single class or struct.

1. **Is A C# Interface The Same As A C++ Abstract Class?**

No, not quite. An abstract class in C++ cannot be instantiated, but it can (and often does) contain  
implementation code and/or data members. A C# interface cannot contain any implementation code or data members - it is simply a group of method names & signatures. A C# interface is more like a COM interface than a C++ abstract class.

1. **Are C# Constructors The Same As C++ Constructors?**

Very similar, but there are some significant differences. First, C# supports constructor chaining. This means one constructor can call another:  
class Person  
{  
public Person( string name, int age ) { ... }  
public Person( string name ) : this( name, 0 ) {}  
public Person() : this( "", 0 ) {}  
}

Another difference is that virtual method calls within a constructor are routed to the most derived implementation - see Can I Call a virtual method from a constructor.

Error handling is also somewhat different. If an exception occurs during construction of a C# object, the destuctor (finalizer) will still be called. This is unlike C++ where the destructor is not called if construction is not completed. (Thanks to Jon Jagger for pointing this out.)

Finally, C# has static constructors. The static constructor for a class runs before the first instance of the class is created.

Also note that (like C++) some C# developers prefer the factory method pattern over constructors. See Brad Wilson's article.

1. **Are C# Destructors The Same As C++ Destructors?**

No. They look the same but they are very different. The C# destructor syntax (with the familiar ~ character) is just syntactic sugar for an override of the System.Object Finalize method. This Finalize method is called by the garbage collector when it determines that an object is no longer referenced, before it frees the memory associated with the object. So far this sounds like a C++ destructor. The difference is that the garbage collector makes no guarantees about when this procedure happens. Indeed, the algorithm employed by the CLR garbage collector means that it may be a long time after the application has finished with the object. This lack of certainty is often termed 'non-deterministic finalization', and it means that C# destructors are not suitable for releasing scarce resources such as database connections, file handles etc.

To achieve deterministic destruction, a class must offer a method to be used for the purpose. The standard approach is for the class to implement the IDisposable interface. The user of the object must call the Dispose() method when it has finished with the object. C# offers the 'using' construct to make this easier.

1. **If C# Destructors Are So Different To C++ Destructors, Why Did Ms Use The Same Syntax?**

Presumably they wanted C++ programmers to feel at home. I think they made a mistake.

1. **Are All Methods Virtual In C#?**

No. Like C++, methods are non-virtual by default, but can be marked as virtual.

1. **How Do I Declare A Pure Virtual Function In C#?**

Use the abstract modifier on the method. The class must also be marked as abstract (naturally). Note that abstract methods cannot have an implementation (unlike pure virtual C++ methods).

1. **Can I Call A Virtual Method From A Constructor/destructor?**

Yes, but it's generally not a good idea. The mechanics of object construction in .NET are quite different from C++, and this affects virtual method calls in constructors.

C++ constructs objects from base to derived, so when the base constructor is executing the object is effectively a base object, and virtual method calls are routed to the base class implementation. By contrast, in .NET the derived constructor is executed first, which means the object is always a derived object and virtual method calls are always routed to the derived implementation. (Note that the C# compiler inserts a call to the base class constructor at the start of the derived constructor, thus preserving standard OO semantics by creating the illusion that the base constructor is executed first.)

The same issue arises when calling virtual methods from C# destructors. A virtual method call in a base destructor will be routed to the derived implementation.

1. **Should I Make My Destructor Virtual?**

A C# destructor is really just an override of the System.Object Finalize method, and so is virtual by definition.

1. **Can I Use Exceptions In C#?**

Yes, in fact exceptions are the recommended error-handling mechanism in C# (and in .NET in general). Most of the .NET framework classes use exceptions to signal errors.

1. **What Types Of Object Can I Throw As Exceptions?**

Only instances of the System.Exception classes, or classes derived from System.Exception. This is in sharp contrast with C++ where instances of almost any type can be thrown.

1. **Can I Define My Own Exceptions?**

Yes, just derive your exception class from System.Exception.

1. **Does The System.exception Class Have Any Cool Features?**

Yes - the feature which stands out is the StackTrace property. This provides a call stack which records where the exception was thrown from. For example, the following code:  
using System;  
class CApp  
{  
public static void Main()  
{  
try  
{  
f();  
}  
catch( Exception e )  
{  
Console.WriteLine( "System.Exception stack trace = \n{0}", e.StackTrace );  
}  
}   
static void f()  
{  
throw new Exception( "f went pear-shaped" );  
}   
}  
produces this output:  
System.Exception stack trace =   
at CApp.f()  
at CApp.Main()

Note, however, that this stack trace was produced from a debug build. A release build may optimise away some of the method calls which could mean that the call stack isn't quite what you expect.

1. **When Should I Throw An Exception?**

This is the subject of some debate, and is partly a matter of taste. However, it is accepted by many that exceptions should be thrown only when an 'unexpected' error occurs. How do you decide if an error is expected or unexpected? This is a judgement call, but a straightforward example of an expected error is failing to read from a file because the seek pointer is at the end of the file, whereas an example of an unexpected error is failing to allocate memory from the heap.

1. **Does C# Have A 'throws' Clause?**

No, unlike Java, C# does not require (or even allow) the developer to specify the exceptions that a method can throw.

1. **How Can I Check The Type Of An Object At Runtime?**

You can use the is keyword. For example:  
using System;  
class CApp  
{  
public static void Main()  
{  
string s = "fred";  
long i = 10;  
Console.WriteLine( "{0} is {1}an integer", s, (IsInteger(s) ? "" : "not ") );  
Console.WriteLine( "{0} is {1}an integer", i, (IsInteger(i) ? "" : "not ") );  
}  
static bool IsInteger( object obj )  
{  
if( obj is int || obj is long )  
return true;  
else   
return false;  
}  
}  
produces the output:  
fred is not an integer   
10 is an integer

1. **Can I Get The Name Of A Type At Runtime?**

Yes, use the GetType method of the object class (which all types inherit from). For example:  
using System;  
class CTest  
{  
class CApp  
{  
public static void Main()  
{  
long i = 10;  
CTest ctest = new CTest();  
DisplayTypeInfo( ctest );  
DisplayTypeInfo( i );  
}  
static void DisplayTypeInfo( object obj )  
{  
Console.WriteLine( "Type name = {0}, full type name = {1}", obj.GetType(), obj.GetType().FullName );  
}  
}  
}  
produces the following output:  
Type name = CTest, full type name = CTest   
Type name = Int64, full type name = System.Int64

1. **How Do I Do A Case-insensitive String Comparison?**

Use the String.Compare function. Its third parameter is a boolean which specifies whether case should be ignored or not.

"fred" == "Fred" // false  
System.String.Compare( "fred", "Fred", true ) // true

1. **Does C# Support A Variable Number Of Arguments?**

Yes, using the params keyword. The arguments are specified as a list of arguments of a specific type, e.g. int. For ultimate flexibility, the type can be object. The standard example of a method which uses this approach is System.Console.WriteLine().

1. **How Can I Process Command-line Arguments?**

Like this:  
using System;  
class CApp  
{  
public static void Main( string[] args )  
{  
Console.WriteLine( "You passed the following arguments:" );  
foreach( string arg in args )  
Console.WriteLine( arg );  
}  
}

1. **Does C# Do Array Bounds Checking?**

Yes. An IndexOutOfRange exception is used to signal an error.

1. **How Can I Make Sure My C# Classes Will Interoperate With Other .net Languages?**

Make sure your C# code conforms to the Common Language Subset (CLS). To help with this, add the [assembly:CLSCompliant(true)] global attribute to your C# source files. The compiler will emit an error if you use a C# feature which is not CLS-compliant.

1. **How Do I Use The 'using' Keyword With Multiple Objects?**

You can nest using statements, like this:  
using( obj1 )  
{  
using( obj2 )  
{  
...   
}  
}

However consider using this more aesthetically pleasing (but functionally identical) formatting:  
using( obj1 )  
using( obj2 )  
{  
...  
}

1. **What Is The Difference Between == And Object.equals?**

For value types, == and Equals() usually compare two objects by value. For example:  
int x = 10;  
int y = 10;  
Console.WriteLine( x == y );  
Console.WriteLine( x.Equals(y) );  
will display:  
True  
True

However things are more complex for reference types. Generally speaking, for reference types == is expected to perform an identity comparison, i.e. it will only return true if both references point to the same object. By contrast, Equals() is expected to perform a value comparison, i.e. it will return true if the references point to objects that are equivalent. For example:  
StringBuilder s1 = new StringBuilder("fred");  
StringBuilder s2 = new StringBuilder("fred");  
Console.WriteLine( s1 == s2 );  
Console.WriteLine( s1.Equals(s2) );  
will display:  
False  
True

s1 and s2 are different objects (hence == returns false), but they are equivalent (hence Equals() returns true).

Unfortunately there are exceptions to these rules. The implementation of Equals() in System.Object (the one you'll inherit by default if you write a class) compares identity, i.e. it's the same as operator==. So Equals() only tests for equivalence if the class author overrides the method (and implements it correctly). Another exception is the string class - its operator== compares value rather than identity.

Bottom line: If you want to perform an identity comparison use the ReferenceEquals() method. If you want to perform a value comparison, use Equals() but be aware that it will only work if the type has overridden the default implementation. Avoid operator== with reference types (except perhaps strings), as it's simply too ambiguous.

1. **How Do I Enforce Const Correctness In C#?**

You can't - at least not in the same way you do in C++. C# (actually, the CLI) has no real concept of const correctness, For example, there's no way to specify that a method should not modify an argument passed in to it. And there's no way to specify that a method does not modify the object on which it is acting.

To get a feel for the angst this causes among some C++ programmers, read the feedback on this post from Raymond Chen.

There are of course ways of addressing this issue. For example, see Brad Abram's post (and associated feedback) for some ideas on adding optional read-only behaviour to collection classes.

1. **What Are The New Features In C# 2.0?**

Support for all of the new framework features such as generics, anonymous methods, partial classes, iterators and static classes.

Delegate inference is a new feature of the C# compiler which makes delegate usage a little simpler. It allows you to write this:  
Thread t = new Thread(ThreadFunc);  
instead of this:  
Thread t = new Thread( new ThreadStart(ThreadFunc) );

Another minor but welcome addition is the explicit global namespace, which fixes a hole in namespace usage in C# 1.x. You can prefix a type name with global:: to indicate that the type belongs to the global namespace, thus avoiding problems where the compiler infers the namespace and gets it wrong.

Finally C# 2.0 includes some syntactic sugar for the new System.Nullable type. You can use T? as a synonym for System.Nullable<T>, where T is a value type. As suggested by the name, this allows values of the type to be 'null', or 'undefined'.

1. **Are C# Generics The Same As C++ Templates?**

No, not really. There are some similarities, but there are also fundamental differences.

1. **Explain What Is An Interface In C#?**

An Interface in C# is created using the interface keyword. An example is shown below.

using System;  
namespace Interfaces  
{  
interface IBankCustomer  
{  
void DepositMoney();  
void WithdrawMoney();  
}  
public class Demo : IBankCustomer  
{  
public void DepositMoney()  
{  
Console.WriteLine("Deposit Money");  
}  
public void WithdrawMoney()  
{  
Console.WriteLine("Withdraw Money");  
}  
public static void Main()  
{  
Demo DemoObject = new Demo();  
DemoObject.DepositMoney();  
DemoObject.WithdrawMoney();  
}  
}  
}

In our example we created IBankCustomer interface. The interface declares 2 methods.  
1. void DepositMoney();  
2. void WithdrawMoney();

Notice that method declarations does not have access modifiers like public, private, etc. By default all interface members are public. It is a compile time error to use access modifiers on interface member declarations. Also notice that the interface methods have only declarations and not implementation. It is a compile time error to provide implementation for any interface member. In our example as the Demo class is inherited from the IBankCustomer interface, the Demo class has to provide the implementation for both the methods (WithdrawMoney() and DepositMoney()) that is inherited from the interface. If the class fails to provide implementation for any of the inherited interface member, a compile time error will be generated. Interfaces can consist of methods, properties, events, indexers, or any combination of those four member types. When a class or a struct inherits an interface, the class or struct must provide implementation for all of the members declared in the interface. The interface itself provides no functionality that a class or struct can inherit in the way that base class functionality can be inherited. However, if a base class implements an interface, the derived class inherits that implementation.

1. **Can An Interface Contain Fields?**

No, an Interface cannot contain fields.

1. **What Is The Difference Between Class Inheritance And Interface Inheritance?**

Classes and structs can inherit from interfaces just like how classes can inherit a base class or struct. However there are 2 differences.  
1. A class or a struct can inherit from more than one interface at the same time where as A class or a struct cannot inherit from more than one class at the same time. An example depicting the same is shown below.

using System;  
namespace Interfaces  
{  
interface Interface1  
{  
void Interface1Method();  
}  
interface Interface2  
{  
void Interface2Method();  
}  
class BaseClass1  
{  
public void BaseClass1Method()  
{  
Console.WriteLine("BaseClass1 Method");  
}  
}  
class BaseClass2  
{  
public void BaseClass2Method()  
{  
Console.WriteLine("BaseClass2 Method");  
}  
}  
//Error : A class cannot inherit from more than one class at the same time  
//class DerivedClass : BaseClass1, BaseClass2  
//{  
//}  
//A class can inherit from more than one interface at the same time  
public class Demo : Interface1, Interface2  
{  
public void Interface1Method()  
{  
Console.WriteLine("Interface1 Method");  
}  
public void Interface2Method()  
{  
Console.WriteLine("Interface2 Method");  
}  
public static void Main()  
{  
Demo DemoObject = new Demo();  
DemoObject.Interface1Method();  
DemoObject.Interface2Method();  
}  
}  
}

2. When a class or struct inherits an interface, it inherits only the method names and signatures, because the interface itself contains no implementations.

1. **Can An Interface Inherit From Another Interface?**

Yes, an interface can inherit from another interface. It is possible for a class to inherit an interface multiple times, through base classes or interfaces it inherits. In this case, the class can only implement the interface one time, if it is declared as part of the new class. If the inherited interface is not declared as part of the new class, its implementation is provided by the base class that declared it. It is possible for a base class to implement interface members using virtual members; in that case, the class inheriting the interface can change the interface behavior by overriding the virtual members.

1. **Can You Create An Instance Of An Interface?**

No, you cannot create an instance of an interface.

1. **If A Class Inherits An Interface, What Are The 2 Options Available For That Class?**

Option 1: Provide Implementation for all the members inheirted from the interface.

namespace Interfaces  
{  
interface Interface1  
{  
void Interface1Method();  
}  
class BaseClass1 : Interface1  
{  
public void Interface1Method()  
{  
Console.WriteLine("Interface1 Method");  
}  
public void BaseClass1Method()  
{  
Console.WriteLine("BaseClass1 Method");  
}  
}  
}

Option 2: If the class does not wish to provide Implementation for all the members inheirted from the interface, then the class has to be marked as abstract.

namespace Interfaces  
{  
interface Interface1  
{  
void Interface1Method();  
}  
abstract class BaseClass1 : Interface1  
{  
abstract public void Interface1Method();  
public void BaseClass1Method()  
{  
Console.WriteLine("BaseClass1 Method");  
}  
}  
}

1. **A Class Inherits From 2 Interfaces And Both The Interfaces Have The Same Method Name As Shown Below. How Should The Class Implement The Drive Method For Both Car And Bus Interface?**

namespace Interfaces  
{  
interface Car  
{  
void Drive();  
}  
interface Bus  
{  
void Drive();  
}  
class Demo : Car,Bus  
{  
//How to implement the Drive() Method inherited from Bus and Car  
}  
}

To implement the Drive() method use the fully qualified name as shown in the example below. To call the respective interface drive method type cast the demo object to the respective interface and then call the drive method.

using System;  
namespace Interfaces  
{  
interface Car  
{  
void Drive();  
}  
interface Bus  
{  
void Drive();  
}  
class Demo : Car,Bus  
{  
void Car.Drive()  
{  
Console.WriteLine("Drive Car");  
}  
void Bus.Drive()  
{  
Console.WriteLine("Drive Bus");  
}  
static void Main()  
{  
Demo DemoObject = new Demo();  
((Car)DemoObject).Drive();  
((Bus)DemoObject).Drive();  
}  
}  
}

1. **What Do You Mean By "explicitly Implemeting An Interface". Give An Example?**

If a class is implementing the inherited interface member by prefixing the name of the interface, then the class is "Explicitly Implemeting an Interface member". The disadvantage of Explicitly Implemeting an Interface member is that, the class object has to be type casted to the interface type to invoke the interface member. An example is shown below.

using System;  
namespace Interfaces  
{  
interface Car  
{  
void Drive();  
}  
class Demo : Car  
{  
// Explicit implementation of an interface member  
void Car.Drive()  
{  
Console.WriteLine("Drive Car");  
}  
static void Main()  
{  
Demo DemoObject = new Demo();  
//DemoObject.Drive();  
// Error: Cannot call explicitly implemented interface method  
// using the class object.  
// Type cast the demo object to interface type Car  
((Car)DemoObject).Drive();  
}  
}  
}

1. **What Is A Partial Class. Give An Example?**

A partial class is a class whose definition is present in 2 or more files. Each source file contains a section of the class, and all parts are combined when the application is compiled. To split a class definition, use the partial keyword as shown in the example below. Student class is split into 2 parts. The first part defines the study() method and the second part defines the Play() method. When we compile this program both the parts will be combined and compiled. Note that both the parts uses partial keyword and public access modifier.

using System;  
namespace PartialClass  
{  
public partial class Student  
{  
public void Study()  
{  
Console.WriteLine("I am studying");  
}  
}  
public partial class Student  
{  
public void Play()  
{  
Console.WriteLine("I am Playing");  
}  
}  
public class Demo  
{  
public static void Main()  
{  
Student StudentObject = new Student();  
StudentObject.Study();  
StudentObject.Play();  
} }}

It is very important to keep the following points in mind when creating partial classes.  
1. All the parts must use the partial keyword.  
2. All the parts must be available at compile time to form the final class.  
3. All the parts must have the same access modifiers - public, private, protected etc.  
4. Any class members declared in a partial definition are available to all the other parts.  
5. The final class is the combination of all the parts at compile time.

1. **What Are The Advantages Of Using Partial Classes?**

1. When working on large projects, spreading a class over separate files enables multiple programmers to work on it at the same time.

2. When working with automatically generated source, code can be added to the class without having to recreate the source file. Visual Studio uses this approach when it creates Windows Forms, Web service wrapper code, and so on. You can create code that uses these classes without having to modify the file created by Visual Studio.

1. **Is It Possible To Create Partial Structs, Interfaces And Methods?**

Yes, it is possible to create partial structs, interfaces and methods. We can create partial structs, interfaces and methods the same way as we create partial classes.

1. **Will The Following Code Compile?**

using System;  
namespace PartialClass  
{  
public partial class Student  
{  
public void Study()  
{  
Console.WriteLine("I am studying");  
}  
}  
public abstract partial class Student  
{  
public void Play()  
{  
Console.WriteLine("I am Playing");  
}  
}  
public class Demo  
{  
public static void Main()  
{  
Student StudentObject = new Student();  
} }}

No, a compile time error will be generated stating "Cannot create an instance of the abstract class or interface "PartialClass.Student". This is because, if any part is declared abstract, then the whole class becomes abstract. Similarly if any part is declared sealed, then the whole class becomes sealed and if any part declares a base class, then the whole class inherits that base class.

1. **Can You Create Partial Delegates And Enumerations?**

No, you cannot create partial delegates and enumerations.

1. **Can Different Parts Of A Partial Class Inherit From Different Interfaces?**

Yes, different parts of a partial class can inherit from different interfaces.

1. **Can You Specify Nested Classes As Partial Classes?**

Yes, nested classes can be specified as partial classes even if the containing class is not partial. An example is shown below.

class ContainerClass  
{  
public partial class Nested  
{  
void Test1() { }  
}  
public partial class Nested  
{  
void Test2() { }  
}  
}

1. **How Do You Create Partial Methods?**

To create a partial method we create the declaration of the method in one part of the partial class and implementation in the other part of the partial class. The implementation is optional. If the implementation is not provided, then the method and all the calls to the method are removed at compile time. Therefore, any code in the partial class can freely use a partial method, even if the implementation is not supplied. No compile-time or run-time errors will result if the method is called but not implemented. In summary a partial method declaration consists of two parts. The definition, and the implementation. These may be in separate parts of a partial class, or in the same part. If there is no implementation declaration, then the compiler optimizes away both the defining declaration and all calls to the method.

The following are the points to keep in mind when creating partial methods.  
1. Partial method declarations must begin partial keyword.  
2. The return type of a partial method must be void.  
3. Partial methods can have ref but not out parameters.  
4. Partial methods are implicitly private, and therefore they cannot be virtual.  
5. Partial methods cannot be extern, because the presence of the body determines whether they are defining or implementing.

1. **What Is The Use Of Partial Methods?**

Partial methods can be used to customize generated code. They allow for a method name and signature to be reserved, so that generated code can call the method but the developer can decide whether to implement the method. Much like partial classes, partial methods enable code created by a code generator and code created by a human developer to work together without run-time costs.

1. **What Is A Nested Type. Give An Example?**

A type(class or a struct) defined inside another class or struct is called a nested type. An example is shown below. InnerClass is inside ContainerClass, Hence InnerClass is called as nested class.

using System;  
namespace Nested  
{  
class ContainerClass  
{  
class InnerClass  
{  
public string str = "A string variable in nested class";  
}  
public static void Main()  
{  
InnerClass nestedClassObj = new InnerClass();  
Console.WriteLine(nestedClassObj.str);  
}  
}  
}

1. **Can The Nested Class Access, The Containing Class. Give An Example?**

Yes, the nested class, or inner class can access the containing or outer class as shown in the example below. Nested types can access private and protected members of the containing type, including any inherited private or protected members.

using System;  
namespace Nested  
{  
class ContainerClass  
{  
string OuterClassVariable = "I am an outer class variable";  
public class InnerClass  
{  
ContainerClass ContainerClassObject = new ContainerClass();  
string InnerClassVariable = "I am an Inner class variable";  
public InnerClass()  
{  
Console.WriteLine(ContainerClassObject.OuterClassVariable);  
Console.WriteLine(this.InnerClassVariable);  
}  
}  
}  
class Demo  
{  
public static void Main()  
{  
ContainerClass.InnerClass nestedClassObj = new ContainerClass.InnerClass();  
}  
}  
}

1. **What Is The Ouput Of The Following Program?**

using System;  
namespace Nested  
{  
class ContainerClass  
{  
public ContainerClass()  
{  
Console.WriteLine("I am a container class");  
}  
public class InnerClass : ContainerClass  
{  
public InnerClass()  
{  
Console.WriteLine("I am an inner class");  
}  
}  
}  
class DemoClass : ContainerClass.InnerClass  
{  
public DemoClass()  
{  
Console.WriteLine("I am a Demo class");  
}  
public static void Main()  
{  
DemoClass DC = new DemoClass();  
}  
}  
}  
Output:  
I am a container class  
I am an inner class  
I am a Demo class

The above program has used the concepts of inheritance and nested classes. The ContainerClass is at the top in the inheritance chain. The nested InnerClass derives from outer ContainerClass. Finally the DemoClass derives from nested InnerClass. As all the 3 classes are related by inheritance we have the above output.

1. **What Is A Destructor?**

A Destructor has the same name as the class with a tilde character and is used to destroy an instance of a class.

1. **Can A Class Have More Than 1 Destructor?**

No, a class can have only 1 destructor.

1. **Can Structs In C# Have Destructors?**

No, structs can have constructors but not destructors, only classes can have destructors.

1. **Can You Pass Parameters To Destructors?**

No, you cannot pass parameters to destructors. Hence, you cannot overload destructors.

1. **Can You Explicitly Call A Destructor?**

No, you cannot explicitly call a destructor. Destructors are invoked automatically by the garbage collector.

1. **Why Is It Not A Good Idea To Use Empty Destructors?**

When a class contains a destructor, an entry is created in the Finalize queue. When the destructor is called, the garbage collector is invoked to process the queue. If the destructor is empty, this just causes a needless loss of performance.

1. **Is It Possible To Force Garbage Collector To Run?**

Yes, it possible to force garbage collector to run by calling the Collect() method, but this is not considered a good practice because this might create a performance over head. Usually the programmer has no control over when the garbage collector runs. The garbage collector checks for objects that are no longer being used by the application. If it considers an object eligible for destruction, it calls the destructor(if there is one) and reclaims the memory used to store the object.

1. **Usually In .net, The Clr Takes Care Of Memory Management. Is There Any Need For A Programmer To Explicitly Release Memory And Resources? If Yes, Why And How?**

If the application is using expensive external resource, it is recommend to explicitly release the resource before the garbage collector runs and frees the object. We can do this by implementing the Dispose method from the IDisposable interface that performs the necessary cleanup for the object. This can considerably improve the performance of the application.

1. **When Do We Generally Use Destructors To Release Resources?**

If the application uses unmanaged resources such as windows, files, and network connections, we use  
destructors to release resources.

1. **What Is A Constructor In C#?**

Constructor is a class method that is executed when an object of a class is created. Constructor has the same name as the class, and usually used to initialize the data members of the new object.

1. **In C#, What Will Happen If You Do Not Explicitly Provide A Constructor For A Class?**

If you do not provide a constructor explicitly for your class, C# will create one by default that instantiates the object and sets all the member variables to their default values.

1. **Structs Are Not Reference Types. Can Structs Have Constructors?**

Yes, even though Structs are not reference types, structs can have constructors.

1. **We Cannot Create Instances Of Static Classes. Can We Have Constructors For Static Classes?**

Yes, static classes can also have constructors.

1. **Can You Prevent A Class From Being Instantiated?**

Yes, a class can be prevented from being instantiated by using a private constructor as shown in the example below.

using System;  
namespace TestConsole  
{  
class Program  
{  
public static void Main()  
{  
//Error cannot create instance of a class with private constructor  
SampleClass SC = new SampleClass();  
}  
}  
class SampleClass  
{  
double PI = 3.141;  
private SampleClass()  
{  
}  
}  
}

1. **Can A Class Or A Struct Have Multiple Constructors?**

Yes, a class or a struct can have multiple constructors. Constructors in csharp can be overloaded.

1. **Can A Child Class Call The Constructor Of A Base Class?**

Yes, a child class can call the constructor of a base class by using the base keyword as shown in the example below.

using System;  
namespace TestConsole  
{  
class BaseClass  
{  
public BaseClass(string str)  
{  
Console.WriteLine(str);  
}  
}  
class ChildClass : BaseClass  
{  
public ChildClass(string str): base(str)  
{  
}  
public static void Main()  
{  
ChildClass CC = new ChildClass("Calling base class constructor from child class");  
}  
}  
}

1. **If A Child Class Instance Is Created, Which Class Constructor Is Called First - Base Class Or Child Class?**

When an instance of a child class is created, the base class constructor is called before the child class constructor. An example is shown below.

using System;  
namespace TestConsole  
{  
class BaseClass  
{  
public BaseClass()  
{  
Console.WriteLine("I am a base class constructor");  
}  
}  
class ChildClass : BaseClass  
{  
public ChildClass()  
{  
Console.WriteLine("I am a child class constructor");  
}  
public static void Main()  
{  
ChildClass CC = new ChildClass();  
}  
}  
}

1. **Can A Class Have Static Constructor?**

Yes, a class can have static constructor. Static constructors are called automatically, immediately before any static fields are accessed, and are generally used to initialize static class members. It is called automatically before the first instance is created or any static members are referenced. Static constructors are called before instance constructors. An example is shown below.

using System;  
namespace TestConsole  
{  
class Program  
{  
static int I;  
static Program()  
{  
I = 100;  
Console.WriteLine("Static Constructor called");  
}  
public Program()  
{  
Console.WriteLine("Instance Constructor called");  
}  
public static void Main()  
{  
Program P = new Program();  
}  
}  
}

1. **Can You Mark Static Constructor With Access Modifiers?**

No, we cannot use access modifiers on static constructor.

1. **Can You Have Parameters For Static Constructors?**

No, static constructors cannot have parameters.

1. **What Happens If A Static Constructor Throws An Exception?**

If a static constructor throws an exception, the runtime will not invoke it a second time, and the type will remain uninitialized for the lifetime of the application domain in which your program is running.

1. **Give 2 Scenarios Where Static Constructors Can Be Used?**

1. A typical use of static constructors is when the class is using a log file and the constructor is used to write entries to this file.  
2. Static constructors are also useful when creating wrapper classes for unmanaged code, when the constructor can call the LoadLibrary method.

1. **Does C# Provide Copy Constructor?**

No, C# does not provide copy constructor.

1. **Is The Following Code Legal?**

using System;  
namespace Demo  
{  
class Program  
{  
public static void Main()  
{  
}  
public void Sum(int FirstNumber, int SecondNumber)  
{  
int Result = FirstNumber + SecondNumber;  
}  
public int Sum(int FirstNumber, int SecondNumber)  
{  
int Result = FirstNumber + SecondNumber;  
}  
}  
}

No, The above code does not compile. You cannot overload a method based on the return type. To overload a method in C# either the number or type of parameters should be different. In general the return type of a method is not part of the signature of the method for the purposes of method overloading. However, it is part of the signature of the method when determining the compatibility between a delegate and the method that it points to.

1. **What Is The Difference Between Method Parameters And Method Arguments. Give An Example?**

In the example below FirstNumber and SecondNumber are method parameters where as FN and LN are method arguments. The method definition specifies the names and types of any parameters that are required. When calling code calls the method, it provides concrete values called arguments for each parameter. The arguments must be compatible with the parameter type but the argument name (if any) used in the calling code does not have to be the same as the parameter named defined in the method.

using System;  
namespace Demo  
{  
class Program  
{  
public static void Main()  
{  
int FN = 10;  
int SN = 20;  
//FN and LN are method arguments  
int Total = Sum(FN, SN);  
Console.WriteLine(Total);  
}  
//FirstNumber and SecondNumber are method parameters  
public static int Sum(int FirstNumber, int SecondNumber)  
{  
int Result = FirstNumber + SecondNumber;  
return Result;  
}  
}  
}

1. **Explain The Difference Between Passing Parameters By Value And Passing Parameters By Reference With An Example?**

We can pass parameters to a method by value or by reference. By default all value types are passed by value where as all reference types are passed by reference. By default, when a value type is passed to a method, a copy is passed instead of the object itself. Therefore, changes to the argument have no effect on the original copy in the calling method.An example is shown below.

using System;  
namespace Demo  
{  
class Program  
{  
public static void Main()  
{  
int I = 10;  
int K = Function(I);  
Console.WriteLine("I = " + I);  
Console.WriteLine("K = " + K);  
}  
public static int Function(int Number)  
{  
int ChangedValue = Number + 1;  
return ChangedValue;  
}  
}  
}

By default, reference types are passed by reference. When an object of a reference type is passed to a method, the reference points to the original object, not a copy of the object. Changes made through this reference will therefore be reflected in the calling method. Reference types are created by using the class keyword as shown in the example below.

using System;  
namespace Demo  
{  
class Program  
{  
public static void Main()  
{  
ReferenceTypeExample Object = new ReferenceTypeExample();  
Object.Number = 20;  
Console.WriteLine("Original Object Value = " + Object.Number);  
Function(Object);  
Console.WriteLine("Object Value after passed to the method= " + Object.Number);  
}  
public static void Function(ReferenceTypeExample ReferenceTypeObject)  
{  
ReferenceTypeObject.Number = ReferenceTypeObject.Number + 5;  
}  
}  
class ReferenceTypeExample  
{  
public int Number;  
}  
}

1. **Can You Pass Value Types By Reference To A Method?**

Yes, we can pass value types by by reference to a method. An example is shown below.

using System;  
namespace Demo  
{  
class Program  
{  
public static void Main()  
{  
int I = 10;  
Console.WriteLine("Value of I before passing to the method = " + I);  
Function(ref I);  
Console.WriteLine("Value of I after passing to the method by reference= " + I);  
}  
public static void Function(ref int Number)  
{  
Number = Number + 5;  
}  
}  
}

1. **If A Method's Return Type Is Void, Can You Use A Return Keyword In The Method?**

Yes, Even though a method's return type is void, you can use the return keyword to stop the execution of the method as shown in the example below.

using System;  
namespace Demo  
{  
class Program  
{  
public static void Main()  
{  
SayHi();  
}  
public static void SayHi()  
{  
Console.WriteLine("Hi");  
return;  
Console.WriteLine("This statement will never be executed");  
}  
}  
}

1. **What Are Properties In C#. Explain With An Example?**

Properties in C# are class members that provide a flexible mechanism to read, write, or compute the values of private fields. Properties can be used as if they are public data members, but they are actually special methods called accessors. This enables data to be accessed easily and still helps promote the safety and flexibility of methods.

In the example below \_firstName and \_lastName are private string variables which are accessible only inside the Customer class. \_firstName and \_lastName are exposed using FirstName and LastName public properties respectively. The get property accessor is used to return the property value, and a set accessor is used to assign a new value. These accessors can have different access levels. The value keyword is used to define the value being assigned by the set accessor. The FullName property computes the full name of the customer. Full Name property is readonly, because it has only the get accessor. Properties that do not implement a set accessor are read only.

The code block for the get accessor is executed when the property is read and the code block for the set accessor is executed when the property is assigned a new value.

using System;  
class Customer  
{  
// Private fileds not accessible outside the class.  
private string \_firstName = string.Empty;  
private string \_lastName = string.Empty;  
private string \_coutry = string.Empty;  
// public FirstName property exposes \_firstName variable  
public string FirstName  
{  
get  
{  
return \_firstName;  
}  
set  
{  
\_firstName = value;  
}  
}  
// public LastName property exposes \_lastName variable  
public string LastName  
{  
get  
{  
return \_lastName;  
}  
set  
{  
\_lastName = value;  
}  
}  
// FullName property is readonly and computes customer full name.  
public string FullName  
{  
get  
{  
return \_lastName + ", " + \_firstName;  
}  
}  
//Country Property is Write Only  
public string Country  
{  
set  
{  
\_coutry = value;  
}  
}  
}  
class MainClass  
{  
public static void Main()  
{  
Customer CustomerObject = new Customer();  
//This line will call the set accessor of FirstName Property  
CustomerObject.FirstName = "David";  
//This line will call the set accessor of LastName Property  
CustomerObject.LastName = "Boon";  
//This line will call the get accessor of FullName Property  
Console.WriteLine("Customer Full Name is : " + CustomerObject.FullName);  
}  
}

1. **Explain The 3 Types Of Properties In C# With An Example?**

1. Read Only Properties: Properties without a set accessor are considered read-only. In the above example FullName is read only property.  
2. Write Only Properties: Properties without a get accessor are considered write-only. In the above example Country is write only property.  
3. Read Write Properties: Properties with both a get and set accessor are considered read-write properties. In the above example FirstName and LastName are read write properties.

1. **What Are The Advantages Of Properties In C#?**

1. Properties can validate data before allowing a change.  
2. Properties can transparently expose data on a class where that data is actually retrieved from some other source such as a database.  
3. Properties can take an action when data is changed, such as raising an event or changing the value of other fields.

1. **What Is A Static Property. Give An Example?**

A property that is marked with a static keyword is considered as static property. This makes the property available to callers at any time, even if no instance of the class exists. In the example below PI is a static property.

using System;  
class Circle  
{  
private static double \_pi = 3.14;  
public static double PI  
{  
get  
{  
return \_pi;  
}  
}  
}  
class MainClass  
{  
public static void Main()  
{  
Console.WriteLine(Circle.PI);  
}  
}

1. **What Is A Virtual Property. Give An Example?**

A property that is marked with virtual keyword is considered virtual property. Virtual properties enable derived classes to override the property behavior by using the override keyword. In the example below FullName is virtual property in the Customer class. BankCustomer class inherits from Customer class and overrides the FullName virtual property. In the output you can see the over riden implementation. A property overriding a virtual property can also be sealed, specifying that for derived classes it is no longer virtual.

using System;  
class Customer  
{  
private string \_firstName = string.Empty;  
private string \_lastName = string.Empty;  
public string FirstName  
{  
get  
{  
return \_firstName;  
}  
set  
{  
\_firstName = value;  
}  
}  
public string LastName  
{  
get  
{  
return \_lastName;  
}  
set  
{  
\_lastName = value;  
}  
}  
// FullName is virtual  
public virtual string FullName  
{  
get  
{ return \_lastName + ", " + \_firstName;  
}  
}  
}  
class BankCustomer : Customer  
{  
// Overiding the FullName virtual property derived from customer class  
public override string FullName  
{  
get  
{  
return "Mr. " + FirstName + " " + LastName;  
}  
}  
}  
class MainClass  
{  
public static void Main()  
{  
BankCustomer BankCustomerObject = new BankCustomer();  
BankCustomerObject.FirstName = "David";  
BankCustomerObject.LastName = "Boon";  
Console.WriteLine("Customer Full Name is : " + BankCustomerObject.FullName);  
}  
}

1. **What Is An Abstract Property. Give An Example?**

A property that is marked with abstract keyword is considered abstract property. An abstract property should not have any implementation in the class. The derived classes must write their own implementation. In the example below FullName property is abstract in the Customer class. BankCustomer class overrides the inherited abstract FullName property with its own implementation.

using System;  
abstract class Customer  
{  
private string \_firstName = string.Empty;  
private string \_lastName = string.Empty;  
public string FirstName  
{  
get  
{  
return \_firstName;  
}  
set  
{  
\_firstName = value;  
}  
}  
public string LastName  
{  
get  
{  
return \_lastName;  
}  
set  
{  
\_lastName = value;  
}  
}  
// FullName is abstract  
public abstract string FullName  
{  
get;  
}  
}  
class BankCustomer : Customer  
{  
// Overiding the FullName abstract property derived from customer class  
public override string FullName  
{  
get  
{  
return "Mr. " + FirstName + " " + LastName;  
}  
}  
}  
class MainClass  
{  
public static void Main()  
{  
BankCustomer BankCustomerObject = new BankCustomer();  
BankCustomerObject.FirstName = "David";  
BankCustomerObject.LastName = "Boon";  
Console.WriteLine("Customer Full Name is : " + BankCustomerObject.FullName);  
}  
}

1. **Can You Use Virtual, Override Or Abstract Keywords On An Accessor Of A Static Property?**

No, it is a compile time error to use a virtual, abstract or override keywords on an accessor of a static property.

1. **What Are Constants In C#?**

Constants in C# are immutable values which are known at compile time and do not change for the life of the program. Constants are declared using the const keyword. Constants must be initialized as they are declared. You cannot assign a value to a constant after it isdeclared. An example is shown below.

using System;  
class Circle  
{  
public const double PI = 3.14;  
public Circle()  
{  
//Error : You can only assign a value to a constant field at the time of declaration  
//PI = 3.15;  
}  
}  
class MainClass  
{  
public static void Main()  
{  
Console.WriteLine(Circle.PI);  
}  
}

1. **Can You Declare A Class Or A Struct As Constant?**

No, User-defined types including classes, structs, and arrays, cannot be const. Only the C# built-in types excluding System.Object may be declared as const. Use the readonly modifier to create a class, struct, or array that is initialized one time at runtime (for example in a constructor) and thereafter cannot be changed.

1. **Does C# Support Const Methods, Properties, Or Events?**

No, C# does not support const methods, properties, or events.

1. **Can You Change The Value Of A Constant Filed After Its Declaration?**

No, you cannot change the value of a constant filed after its declaration. In the example below, the constant field PI is always 3.14, and it cannot be changed even by the class itself. In fact, when the compiler encounters a constant identifier in C# source code (for example, PI), it substitutes the literal value directly into the intermediate language (IL) code that it produces. Because there is no variable address associated with a constant at run time, const fields cannot be passed by reference.

using System;  
class Circle  
{  
public const double PI = 3.14;  
}

1. **How Do You Access A Constant Field Declared In A Class?**

Constants are accessed as if they were static fields because the value of the constant is the same for all instances of the type. You do not use the static keyword to declare them. Expressions that are not in the class that defines the constant must use the class name, a period, and the name of the constant to access the constant. In the example below constant field PI can be accessed in the Main method using the class name and not the instance of the class. Trying to access a constant field using a class instance will generate a compile time error.

using System;  
class Circle  
{  
public const double PI = 3.14;  
}  
class MainClass  
{  
public static void Main()  
{  
Console.WriteLine(Circle.PI);  
Circle C = new Circle();  
// Error : PI cannot be accessed using an instance  
// Console.WriteLine(C.PI);  
}  
}

1. **What Are The 2 Broad Classifications Of Fields In C#?**

1. Instance fields  
2. Static fields

1. **What Are Instance Fields In C#?**

Instance fields are specific to an instance of a type. If you have a class T, with an instance field F, you can create two objects of type T, and modify the value of F in each object without affecting the value in the other object.

1. **What Is A Static Field?**

A static field belongs to the class itself, and is shared among all instances of that class. Changes made from instance A will be visible immediately to instances B and C if they access the field.

1. **Can You Declare A Field Readonly?**

Yes, a field can be declared readonly. A read-only field can only be assigned a value during initialization or in a constructor. An example is shown below.

using System;  
class Area  
{  
public readonly double PI = 3.14;  
}  
class MainClass  
{  
public static void Main()  
{  
Area A = new Area();  
Console.WriteLine(A.PI);  
}  
}

1. **What Is Wrong With The Sample Program Below?**

using System;  
class Area  
{  
public const double PI = 3.14;  
static Area()  
{  
Area.PI = 3.15;  
}  
}  
class MainClass  
{  
public static void Main()  
{  
Console.WriteLine(Area.PI);  
}  
}  
You cannot assign a value to the constant PI field.

1. **What Is The Difference Between A Constant And A Static Readonly Field?**

A static readonly field is very similar to a constant, except that the C# compiler does not have access to the value of a static read-only field at compile time, only at run time.

1. **What Are Access Modifiers In C#?**

In C# there are 5 different types of Access Modifiers.  
Public  
The public type or member can be accessed by any other code in the same assembly or another assembly that references it.

Private  
The type or member can only be accessed by code in the same class or struct.

Protected  
The type or member can only be accessed by code in the same class or struct, or in a derived class.

Internal  
The type or member can be accessed by any code in the same assembly, but not from another assembly.

Protected Internal  
The type or member can be accessed by any code in the same assembly, or by any derived class in another assembly.

1. **What Are Access Modifiers Used For?**

Access Modifiers are used to control the accessibilty of types and members with in the types.

1. **Can You Use All Access Modifiers For All Types?**

No, Not all access modifiers can be used by all types or members in all contexts, and in some cases the accessibility of a type member is constrained by the accessibility of its containing type.

1. **Can Derived Classes Have Greater Accessibility Than Their Base Types?**

No, Derived classes cannot have greater accessibility than their base types. For example the following code is illegal. using System;  
internal class InternalBaseClass  
{  
public void Print()  
{  
Console.WriteLine("I am a Base Class Method");  
}  
}  
public class PublicDerivedClass : InternalBaseClass  
{  
public static void Main()  
{  
Console.WriteLine("I am a Public Derived Class Method");  
}  
}

When you compile the above code an error will be generated stating "Inconsistent accessibility: base class InternalBaseClass is less accessible than class PublicDerivedClass".To make this simple, you cannot have a public class B that derives from an internal class A. If this were allowed, it would have the effect of making A public, because all protected or internal members of A are accessible from the derived class.

1. **Can You Declare Struct Members As Protected?**

No, struct members cannot be declared protected. This is because structs do not support inheritance.

1. **Can The Accessibility Of A Type Member Be Greater Than The Accessibility Of Its Containing Type?**

No, the accessibility of a type member can never be greater than the accessibility of its containing type. For example, a public method declared in an internal class has only internal accessibility.

1. **Can Destructors Have Access Modifiers?**

No, destructors cannot have access modifiers.

1. **What Does Protected Internal Access Modifier Mean?**

The protected internal access means protected OR internal, not protected AND internal. In simple terms, a protected internal member is accessible from any class in the same assembly, including derived classes. To limit accessibility to only derived classes in the same assembly, declare the class itself internal, and declare its members as protected.

1. **What Is The Default Access Modifier For A Class,struct And An Interface Declared Directly With A Namespace?**

internal.

1. **Can You Specify An Access Modifier For An Enumeration?**

Enumeration members are always public, and no access modifiers can be specified.

1. **What Are The 3 Types Of Comments In C#?**

1. Single Line Comments. You define single line comments with // as shown below.  
//This is an example for single line comment  
2. Multi line comments. You define multi line comments with /\* \*/ as shown below.  
/\*This is an example for Multi Line comments\*/  
3. XML Comments. You define XML comments with /// as shown below. ///This is an example for defining XML comments.

1. **Is C# A Strongly-typed Language?**

Yes.

1. **What Are The 2 Broad Classifications Of Data Types Available In C#?**

1. Built in data types.  
2. User defined data types.

1. **Give Some Examples For Built In Datatypes In C#?**

1. int  
2. float  
3. bool

1. **How Do You Create User Defined Data Types In C#?**

You use the struct, class, interface, and enum constructs to create your own custom types. The .NET Framework class library itself is a collection of custom types provided by Microsoft that you can use in your own applications.

1. **What Are The 2 Types Of Data Types Available In C#?**

1. Value Types  
2. Reference Types

1. **If You Define A User Defined Data Type By Using The Struct Keyword, Is It A Value Type Or Reference Type?**

Value Type.

1. **If You Define A User Defined Data Type By Using The Class Keyword, Is It A Value Type Or Reference Type?**

Reference type

1. **Are Value Types Sealed?**

Yes, Value types are sealed.

1. **What Is The Base Class From Which All Value Types Are Derived?**

System.ValueType.

1. **Give Examples For Value Types?**

Enum  
Struct

1. **Give Examples For Reference Types?**

Class  
Delegate  
Array  
Interface.

1. **What Are The Differences Between Value Types And Reference Types?**

1. Value types are stored on the stack where as reference types are stored on the managed heap.  
2. Value type variables directly contain their values where as reference variables holds only a reference to the location of the object that is created on the managed heap.  
3. There is no heap allocation or garbage collection overhead for value-type variables. As reference types are stored on the managed heap, they have the over head of object allocation and garbage collection.  
4. Value Types cannot inherit from another class or struct. Value types can only inherit from interfaces. Reference types can inherit from another class or interface.

1. **What Do You Mean By Casting A Data Type?**

Converting a variable of one data type to another data type is called casting. This is also called as data type conversion.

1. **What Are The 2 Kinds Of Data Type Conversions In C#?**

Implicit conversions: No special syntax is required because the conversion is type safe and no data will be lost. Examples include conversions from smaller to larger integral types, and conversions from derived classes to base classes.

Explicit conversions: Explicit conversions require a cast operator. The source and destination variables are compatible, but there is a risk of data loss because the type of the destination variable is a smaller size than (or is a base class of) the source variable.

1. **What Is The Difference Between An Implicit Conversion And An Explicit Conversion?**

1. Explicit conversions require a cast operator where as an implicit converstion is done automatically.  
2. Explicit conversion can lead to data loss where as with implicit conversions there is no data loss.

1. **What Type Of Data Type Conversion Happens When The Compiler Encounters The Following Code?**

ChildClass CC = new ChildClass();  
ParentClass PC = new ParentClass();

Implicit Conversion. For reference types, an implicit conversion always exists from a class to any one of its direct or indirect base classes or interfaces. No special syntax is necessary because a derived class always contains all the members of a base class.

1. **If You Want To Convert A Base Type To A Derived Type, What Type Of Conversion Do You Use?**

Explicit conversion as shown below.  
//Create a new derived type.  
Car C1 = new Car();  
// Implicit conversion to base type is safe.  
Vehicle V = C1;

// Explicit conversion is required to cast back to derived type. The code below will compile but throw an exception at run time if the right-side object is not a Car object.  
Car C2 = (Car) V;

1. **What Operators Can Be Used To Cast From One Reference Type To Another Without The Risk Of Throwing An Exception?**

The is and as operators can be used to cast from one reference type to another without the risk of throwing an exception.

1. **If Casting Fails What Type Of Exception Is Thrown?**

InvalidCastException.

1. **What Is Boxing And Unboxing?**

Boxing - Converting a value type to reference type is called boxing. An example is shown below.  
int i = 101;  
object obj = (object)i; // Boxing

Unboxing - Converting a reference type to a value typpe is called unboxing. An example is shown below.  
obj = 101;  
i = (int)obj; // Unboxing

1. **Is Boxing An Implicit Conversion?**

Yes, boxing happens implicitly.

1. **Is Unboxing An Implicit Conversion?**

No, unboxing is an explicit conversion.

1. **What Happens During The Process Of Boxing?**

Boxing is used to store value types in the garbage-collected heap. Boxing is an implicit conversion of a value type to the type object or to any interface type implemented by this value type. Boxing a value type allocates an object instance on the heap and copies the value into the new object. Due to this boxing and unboxing can have performance impact.

1. **What Is An Array?**

An array is a data structure that contains several variables of the same type.

1. **What Are The 3 Different Types Of Arrays?**

1. Single-Dimensional  
2. Multidimensional  
3. Jagged

1. **What Is Jagged Array?**

A jagged array is an array of arrays.

1. **Are Arrays Value Types Or Reference Types?**

Arrays are reference types.

1. **What Is The Base Class For Array Types?**

System.Array.

1. **Can You Use Foreach Iteration On Arrays In C#?**

Yes,Since array type implements IEnumerable, you can use foreach iteration on all arrays in C#.

1. **What Is The Difference Between String Keyword And System.string Class?**

string keyword is an alias for Syste.String class. Therefore, System.String and string keyword are the same, and you can use whichever naming convention you prefer. The String class provides many methods for safely creating, manipulating, and comparing strings.

1. **Are String Objects Mutable Or Immutable?**

String objects are immutable.

1. **What Do You Mean By String Objects Are Immutable?**

String objects are immutable means, they cannot be changed after they have been created. All of the String methods and C# operators that appear to modify a string actually return the results in a new string object. In the following example, when the contents of s1 and s2 are concatenated to form a single string, the two original strings are unmodified. The += operator creates a new string that contains the combined contents. That new object is assigned to the variable s1, and the original object that was assigned to s1 is released for garbage collection because no other variable holds a reference to it.

string s1 = "First String ";  
string s2 = "Second String";

// Concatenate s1 and s2. This actually creates a new  
// string object and stores it in s1, releasing the  
// reference to the original object.  
s1 += s2;

System.Console.WriteLine(s1);  
// Output: First String Second String.

1. **What Will Be The Output Of The Following Code?**

string str1 = "Hello ";  
string str2 = s1;  
str1 = str1 + "C#";  
System.Console.WriteLine(s2);

The output of the above code is "Hello" and not "Hello C#". This is bcos, if you create a reference to a string, and then "modify" the original string, the reference will continue to point to the original object instead of the new object that was created when the string was modified.

1. **What Is A Verbatim String Literal And Why Do We Use It?**

The "@" symbol is the verbatim string literal. Use verbatim strings for convenience and better readability when the string text contains backslash characters, for example in file paths. Because verbatim strings preserve new line characters as part of the string text, they can be used to initialize multiline strings. Use double quotation marks to embed a quotation mark inside a verbatim string. The following example shows some common uses for verbatim strings:

string ImagePath = @"C:\Images\Buttons\SaveButton.jpg";  
//Output: C:\Images\Buttons\SaveButton.jpg

string MultiLineText = @"This is multiline Text written to be in three lines.";  
/\* Output:  
This is multiline Text written to be in three lines.  
\*/

string DoubleQuotesString = @"My Name is ""Vankat.""";  
//Output: My Name is "Vankat."

1. **Will The Following Code Compile And Run?**

string str = null;  
Console.WriteLine(str.Length);  
The above code will compile, but at runtime System.NullReferenceException will be thrown

1. **How Do You Create Empty Strings In C#?**

Using string.empty as shown in the example below.  
string EmptyString = string.empty;

1. **What Is The Difference Between System.text.stringbuilder And System.string?**

1. Objects of type StringBuilder are mutable where as objects of type System.String are immutable.  
2. As StringBuilder objects are mutable, they offer better performance than string objects of type System.String  
3. StringBuilder class is present in System.Text namespace where String class is present in System namespace.

1. **How Do You Determine Whether A String Represents A Numeric Value?**

To determine whether a String represents a numeric value use TryParse method as shown in the example below. If the string contains nonnumeric characters or the numeric value is too large or too small for the particular type you have specified, TryParse returns false and sets the out parameter to zero. Otherwise, it returns true and sets the out parameter to the numeric value of the string.

string str = "One";  
int i = 0;  
if(int.TryParse(str,out i))  
{  
Console.WriteLine("Yes string contains Integer and it is " + i);  
}  
else  
{  
Console.WriteLine("string does not contain Integer");  
}

1. **What Is The Difference Between Int.parse And Int.tryparse Methods?**

Parse method throws an exception if the string you are trying to parse is not a valid number where as TryParse returns false and does not throw an exception if parsing fails. Hence TryParse is more efficient than Parse.

1. **Why Should You Override The Tostring() Method?**

All types in .Net inherit from system.object directly or indirectly. Because of this inheritance, every type in .Net inherit the ToString() method from System.Object class. Consider the example below.

using System;  
public class MainClass  
{  
public static void Main()  
{  
int Number = 10;  
Console.WriteLine(Number.ToString());  
}  
}

In the above example Number.ToString() method will correctly give the string representaion of int 10, when you call the ToString() method.

If you have a Customer class as shown in the below example and when you call the ToString() method the output doesnot make any sense. Hence you have to override the ToString() method, that is inherited from the System.Object class.

using System;  
public class Customer  
{  
public string FirstName;  
public string LastName;  
}  
public class MainClass  
{  
public static void Main()  
{  
Customer C = new Customer();  
C.FirstName = "David";  
C.LastName = "Boon";  
Console.WriteLine(C.ToString());  
}  
}

The code sample below shows how to override the ToString() method in a class, that would give the output you want.

using System;  
public class Customer  
{  
public string FirstName;  
public string LastName;  
public override string ToString()  
{  
return LastName + ", " + FirstName;  
}  
}  
public class MainClass  
{  
public static void Main()  
{  
Customer C = new Customer();  
C.FirstName = "David";  
C.LastName = "Boon";  
Console.WriteLine(C.ToString());  
}  
}

Conclusion : If you have a class or a struct, make sure you override the inherited ToString() method.

1. **Explain Polymorphism In C# With A Simple Example?**

Polymorphism allows you to invoke derived class methods through a base class reference during run-time. An example is shown below.  
using System;  
public class DrawingObject  
{  
public virtual void Draw()  
{  
Console.WriteLine("I am a drawing object.");  
}  
}  
public class Triangle : DrawingObject  
{  
public override void Draw()  
{  
Console.WriteLine("I am a Triangle.");  
}  
}  
public class Circle : DrawingObject  
{  
public override void Draw()  
{  
Console.WriteLine("I am a Circle.");  
}  
}  
public class Rectangle : DrawingObject  
{  
public override void Draw()  
{  
Console.WriteLine("I am a Rectangle.");  
}  
}  
public class DrawDemo  
{  
public static void Main()  
{  
DrawingObject[] DrawObj = new DrawingObject[4];  
DrawObj[0] = new Triangle();  
DrawObj[1] = new Circle();  
DrawObj[2] = new Rectangle();  
DrawObj[3] = new DrawingObject();  
foreach (DrawingObject drawObj in DrawObj)  
{  
drawObj.Draw();  
}  
}  
}

1. **When Can A Derived Class Override A Base Class Member?**

A derived class can override a base class member only if the base class member is declared as virtual or abstract.

1. **What Is The Difference Between A Virtual Method And An Abstract Method?**

A virtual method must have a body where as an abstract method should not have a body.

1. **Can Fields Inside A Class Be Virtual?**

No, Fields inside a class cannot be virtua. Only methods, properties, events and indexers can be virtual.

1. **Give An Example To Show For Hiding Base Class Methods?**

Use the new keyword to hide a base class method in the derived class as shown in the example below.  
using System;  
public class BaseClass  
{  
public virtual void Method()  
{  
Console.WriteLine("I am a base class method.");  
}  
}  
public class DerivedClass : BaseClass  
{  
public new void Method()  
{  
Console.WriteLine("I am a child class method.");  
}  
public static void Main()  
{  
DerivedClass DC = new DerivedClass();  
DC.Method();  
}  
}

1. **Can You Access A Hidden Base Class Method In The Derived Class?**

Yes, Hidden base class methods can be accessed from the derived class by casting the instance of the derived class to an instance of the base class as shown in the example below.  
using System;  
public class BaseClass  
{  
public virtual void Method()  
{  
Console.WriteLine("I am a base class method.");  
}  
}  
public class DerivedClass : BaseClass  
{  
public new void Method()  
{  
Console.WriteLine("I am a child class method.");  
}  
public static void Main()  
{  
DerivedClass DC = new DerivedClass();  
((BaseClass)DC).Method();  
}  
}

1. **What Is An Abstract Class?**

An abstract class is an incomplete class and must be implemented in a derived class.

1. **Can You Create An Instance Of An Abstract Class?**

No, abstract classes are incomplete and you cannot create an instance of an abstract class.

1. **What Is A Sealed Class?**

A sealed class is a class that cannot be inherited from. This means, If you have a class called Customer that is marked as sealed. No other class can inherit from Customer class. For example, the below code generates a compile time error "MainClass cannot derive from sealed type Customer. using System;

public sealed class Customer

{

}

public class MainClass : Customer

{

public static void Main()

{

}

}

1. **What Are Abstract Methods?**

Abstract methods are methods that only the declaration of the method and no implementation.

1. **How Can You Force Derived Classes To Provide New Method Implementations For Virtual Methods?**

Abstract classes can be used to force derived classes to provide new method implementations for virtual methods. An example is shown below.  
public class BaseClass  
{  
public virtual void Method()  
{  
// Original Implementation.  
}  
}  
public abstract class AbstractClass : BaseClass  
{  
public abstract override void Method();  
}  
public class NonAbstractChildClass : AbstractClass  
{  
public override void Method()  
{  
// New implementation.  
}  
}

When an abstract class inherits a virtual method from a base class, the abstract class can override the virtual method with an abstract method. If a virtual method is declared abstract, it is still virtual to any class inheriting from the abstract class. A class inheriting an abstract method cannot access the original implementation of the method. In the above example, Method() on class NonAbstractChildClass cannot call Method() on class BaseClass. In this way, an abstract class can force derived classes to provide new method implementations for virtual methods.

1. **Can A Sealed Class Be Used As A Base Class?**

No, sealed class cannot be used as a base class. A compile time error will be generated.

1. **What Are The 4 Pillars Of Any Object Oriented Programming Language?**

1. Abstraction  
2. Inheritance  
3. Encapsulation  
4. Polymorphism

1. **Do Structs Support Inheritance?**

No, structs do not support inheritance, but they can implement interfaces.

1. **What Is The Main Advantage Of Using Inheritance?**

Code reuse.

1. **Does C# Support Multiple Class Inheritance?**

No, C# supports single class inheritance only. However classes can implement multiple interfaces at the same time.

1. **Can A Struct Have A Default Constructor (a Constructor Without Parameters) Or A Destructor In C#?**

No.

1. **Can You Instantiate A Struct Without Using A New Operator In C#?**

Yes, you can instantiate a struct without using a new operator.

1. **Can A Struct Inherit From Another Struct Or Class In C#?**

No, a struct cannot inherit from another struct or class, and it cannot be the base of a class.

1. **Can A Struct Inherit From An Interface In C#?**

Yes.

1. **Are Structs Value Types Or Reference Types?**

Structs are value types.

1. **What Is The Base Type From Which All Structs Inherit Directly?**

All structs inherit directly from System.ValueType, which inherits from System.Object.

1. **What Do You Mean By Saying A "class Is A Reference Type"?**

A class is a reference type means when an object of the class is created, the variable to which the object is assigned holds only a reference to that memory. When the object reference is assigned to a new variable, the new variable refers to the original object. Changes made through one variable are reflected in the other variable because they both refer to the same data.

1. **What Do You Mean By Saying A "struct Is A Value Type"?**

A struct is a value type mean when a struct is created, the variable to which the struct is assigned holds the struct's actual data. When the struct is assigned to a new variable, it is copied. The new variable and the original variable therefore contain two separate copies of the same data. Changes made to one copy do not affect the other copy.

1. **When Do You Generally Use A Class Over A Struct?**

A class is used to model more complex behavior, or data that is intended to be modified after a class object is created. A struct is best suited for small data structures that contain primarily data that is not intended to be modified after the struct is created.

1. **List The 5 Different Access Modifiers In C#?**

1. public  
2. protected  
3. internal  
4. protected internal  
5. private

1. **If You Donot Specify An Access Modifier For A Method, What Is The Default Access Modifier?**

private.

1. **Classes And Structs Support Inheritance. Is This Statement True Or False?**

False, Only classes support inheritance. structs donot support inheritance.

1. **If A Class Derives From Another Class, Will The Derived Class Automatically Contain All The Public, Protected, And Internal Members Of The Base Class?**

Yes, the derived class will automatically contain all the public, protected, and internal members of the base class except its constructors and destructors.

1. **Can You Create An Instance For An Abstract Class?**

No, you cannot create an instance for an abstract class.

1. **How Do You Prevent A Class From Being Inherited By Another Class?**

Use the sealed keyword to prevent a class from being inherited by another class.

1. **Classes And Structs Can Be Declared As Static, Is This Statement True Or False?**

False, only classes can be declared as static and not structs.

1. **Can You Create An Instance Of A Static Class?**

No, you cannot create an instance of a static class.

1. **Can A Static Class Contain Non Static Members?**

No, a static class can contain only static members.

1. **Does C# Support Multiple-inheritance?**

No, but you can implement more than one interfaces.

1. **Who Is A Protected Class-level Variable Available To?**

It is available to any sub-class (a class inheriting this class).

1. **Are Private Class-level Variables Inherited?**

Yes, but they are not accessible. Although they are not visible or accessible via the class interface, they are inherited.

1. **Describe The Accessibility Modifier "protected Internal".**

It is available to classes that are within the same assembly and derived from the specified base class.

1. **What's The Top .net Class That Everything Is Derived From?**

System.Object.

1. **What Does The Term Immutable Mean?**

The data value may not be changed. Note: The variable value may be changed, but the original immutable data value was discarded and a new data value was created in memory.

1. **What's The Difference Between System.string And System.text.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.

1. **What's The .net Collection Class That Allows An Element To Be Accessed Using A Unique Key?**

HashTable, Dictionary, NameValueCollection.

1. **What Class Is Underneath The Sortedlist Class?**

A sorted HashTable.

1. **Will The Finally Block Get Executed If An Exception Has Not Occurred?**

Yes. Finally block always get executed.

1. **What's The C# Syntax To Catch Any Possible Exception?**

A catch block that catches the exception of type System.Exception. You can also omit the parameter data type in this case and just write catch {}.

1. **Can Multiple Catch Blocks Be Executed For A Single Try Statement?**

No. Once the proper catch block processed, control is transferred to the finally block (if there are any).

1. **Explain The Three Services Model Commonly Know As A Three-tier Application.**

Presentation (UI), Business (logic and underlying code) and Data (from storage or other sources).

1. **If A.equals(b) Is True Then A.gethashcode & B.gethashcode Must Always Return Same Hash Code.**

The answer is False because it is given that A.equals(B) returns true i.e. objects are equal and now its hash Code is asked which is always independent of the fact that whether objects are equal or not. So, Get HashCode for both of the objects returns different value.

1. **What Is The Syntax To Inherit From A Class In C#?**

Place a colon and then the name of the base class.  
Example: class MyNewClass : MyBaseClass.

1. **Can You Prevent Your Class From Being Inherited By Another Class?**

Yes. The keyword “sealed” will prevent the class from being inherited.

1. **Can You Allow A Class To Be Inherited, But Prevent The Method From Being Over-ridden?**

Yes. Just leave the class public and make the method sealed.

1. **What's An Abstract Class?**

A class that cannot be instantiated. An abstract class is a class that must be inherited and have the methods overridden. An abstract class is essentially a blueprint for a class without any implementation.

1. **When Do You Absolutely Have To Declare A Class As Abstract?**

1. When the class itself is inherited from an abstract class, but not all base abstract methods have been overridden.  
2. When at least one of the methods in the class is abstract.

1. **What Is An Interface Class?**

Interfaces, like classes, define a set of properties, methods, and events. But unlike classes, interfaces do not provide implementation. They are implemented by classes, and defined as separate entities from classes.

1. **Why Can't You Specify The Accessibility Modifier For Methods Inside The Interface?**

They all must be public, and are therefore public by default.

1. **Can You Inherit Multiple Interfaces?**

Yes. .NET does support multiple interfaces.

1. **What Happens If You Inherit Multiple Interfaces And They Have Conflicting Method Names?**

It’s up to you to implement the method inside your own class, so implementation is left entirely up to you. This might cause a problem on a higher-level scale if similarly named methods from different interfaces expect different data, but as far as compiler cares you’re okay.  
To Do: Investigate.

1. **What's The Difference Between An Interface And Abstract Class?**

In an interface class, all methods are abstract - there is no implementation. In an abstract class some methods can be concrete. In an interface class, no accessibility modifiers are allowed. An abstract class may have accessibility modifiers.

1. **What Is The Difference Between A Struct And A Class?**

Structs are value-type variables and are thus saved on the stack, additional overhead but faster retrieval. Another difference is that structs cannot inherit.

1. **What's The Implicit Name Of The Parameter That Gets Passed Into The Set Method/property Of A Class?**

Value. The data type of the value parameter is defined by whatever data type the property is declared as.

1. **What Does The Keyword "virtual" Declare For A Method Or Property?**

The method or property can be overridden.

1. **How Is Method Overriding Different From Method Overloading?**

When overriding a method, you change the behavior of the method for the derived class. Overloading a method simply involves having another method with the same name within the class.

1. **Can You Declare An Override Method To Be Static If The Original Method Is Not Static?**

No. The signature of the virtual method must remain the same. (Note: Only the keyword virtual is changed to keyword override)

1. **What Are The Different Ways A Method Can Be Overloaded?**

Different parameter data types, different number of parameters, different order of parameters.

1. **If A Base Class Has A Number Of Overloaded Constructors, And An Inheriting Class Has A Number Of Overloaded Constructors; Can You Enforce A Call From An Inherited Constructor To A Specific Base Constructor?**

Yes, just place a colon, and then keyword base (parameter list to invoke the appropriate constructor) in the overloaded constructor definition inside the inherited class.

1. **What Does Assert() Method Do?**

In debug compilation, assert takes in a Boolean condition as a parameter, and shows the error dialog if the condition is false. The program proceeds without any interruption if the condition is true.

1. **What's The Difference Between The Debug Class And Trace Class?**

Documentation looks the same. Use Debug class for debug builds, use Trace class for both debug and release builds.

1. **What Is The Role Of The Datareader Class In Ado.net Connections?**

It returns a read-only, forward-only rowset from the data source. A DataReader provides fast access when a forward-only sequential read is needed.

1. **What Is The Wildcard Character In Sql?**

Let’s say you want to query database with LIKE for all employees whose name starts with La. The wildcard character is %, the proper query with LIKE would involve ‘La%’.

1. **Between Windows Authentication And Sql Server Authentication, Which One Is Trusted And Which One Is Untrusted?**

Windows Authentication is trusted because the username and password are checked with the Active Directory, the SQL Server authentication is untrusted, since SQL Server is the only verifier participating in the transaction.

1. **What Does The Dispose Method Do With The Connection Object?**

Dispose places the connection backing the managed pool. So that other objects/class can use the connection for further use.

1. **How Is The Dll Hell Problem Solved In .net?**

Assembly versioning allows the application to specify not only the library it needs to run (which was available under Win32), but also the version of the assembly.

1. **What Is A Satellite Assembly?**

When you write a multilingual or multi-cultural application in .NET, and want to distribute the core application separately from the localized modules, the localized assemblies that modify the core application are called satellite assemblies.

1. **What Is The Smallest Unit Of Execution In .net?**

an Assembly.

1. **When Should You Call The Garbage Collector In .net?**

As a good rule, you should not call the garbage collector. However, you could call the garbage collector when you are done using a large object (or set of objects) to force the garbage collector to dispose of those very large objects from memory. However, this is usually not a good practice.

1. **How Do You Convert A Value-type To A Reference-type?**

Use Boxing.

1. **What Happens In Memory When You Box And Unbox A Value-type?**

Boxing converts a value-type to a reference-type, thus storing the object on the heap. Unboxing converts a reference-type to a value-type, thus storing the value on the stack.

1. **What's C# ?**

C# (pronounced C-sharp) is a new object oriented language from Microsoft and is derived from C and C++. It also borrows a lot of concepts from Java too including garbage collection.

1. **Is It Possible To Inline Assembly Or Il In C# Code?**

No.

1. **Is It Possible To Have Different Access Modifiers On The Get/set Methods Of A Property?**

No. The access modifier on a property applies to both its get and set accessors. What you need to do if you want them to be different is make the property read-only (by only providing a get accessor) and create a private/internal set method that is separate from the property.

1. **Is It Possible To Have A Static Indexer In C#? Allowed In C#.**

No. Static indexers are not

1. **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).

1. **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) { }

1. **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

1. **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.

1. **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.

1. **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);  
}

1. **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 Message BoxA. For more information, look at the Platform Invoke tutorial in the documentation.

1. **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.

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.

1. **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.

1. **What's A Strong Name?**

A strong name includes the name of the assembly, version number, culture identity, and a public key token.

1. **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.

1. **How Can You Debug Failed Assembly Binds?**

Use the Assembly Binding Log Viewer (fuslogvw.exe) to find out the paths searched.

1. **Where Are Shared Assemblies Stored?**

Global assembly cache.

1. **How Can You Create A Strong Name For A .net Assembly?**

With the help of Strong Name tool (sn.exe).

1. **Where's Global Assembly Cache Located On The System?**

Usually C:\winnt\assembly or C:\windows\assembly.

1. **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.

1. **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.

1. **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.

1. **Is There An Equivalent Of Exit() For Quitting A C# .net Application?**

Yes, you can use System.Environment.Exit(int exitCode) to exit the application or Application.Exit() if it's a Windows Forms app.

1. **Can You Prevent Your Class From Being Inherited And Becoming A Base Class For Some Other Classes?**

Yes, that is 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 is the same concept as final class in Java.

1. **If A Base Class Has A Bunch Of Overloaded Constructors, And An Inherited Class Has Another Bunch Of Overloaded Constructors, Can You Enforce A Call From An Inherited Constructor To An Arbitrary Base Constructor?**

Yes, just place a colon, and then keyword base (parameter list to invoke the appropriate constructor) in the overloaded constructor definition inside the inherited class.

1. **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) { }

1. **How Do I Make A Dll In C#?**

You need to use the /target:library compiler option.

1. **What Is The C# Equivalent Of C++ Catch (....), Which Was A Catch-all Statement For Any Possible Exception? Does C# Support Try-catch-finally Blocks?**

Yes. Try-catch-finally blocks are supported by the C# compiler. Here's an example of a try-catch-finally block:  
using System;  
public class TryTest  
{  
static void Main()  
{  
try  
{  
Console.WriteLine("In Try block");  
throw new ArgumentException();  
}  
catch(ArgumentException n1)  
{  
Console.WriteLine("Catch Block");  
}  
finally  
{  
Console.WriteLine("Finally Block");  
}  
}  
}  
Output: In Try Block  
Catch Block  
Finally Block

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, as shown in the following example:  
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).

1. **Is There Regular Expression (regex) Support Available To C# Developers?**

Yes. The .NET class libraries provide support for regular expressions. Look at the documentation for the System. Text.Regular Expressions namespace.

1. **Is There A Way To Force Garbage Collection?**

Yes. Set all references to null and then call System.GC.Collect(). If you need to have some objects destructed, and System.GC.Collect() doesn't seem to be doing it for you, you can force finalizers to be run by setting all the references to the object to null and then calling System.GC.RunFinalizers().

1. **Does C# Support Properties Of Array Types?**

Yes. Here's a simple example:  
using System;  
class Class1  
{  
private string[] MyField;  
public string[] MyProperty  
{  
get { return MyField; }  
set { MyField = value; }  
}  
}  
class MainClass  
{  
public static int Main(string[] args)  
{  
Class1 c = new Class1();  
string[] arr = new string[] {"apple", "banana"};  
c.MyProperty = arr;  
Console.WriteLine(c.MyProperty[0]); // "apple"  
return 0;  
}  
}

1. **How Is 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.

1. **When Do You Absolutely Have To Declare A Class As Abstract (as Opposed To Free-willed Educated Choice Or Decision Based On Uml Diagram)?**

When at least one of the methods in the class is abstract. When the class itself is inherited from an abstract class, but not all base abstract methods have been over-ridden.

1. **Why Would You Use Untrusted Verification?**

Web Services might use it, as well as non-Windows applications.

1. **What Is The Implicit Name Of The Parameter That Gets Passed Into The Class Set Method?**

Value, and its datatype depends on whatever variable we are changing.

1. **How Do I Register My Code For Use By Classic Com Clients?**

Use the regasm.exe utility to generate a type library (if needed) and the necessary entries in the Windows Registry to make a class available to classic COM clients. Once a class is registered in the Windows Registry with regasm.exe, a COM client can use the class as though it were a COM class.

1. **How Do I Do Implement A Trace And Assert?**

Use a conditional attribute on the method, as shown below:  
class Debug  
{  
[conditional("TRACE")]  
public void Trace(string s)  
{  
Console.WriteLine(s);  
}  
}  
class MyClass  
{  
public static void Main()  
{  
Debug.Trace("hello");  
}  
}

In this example, the call to Debug.Trace() is made only if the preprocessor symbol TRACE is defined at the call site. You can define preprocessor symbols on the command line by using the /D switch. The restriction on conditional methods is that they must have void return type.

1. **How Do I Create A Multi Language, Multi File Assembly?**

Unfortunately, this is currently not supported in the IDE. To do this from the command line, you must compile your projects into netmodules (/target:module on the C# compiler), and then use the command line tool al.exe (alink) to link these netmodules together.

1. **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 is no implementation in.

1. **What Is The Equivalent To Regsvr32 And Regsvr32 /u A File In .net Development?**

Try using RegAsm.exe. The general syntax would be: RegAsm. A good description of RegAsm and its associated switches is located in the .NET SDK docs. Just search on "Assembly Registration Tool".Explain ACID rule of thumb for transactions.

Transaction must be Atomic (it is one unit of work and does not dependent on previous and following transactions), Consistent (data is either committed or roll back, no in-between case where something has been updated and something hasnot), Isolated (no transaction sees the intermediate results of the current transaction), Durable (the values persist if the data had been committed even if the system crashes right after).

1. **How Do I Create A Multilanguage, Single-file Assembly?**

This is currently not supported by Visual Studio .NET.

1. **Why Cannot You Specify The Accessibility Modifier For Methods Inside The Interface?**

They all must be public. Therefore, to prevent you from getting the false impression that you have any freedom of choice, you are not allowed to specify any accessibility, it is public by default.

1. **Is It Possible To Restrict The Scope Of A Field/method Of A Class To The Classes In The Same Namespace?**

There is no way to restrict to a namespace. Namespaces are never units of protection. But if you're using assemblies, you can use the 'internal' access modifier to restrict access to only within the assembly.

1. **Why Do I Get A Syntax Error When Trying To Declare A Variable Called Checked?**

The word checked is a keyword in C#.

1. **What Is The Syntax For Calling An Overloaded Constructor Within A Constructor (this() And Constructorname() Does Not Compile)?**

The syntax for calling another constructor is as follows:  
class B  
{  
B(int i)  
{ }  
}  
class C : B  
{  
C() : base(5) // call base constructor B(5)  
{ }  
C(int i) : this() // call C()  
{ }  
public static void Main() {}  
}

1. **Why Do I Get A "cs5001: Does Not Have An Entry Point Defined" Error When Compiling?**

The most common problem is that you used a lowercase 'm' when defining the Main method. The correct way to implement the entry point is as follows:  
class test  
{  
static void Main(string[] args) {}  
}

1. **What Does The Keyword Virtual Mean In The Method Definition?**

The method can be over-ridden.

1. **What Optimizations Does The C# Compiler Perform When You Use The /optimize+ Compiler Option?**

The following is a response from a developer on the C# compiler team:  
We get rid of unused locals (i.e., locals that are never read, even if assigned).  
We get rid of unreachable code.  
We get rid of try-catch w/ an empty try.  
We get rid of try-finally w/ an empty try (convert to normal code...).  
We get rid of try-finally w/ an empty finally (convert to normal code...).  
We optimize branches over branches:  
gotoif A, lab1  
goto lab2:  
lab1:  
turns into: gotoif !A, lab2  
lab1:  
We optimize branches to ret, branches to next instruction, and branches to branches.

1. **How Can I Create A Process That Is Running A Supplied Native Executable (e.g., Cmd.exe)?**

The following code should run the executable and wait for it to exit before continuing:  
using System;  
using System.Diagnostics;  
public class ProcessTest {  
public static void Main(string[] args) {  
Process p = Process.Start(args[0]);  
p.WaitForExit();  
Console.WriteLine(args[0] + " exited.");  
}  
}  
Remember to add a reference to System.Diagnostics.dll when you compile.

1. **What Is The Difference Between The System.array.copyto() And System.array.clone()?**

The first one performs a deep copy of the array, the second one is shallow.

1. **How Do I Declare Inout Arguments In C#?**

The equivalent of inout in C# is ref. , as shown in the following example:  
public void MyMethod (ref String str1, out String str2)  
{  
...  
}  
When calling the method, it would be called like this:  
String s1;  
String s2;  
s1 = "Hello";  
MyMethod(ref s1, out s2);  
Console.WriteLine(s1);  
Console.WriteLine(s2);  
Notice that you need to specify ref when declaring the function and calling it.

1. **Is There A Way Of Specifying Which Block Or Loop To Break Out Of When Working With Nested Loops?**

The easiest way is to use goto:  
using System;  
class BreakExample  
{  
public static void Main(String[] args)  
{  
for(int i=0; i<3; i++)  
{  
Console.WriteLine("Pass {0}: ", i);  
for( int j=0 ; j<100 ; j++ )  
{  
if ( j == 10) goto done;  
Console.WriteLine("{0} ", j);  
}  
Console.WriteLine("This will not print");  
}  
done:  
Console.WriteLine("Loops complete.");  
}  
}

1. **What Is The Difference Between Const And Static Read-only?**

The difference is that static read-only can be modified by the containing class, but const can never be modified and must be initialized to a compile time constant. To expand on the static read-only case a bit, the containing class can only modify it: -- in the variable declaration (through a variable initializer).  
-- in the static constructor (instance constructors if it's not static).

1. **What Is 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.

1. **What Is The Top .net Class That Everything Is Derived From?**

System.Object.

1. **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.

1. **From A Versioning Perspective, What Are The Drawbacks Of Extending An Interface As Opposed To Extending A Class?**

With regard to versioning, interfaces are less flexible than classes. With a class, you can ship version 1 and then, in version 2, decide to add another method. As long as the method is not abstract (i.e., as long as you provide a default implementation of the method), any existing derived classes continue to function with no changes. Because interfaces do not support implementation inheritance, this same pattern does not hold for interfaces. Adding a method to an interface is like adding an abstract method to a base class--any class that implements the interface will break, because the class doesn't implement the new interface method.

1. **Does Console.writeline() Stop Printing When It Reaches A Null Character Within A String?**

Strings are not null terminated in the runtime, so embedded nulls are allowed. Console.WriteLine() and all similar methods continue until the end of the string.

1. **What Is The Advantage Of Using System.text.stringbuilder Over System.string?**

StringBuilder is more efficient in the cases, where a lot of manipulation is done to the text. Strings are im mutable , so each time it is being operated on, a new instance is created.

1. **Why Do I Get A Security Exception When I Try To Run My C# App?**

Some security exceptions are thrown if you are working on a network share. There are some parts of the frameworks that will not run if being run off a share (roaming profile, mapped drives, etc.). To see if this is what's happening, just move the executable over to your local drive and see if it runs without the exceptions. One of the common exceptions thrown under these conditions is System.Security.SecurityException.

To get around this, you can change your security policy for the intranet zone, code group 1.2, (the zone that running off shared folders falls into) by using the caspol.exe tool.

1. **Is There Any Sample C# Code For Simple Threading?**

Some sample code follows: using System;  
using System.Threading;  
class ThreadTest  
{  
public void runme()  
{  
Console.WriteLine("Runme Called");  
}  
public static void Main(String[] args)  
{  
ThreadTest b = new ThreadTest();  
Thread t = new Thread(new ThreadStart(b.runme));  
t.Start();  
}  
}

1. **What Is The Difference Between // Comments, /\* \*/ Comments And /// Comments?**

Single-line, multi-line and XML documentation comments.

1. **How Do You Inherit From A Class In C#?**

Place a colon and then the name of the base class. Notice that it is double colon in C++.

1. **How Do I Port "synchronized" Functions From Visual J++ To C#?**

Original Visual J++ code: public synchronized void Run()  
{  
// function body  
}  
Ported C# code: class C  
{  
public void Run()  
{  
lock(this)  
{  
// function body  
}  
}  
public static void Main() {}  
}

1. **Can I Define A Type That Is An Alias Of Another Type (like Typedef In C++)?**

Not exactly. You can create an alias within a single file with the "using" directive: using System; using Integer = System.Int32; // alias  
But you can't create a true alias, one that extends beyond the file in which it is declared. Refer to the C# spec for more info on the 'using' statement's scope.

1. **Is It Possible To Have A Static Indexer In C#?**

No. Static indexers are not allowed in C#.

1. **Does C# Support #define For Defining Global Constants?**

No. If you want to get something that works like the following C code:  
#define A 1  
use the following C# code: class MyConstants  
{  
public const int A = 1;  
}  
Then you use MyConstants.A where you would otherwise use the A macro.  
Using MyConstants.A has the same generated code as using the literal 1.

1. **Does C# Support Templates?**

No. However, there are plans for C# to support a type of template known as a generic. These generic types have similar syntax but are instantiated at run time as opposed to compile time. You can read more about them here.

1. **Does C# Support Parameterized Properties?**

No. C# does, however, support the concept of an indexer from language spec. An indexer is a member that enables an object to be indexed in the same way as an array. Whereas properties enable field-like access, indexers enable array-like access. As an example, consider the Stack class presented earlier. The designer of this class may want to expose array-like access so that it is possible to inspect or alter the items on the stack without performing unnecessary Push and Pop operations. That is, Stack is implemented as a linked list, but it also provides the convenience of array access.

Indexer declarations are similar to property declarations, with the main differences being that indexers are nameless (the name used in the declaration is this, since this is being indexed) and that indexers include indexing parameters. The indexing parameters are provided between square brackets.

1. **Does C# Support C Type Macros?**

No. C# does not have macros. Keep in mind that what some of the predefined C macros (for example, \_\_LINE\_\_ and \_\_FILE\_\_) give you can also be found in .NET classes like System.Diagnostics (for example, StackTrace and StackFrame), but they'll only work on debug builds.

1. **Can You Declare The Override Method Static While The Original Method Is Non-static?**

No, you cannot, the signature of the virtual method must remain the same, only the keyword virtual is changed to keyword override.

1. **Does C# Support Multiple Inheritance?**

No, use interfaces instead.

1. **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.

1. **What Is The Data Provider Name To Connect To Access Database?**

Microsoft.Access.

1. **Why Does My Windows Application Pop Up A Console Window Every Time I Run It?**

Make sure that the target type set in the project properties setting is set to Windows Application, and not Console Application. If you're using the command line, compile with /target:winexe & not target:exe.

1. **Describe The Accessibility Modifier Protected Internal?**

It is available to derived classes and classes within the same Assembly (and naturally from the base class it is declared in).

1. **How Do I Get Deterministic Finalization In C#?**

In a garbage collected environment, it's impossible to get true determinism. However, a design pattern that we recommend is implementing IDisposable on any class that contains a critical resource. Whenever this class is consumed, it may be placed in a using statement, as shown in the following example:  
using(FileStream myFile = File.Open(@"c:temptest.txt", FileMode.Open))  
{  
int fileOffset = 0;  
while(fileOffset < myFile.Length)  
{  
Console.Write((char)myFile.ReadByte());  
fileOffset++;  
}  
}  
When myFile leaves the lexical scope of the using, its dispose method will be called.

1. **How Can I Get Around Scope Problems In A Try/catch?**

If you try to instantiate the class inside the try, it'll be out of scope when you try to access it from the catch block. A way to get around this is to do the following:  
Connection conn = null;  
try  
{  
conn = new Connection();  
conn.Open();  
}  
finally  
{  
if (conn != null) conn.Close();  
}  
By setting it to null before the try block, you avoid getting the CS0165 error (Use of possibly unassigned local variable 'conn').

1. **Why Do I Get An Error (cs1006) When Trying To Declare A Method Without Specifying A Return Type?**

If you leave off the return type on a method declaration, the compiler thinks you are trying to declare a constructor. So if you are trying to declare a method that returns nothing, use void. The following is an example: // This results in a CS1006 error public static staticMethod (mainStatic obj) // This will work as wanted public static void staticMethod (mainStatic obj)

1. **How Do I Convert A String To An Int In C#?**

Here's an example:  
using System;  
class StringToInt  
{  
public static void Main()  
{  
String s = "105";  
int x = Convert.ToInt32(s);  
Console.WriteLine(x);  
}  
}

1. **What Is The Difference Between A Struct And A Class In C#?**

From language spec:  
The list of similarities between classes and structs is as follows. Longstructs can implement interfaces and can have the same kinds of members as classes. Structs differ from classes in several important ways; however, structs are value types rather than reference types, and inheritance is not supported for structs. Struct values are stored on the stack or in-line. Careful programmers can sometimes enhance performance through judicious use of structs. For example, the use of a struct rather than a class for a Point can make a large difference in the number of memory allocations performed at runtime. The program below creates and initializes an array of 100 points. With Point implemented as a class, 101 separate objects are instantiated-one for the array and one each for the 100 elements.

1. **What Is The Difference Between The Debug Class And Trace Class?**

Documentation looks the same. Use Debug class for debug builds, use Trace class for both debug and release builds.

1. **How Can You Overload A Method?**

Different parameter data types, different number of parameters, different order of parameters.

1. **When You Inherit A Protected Class-level Variable, Who Is It Available To?**

Classes in the same namespace.

1. **How Can I Get The Ascii Code For A Character In C#?**

Casting the char to an int will give you the ASCII value: char c = 'f';  
System.Console.WriteLine((int)c); or for a character in a string:  
System.Console.WriteLine((int)s[3]);  
The base class libraries also offer ways to do this with the Convert class or Encoding classes if you need a particular encoding.

1. **Is There An Equivalent To The Instanceof Operator In Visual J++?**

C# has the is operator:  
expr is type

1. **How Do I Create A Delegate/multicastdelegate?**

C# requires only a single parameter for delegates: the method address. Unlike other languages, where the programmer must specify an object reference and the method to invoke, C# can infer both pieces of information by just specifying the method's name. For example, let's use System.Threading.ThreadStart: Foo MyFoo = new Foo(); ThreadStart del = new ThreadStart(MyFoo.Baz); This means that delegates can invoke static class methods and instance methods with the exact same syntax!

1. **How Do Destructors And Garbage Collection Work In C#?**

C# has finalizers (similar to destructors except that the runtime doesn't guarantee they'll be called), and they are specified as follows:  
class C  
{  
~C()  
{  
// your code  
}  
public static void Main() {}  
}  
Currently, they override object.Finalize(), which is called during the GC process.

1. **My Switch Statement Works Differently! Why?**

C# does not support an explicit fall through for case blocks. The following code is not legal and will not compile in C#:  
switch(x)  
{  
case 0:  
// do something  
case 1:  
// do something in common with 0  
default:  
// do something in common with  
//0, 1 and everything else  
break;  
}  
To achieve the same effect in C#, the code must be modified as shown below (notice how the control flows are explicit):  
class Test  
{  
public static void Main()  
{  
int x = 3;  
switch(x)  
{  
case 0:  
// do something  
goto case 1;  
case 1:  
// do something in common with 0  
goto default;  
default:  
// do something in common with 0, 1, and anything else  
break;  
}  
}  
}

1. **How Can I Access The Registry From C# Code?**

By using the Registry and RegistryKey classes in Microsoft.Win32, you can easily access the registry. The following is a sample that reads a key and displays its value:  
using System;using Microsoft.Win32;  
class regTest  
{  
public static void Main(String[] args)  
{  
RegistryKey regKey;  
Object value;  
regKey = Registry.LocalMachine;  
regKey = regKey.OpenSubKey("HARDWAREDESCRIPTIONSystemCentralProcessor ");  
value = regKey.GetValue("VendorIdentifier");  
Console.WriteLine("The central processor of this machine is: {0}.", value);  
}  
}

1. **Why Does Dllimport Not Work For Me?**

All methods marked with the DllImport attribute must be marked as public static extern.

1. **What Is The Difference Between An Interface And Abstract Class?**

In the interface all methods must be abstract; in the abstract class some methods can be concrete. In the interface no accessibility modifiers are allowed, which is ok in abstract classes.

1. **Ctype(123.34,integer) - Should It Throw An Error? Why Or Why Not?**

It would work fine. As the runtime type of 123.34 would be double, and Double can be converted to Integer.

the ctype(123.34,integer) will work fine no errors.

1. **Directcast(123.34,integer) - Should It Throw An Error? Why Or Why Not?**

It would throw an InvalidCast exception as the runtime type of 123.34 (double) doesnt match with Integer.

1. **Difference Between A Sub And A Function.**

A Sub does not return anything whereas a Function returns something.

-A Sub Procedure is a method will not return a value  
-A sub procedure will be defined with a “Sub” keyword

Sub ShowName(ByVal myName As String)  
Console.WriteLine(”My name is: ” & myName)  
End Sub

-A function is a method that will return value(s).  
-A function will be defined with a “Function” keyword

Function FindSum(ByVal num1 As Integer, ByVal num2 As Integer) As Integer  
Dim sum As Integer = num1 + num2  
Return sum  
End Function

1. **Explain Manifest & Metadata.**

Manifest is metadata about assemblies. Metadata is machine-readable information about a resource, or “”data about data.” In .NET, metadata includes type definitions, version information, external assembly references, and other standardized information.

Manifest: Manifest describes assembly itself. Assembly Name, version number, culture, strong name, list of all files, Type references, and referenced assemblies.

Metadata: Metadata describes contents in an assembly classes, interfaces, enums, structs, etc., and their containing namespaces, the name of each type, its visibility/scope, its base class, the nterfaces it implemented, its methods and their scope, and each method’s parameters, type’s properties, and so on.

1. **Difference Between Imperative And Interrogative Code.**

There are imperative and interrogative functions. Imperative functions are the one which return a value while the interrogative functions do not return a value.

1. **Difference Between Value And Reference Type. What Are Value Types And Reference Types?**

Value type - bool, byte, chat, decimal, double, enum , float, int, long, sbyte, short, strut, uint, ulong, ushort  
Value types are stored in the Stack  
Reference type - class, delegate, interface, object, string  
Reference types are stored in the Heap

1. **What Are The Two Kinds Of Properties**

Two types of properties in .Net: Get and Set

1. **Explain Constructor.**

Constructor is a method in the class which has the same name as the class (in VB.Net its New()). It initializes the member attributes whenever an instance of the class is created.

1. **Describe Ways Of Cleaning Up Objects.**

There is a perfect tool provide by .net frameworks calls Garbage collector, where by mean of GC we can clean up the object and reclaim the memory. The namespace used is System.GC

the run time will maintain a service called as garbage collector. This service will take care of deallocating memory corresponding to objects. it works as a thread with least priority. when application demands for memory the runtime will take care of setting the high priority for the garbage collector, so that it will be called for execution and memory will be released. the programmer can make a call to garbage collector by using GC class in system name space.

1. **How Can You Clean Up Objects Holding Resources From Within The Code?**

Call the dispose method from code for clean up of objects.

1. **Which Controls Do Not Have Events?**

Timer control.

1. **What Is The Maximum Size Of The Textbox?**

65536.

1. **Which Property Of The Textbox Cannot Be Changed At Runtime?**

Locked Property.

1. **Which Control Cannot Be Placed In Mdi?**

The controls that do not have events.

1. **What Is The Difference Between Proc. Sent By Val And By Sub?**

BY VAL: changes will not be reflected back to the variable.  
By REF: changes will be reflected back to that variable.( same as & symbol in c, c++)

1. **What's The Advantage Of Using System.text.stringbuilder Over System.string?**

StringBuilder is more efficient in the cases, where a lot of manipulation is done to the text. Strings are immutable, so each time it’s being operated on, a new instance is created.

1. **What's The Difference Between The System.array.copyto() And System.array.clone()?**

The first one performs a deep copy of the array, the second one is shallow

1. **What's Class Sortedlist Underneath?**

A sorted HashTable.

1. **What's The C# Equivalent Of C++ Catch (...), Which Was A Catch-all Statement For Any Possible Exception?**

A catch block that catches the exception of type System.Exception. You can also omit the parameter data type in this case and just write catch {}.

1. **What's A Delegate?**

A delegate object encapsulates a reference to a method. In C++ they were referred to as function pointers.

1. **What's A Multicast Delegate?**

It’s a delegate that points to and eventually fires off several methods.

1. **How's The Dll Hell Problem Solved In .net?**

Assembly versioning allows the application to specify not only the library it needs to run (which was available under Win32), but also the version of the assembly.

1. **What's A Satellite Assembly?**

When you write a multilingual or multi-cultural application in .NET, and want to distribute the core application separately from the localized modules, the localized assemblies that modify the core application are called satellite assemblies.

1. **What's The Difference Between // Comments, /\* \*/ Comments And /// Comments?**

Single-line, multi-line and XML documentation comments.

1. **What's The Difference Between <c> And <code> Xml Documentation Tag?**

Single line code example and multiple-line code example.

1. **What's The Implicit Name Of The Parameter That Gets Passed Into The Set Method/property Of A Class?**

Value. The data type of the value parameter is defined by whatever data type the property is declared as.