Interview Preparation for Senior .NET Developer Full Stack

Gang of Four Design Patterns

1. What is UML

   Unified Modeling Language is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system.
2. Singleton- A class of which only a single instance can exist The **Singleton pattern** is used to design the classes which provides the configuration settings for an application. By implementing configuration classes as **Singleton** not only that we provide a global access point, but we also keep the instance we **use** as a cache object.

1.What is Design Pattern?

A design pattern is a general reusable solution to a commonly occurring problem within a given context in software design. A design pattern is not a finished design that can be transformed directly into source or machine code. It is a description or template for how to solve a problem that can be used in many different situations. Patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system

2.What all are the advantages of Design Patterns?

Patterns solve software structural and non functional problems.

Structural problems like:

- *Abstraction*

- *Encapsulation*
- *Data Hiding*
- *Separation of Concerns*
- *Separation of Interface and Implementation*
- *Single point of reference*
- *Coupling and Cohesion etc..*

Non Functional problems like:

- *Efficiency*
- *Reliability*
- *Interoperability*
- *Testability*
- *Reusability etc..*

3.What all are the types of Design Patterns?

There are 3 types of Design Pattern.

- Creational Patterns
    - This type of pattern address problems of creating an object and separating it from operations
- Structural Patterns
    - This type of pattern address problems of using object oriented constructs to organize classes and objects
- Behavioral Patterns
    - This type of pattern address problems of assigning responsibilities to classes

4.What is Creational Design Pattern?

In software engineering, creational design patterns are design patterns that deal with object creation mechanisms, trying to create objects in a manner suitable to the situation.

In C#, we have 5 types of Design Patterns in Creational Catagory.

- Singleton
- Factory
- Abstract Factory
- Prototype
- Builder

5.What is Structural Design Pattern?

Structural patterns are concerned with how classes and objects are composed to form larger structures; the class form of the Adapterdesign pattern is an example.

Structural class patterns use inheritance to compose interface or implementations.

Structural Design Patterns are Design Patterns that ease the design by identifying a simple way to realize relationships between entities.

In C#, We have 7 types of design patterns in Structural Catagory.

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

6.What is Singleton Design Pattern?

- Singleton ensures a class only has one instance.
- Singleton Provides a global point of access to it.

You can see more details on Singleton Design patterns **here**.

7.What is Factory Design Pattern?

Factory Design patterns:

- Define an Interface for creating an object but let subclasses decide which class to instantiate
- Lets a class defer instantiation to subclasses

You can see more details on Factory Design patterns **here**.

8.What is Abstract Factory Design Pattern?

Abstract Factory Design patterns:

- Provides an interface for creating families of related or dependent objects without specifying their concrete classes.
- Abstract Factory patterns acts a super-factory which creates other factories. This pattern is also called as Factory of factories

You can see more details on Abstract Factory Design patterns **here**.

9.What is Prototype Design Pattern?

Prototype Design patterns:

- Prototype pattern specifies the kind of objects to create using a prototypical instance, and create new objects by copying this prototype.
- It is used to create a duplicate object or clone of the current object to enhance performance.

You can see more details on Prototype Design patterns **here**.

10.What is Builder Design Pattern?

Builder Design patterns:

- Separate the construction of a complex object from its representation so that the same construction process can create different representations.
- In other words,you will have to design the system in such a way that the client application will simply specify the parameters that should be used to create the complex object and the builder will take care of building the complex object.

You can see more details on Builder Factory Design patterns **here**.

11.What is Adapter Design Pattern?

Adapter Design patterns:

- The adapter pattern is adapting between classes and objects
- This pattern involves a single class called adapter which is responsible for communication between two independent or incompatible interfaces
- This works like a bridge between two incompatible interfaces

You can see more details on Abstract Design patterns **here**.

12.What is Bridge Design Pattern?

Bridge Design patterns:

- Bridge Pattern separates abstraction from its implementation, so that both can be modified Independently
- Bridge Pattern behaves like a bridge between abstraction class and Implementer class.

You can see more details on Bridge Design patterns **here**.

13.What is Composite Design Pattern?

Composite Design patterns:

- Composite pattern composes objects in term of a tree structure to represent part as well as whole hierarchies.
- Composite pattern creates a class contains group of its own objects. This class provides ways to modify its group of same objects.
- Composite pattern is used when we need to treat a group of objects and a single object in the same way

You can see more details on Composite Design patterns **here**.

14.What is Decorator Design Pattern?

Decorator Design patterns:

- Decorator pattern is used to add new functionality to an existing object without changing its structure.
- Decorators provide a flexible alternative to subclass for extending functionality.
- This pattern creates a decorator class which wraps the original class and add new behaviors/operations to an object at run-time.

You can see more details on Decorator Design patterns **here**.

15.What is Facade Design Pattern?

Facade Design patterns:

- Facade Design Pattern makes a software library easier to use, understand and test
- Facade Design Pattern make the library more readable
- Facade Design Pattern reduce dependencies of outside code on the inner workings of a library
- Facade Design Pattern wrap a poorly designed collection of APIs with a single well-designed API.

You can see more details on Facade Design patterns **here**.

16.What is Flyweight Design Pattern?

Flyweight Design patterns:

- Flyweight design pattern is an object that minimizes memory use by sharing as much data as possible with other similar objects
- Flyweight pattern is used to reduce the number of objects created, to decrease memory and resource usage. As a result it increase performance
- Flyweight design pattern provides a way to use objects in large numbers when a simple repeated representation would use an unacceptable amount of memory.
- The flyweight pattern uses the concepts of intrinsic and extrinsic data.Intrinsic datais held in the properties of the shared flyweight objects. This information is stateless and generally remains unchanged, if any change occurs it would be reflected among all of the objects that reference the flyweight.Extrinsic data is computed on the fly means at runtime and it is held outside of a flyweight object. Hence it can be stateful.

You can see more details on Flyweight Design patterns **here**.

17.What is Proxy Design Pattern?

Proxy Design patterns:

- Proxy Design pattern involves a class, called proxy class, which represents functionality of another class.
- Proxy is a wrapper or agent object that is being called by the client to access the real serving object behind the scenes.

You can see more details on Proxy Design patterns **here**.

Four Pillars of Object Oriented Development

- Abstraction - **Abstraction** is one of the principle of object oriented programming. It is used to display only necessary and essential features of an object to outside the world.Means displaying what is necessary and encapsulate the unnecessary things to outside the world.Hiding can be achieved by using "private" access modifiers.
- Polymorphism - is the ability of objects of different types to provide a unique interface for different implementations of methods. ... **Polymorphism** forms one of the fundamental concepts of object-oriented programming, along with encapsulation and inheritance.
- Inheritance - An **interface** in **C#** contains only the declaration of the methods, properties, and events, but not the implementation. It is left to the class that implements the **interface** by providing implementation for all the members of the **interface**. **Interface** makes it easy to maintain a program.
- Encapsulation - in the context of **C#**, refers to an object's ability to hide data and behavior that are not necessary to its user. **Encapsulation** enables a group of properties, methods and other members to be considered a single unit or object.

Important features of C# –

- Boolean Conditions
- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management
- Easy-to-use Generics – .NET 2.0 Generics were added to version 2.0 of the C# language and the common language runtime (CLR). Generics introduce to the .NET Framework the concept of type parameters, which make it possible to

design classes and methods that defer the specification of one or more types until the class or method is declared and instantiated by client code. For example, by using a generic type parameter T you can write a single class that other client code can use without incurring the cost or risk of runtime casts or boxing operations

- Indexers
- Conditional Compilation
- Simple Multithreading
- LINQ **Language Integrated Query** and Lambda Expressions
- Integration with Windows

GAC - The Global Assembly Cache (**GAC**) is a folder in Windows directory to store the .NET assemblies that are specifically designated to be shared by all applications executed on a system.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. As a general guideline, keep assembly dependencies private, and locate assemblies in the application directory unless sharing an assembly is explicitly required. In addition, it is not necessary to install assemblies into the global assembly cache to make them accessible to COM interop or unmanaged code.

What makes a strong-named assembly?

A strong named assembly is generated by using the private key that corresponds to the public key distributed with the assembly, and the assembly itself. The assembly includes the assembly manifest, which contains the names and hashes of all the files that make up the assembly. Assemblies that have the same strong name should be identical.+

You can strong-name assemblies by using Visual Studio or a command-line tool. For more information, see How to: Sign an Assembly with a Strong Name or Sn.exe (Strong Name Tool).

When a strong-named assembly is created, it contains the simple text name of the assembly, the version number, optional culture information, a digital signature, and the public key that corresponds to the private key used for signing.

When a value type is converted to object type, it is called **boxing** and on the other hand, when an object type is converted to a value type, it is called **unboxing**.

6) Explain sealed class in C#?

Sealed class is used to prevent the class from being inherited from other classes. So “sealed” modifier also can be used with methods to avoid the methods to override in the child classes.

13) Explain “static” keyword in C#?

“Static” keyword can be used for declaring a static member. If the class is made static then all the members of the class are also made static. If the variable is made static then it will have a single instance and the value change is updated in this instance.

14) What is the difference between “dispose” and “finalize” variables in C#?

- Dispose - This method uses the interface – “IDisposable” interface and it will free up both managed and unmanaged codes like – database connection, files etc.
- Finalize - This method is called internally unlike the Dispose method which is called explicitly. It is called by garbage collector and can’t be called from the code.

18) What is the difference between “finalize” and “finally” methods in C#?

- Finalize – This method is used for garbage collection. So before destroying an object this method is called as part of clean up activity.
- Finally – This method is used for executing the code irrespective of exception occurred or not.

19) What is the difference between “throw ex” and “throw” methods in C#?

- **throw**(**ex**) will reset your stack trace so error will appear from the line where **throw**(**ex**) written while **throw** does not reset stack trace and you will get information about original **exception**. In MSIL code when you use **throw**(**ex**) it will generate code as **throw** and if you use **throw** it will create rethrow

23) Mention the assembly name where System namespace lies in C#?

Assembly Name – mscorlib.dll

Common **Intermediate Language** (CIL, pronounced either "sil" or "kil"), formerly called Microsoft **Intermediate Language** or MSIL, is the lowest-level human-readable programming **language** defined by the Common **Language** Infrastructure (CLI) specification and is used by the .**NET Framework** and Mono.

31) Explain access modifier – “protected internal” in C#?

“protected internal” can be accessed in the same assembly and the child classes can also access these methods.

36) What is the difference between methods – “System.Array.Clone()” and “System.Array.CopyTo()” in C#?

- “CopyTo()” method can be used to copy the elements of one array to other.
- “Clone()” method is used to create a new array to contain all the elements which are in the original array.

The Basic OOP Concepts

If you are new to object-oriented programming languages, you will need to know a few basics before you can get started with code. The following Webopedia definitions will help you better understand object-oriented programming:

- **Abstraction: The process of picking out (abstracting) common features of objects and procedures.**
- **Class: A category of objects. The class defines all the common properties of the different objects that belong to it.**
- **Encapsulation: The process of combining elements to create a new entity. A procedure is a type of encapsulation because it combines a series of computer instructions.**
- **Information hiding: The process of hiding details of an object or function. Information hiding is a powerful programming technique because it reduces complexity.**

- **Inheritance: a feature that represents the "is a" relationship between different classes.**
- **Interface: the languages and codes that the applications use to communicate with each other and with the hardware.**
- **Messaging: Message passing is a form of communication used in parallel programming and object-oriented programming.**
- **Object: a self-contained entity that consists of both data and procedures to manipulate the data.**
- **Polymorphism: A programming language's ability to process objects differently depending on their data type or class.**
- **Procedure: a section of a program that performs a specific task.**

C# Object Oriented

1. What is C#?

C# is an object oriented, type safe and managed language that is compiled by .Net framework to generate Microsoft Intermediate Language.

Interfaces are a contract consisting of Public sets of Members:

- Properties
- Methods
- Events
- Indexers

Visibility is a must,otherwise why have a contract or Interface.

Concrete Class = Standard Class that we use everyday….

Abstract Class = Class member, but not implemented using the public abstract class

public abstract double GetArea();

This will force the compiler that we must implement an abstract method in a child class

public interface IRegularPolygon ---- C# interface starts with capital letter I

Declarations are listed in the interface….

Abstract and Interfaces require you to State Methods

Compile time error is a lot better than a runtime error Abstract Class may contain implementation.

Abstract Classes can contain the following:

Fields

Properties

Constructors

Destructors

Methods

Events

Indexers

Interfaces

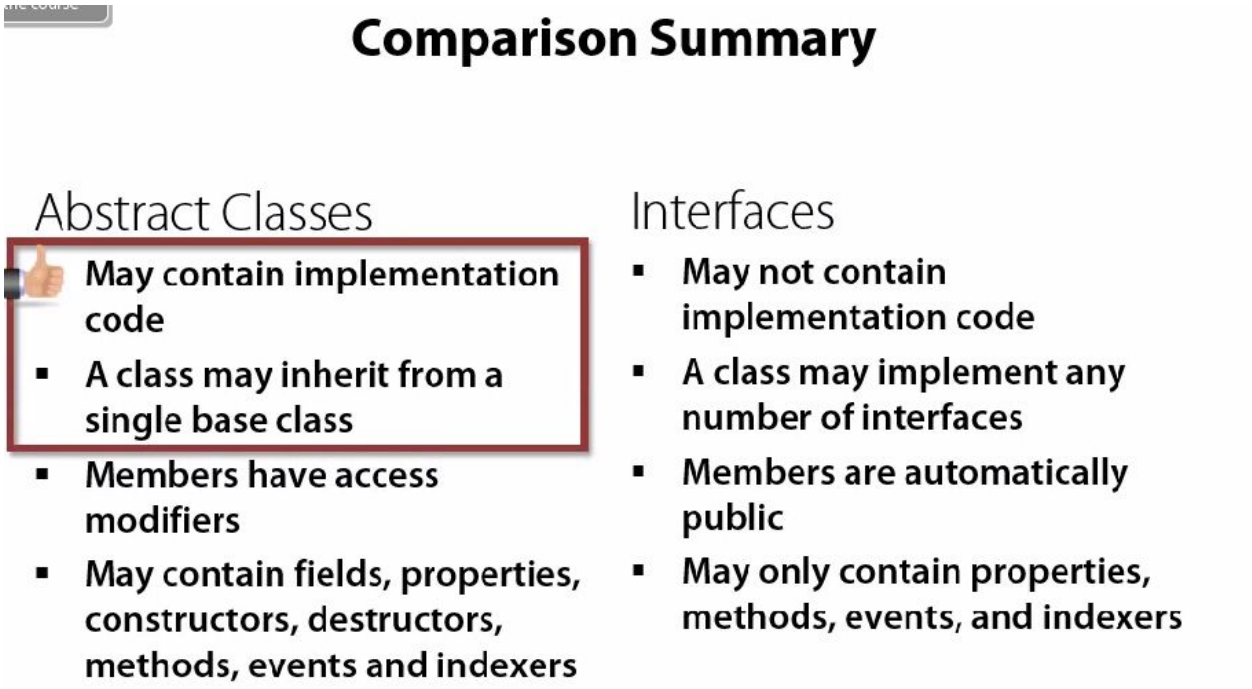
Properties

Methods

Events

Indexers

Abstract Classes vs Interfaces



Abstract vs Concrete

Abstract classes cannot be instantiated directly. Declaring a class as abstract means that you do not want it to be instantiated and that the class can only be inherited. You are imposing a rule in your code.

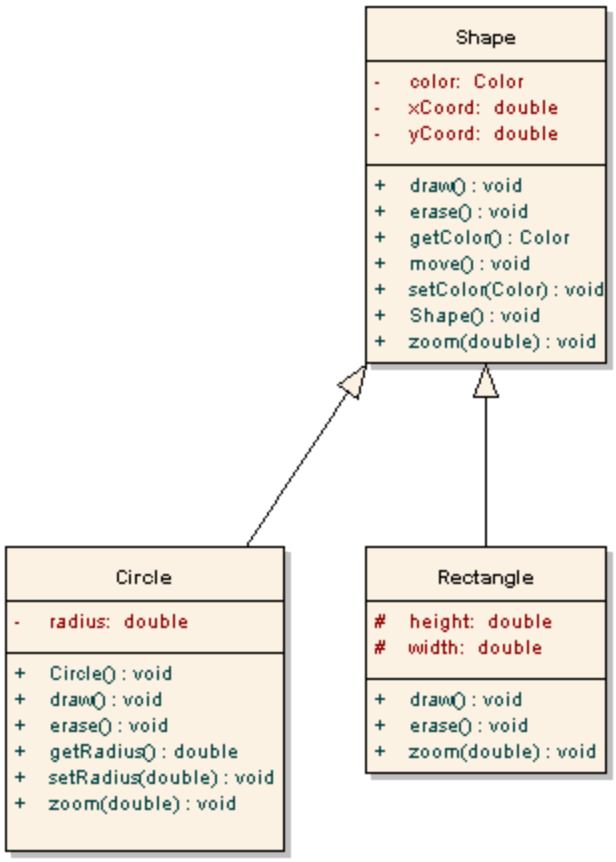
If you extend your Parent/Child relationship example further to include a Person class then it would make good sense for Person to be abstract. Parent is a concrete idea and so is child. Person is an abstract concept in reality as well as in code.

One benefit is that you **explicitly** define and protect the idea of the abstract class. When you declare a class as an abstract there's no way that you or anyone else using your code uses it incorrectly by instantiating it. This reasoning is similar to why we specify functions

and fields as public, private or protected. If you declare a function or member as private you are in effect protecting it from improper access from client code. Privates are meant to be used within the class and that's it. Abstract classes are meant to be inherited and that's that.

Now, *do you have to use abstract classes* and define functions and fields as private instead of public? **No, you don't**. But these concepts are provided to help keep code clean and well-organized. The abstract class is implemented in all object-oriented languages to my knowledge. If you look around you will see that C++, C#, VB.NET etc. all use this concept.

A better, specific example:



What does it mean, we cannot create a implementation of the class

BaseClass gets Overridden by child classes via inheritance

2. What are the types of comment in C# with examples?

Single line

Eg:

[csharp] //This is a Single line comment[/csharp]

ii. Multiple line (/* */)

Eg:

[csharp] /*This is a multiple line comment

We are in line 2

Last line of comment*/[/csharp]

iii. XML Comments (///).

Eg:

[csharp]/// summary;

/// Set error message for multilingual language.

/// summary[/csharp]

3. Can multiple catch blocks be executed?

No, Multiple catch blocks can't be executed. Once the proper catch code executed, the control is transferred to the finally block and then the code that follows the finally block gets executed.

4. What is the difference between public, static and void?

All these are access modifiers in C#. Public declared variables or methods are accessible anywhere in the application. Static declared variables or methods are globally accessible without creating an instance of the class. The compiler stores the address of the method as the

entry point and uses this information to begin execution before any objects are created. And Void is a type modifier that states that the method or variable does not return any value.

5. What is an object?

An object is an instance of a class through which we access the methods of that class. “New” keyword is used to create an object. A class that creates an object in memory will contain the information about the methods, variables and behavior of that class.

6. Define Constructors?

A constructor is a member function in a class that has the same name as its class. The constructor is automatically invoked whenever an object class is created. It constructs the values of data members while initializing the class.

7. What is Jagged Arrays?

The array which has elements of type array is called jagged array. The elements can be of different dimensions and sizes. We can also call jagged array as Array of arrays.

8. What is the difference between ref & out parameters?

An argument passed as ref must be initialized before passing to the method whereas out parameter needs not to be initialized before passing to a method.

9. What is the use of using statement in C#?

The using block is used to obtain a resource and use it and then automatically dispose of when the execution of block completed.

10. What is serialization?

When we want to transport an object through network then we have to convert the object into a stream of bytes. The process of converting an object into a stream of bytes is called Serialization. For an object to be serializable, it should inherit ISerialize Interface.

De-serialization is the reverse process of creating an object from a stream of bytes.

11. Can “this” be used within a static method?

We can’t use ‘This’ in a static method because we can only use static variables/methods in a static method.

12. What is difference between constants and read-only?

Constant variables are declared and initialized at compile time. The value can’t be changed afterwards. Read-only variables will be initialized only from the Static constructor of the class. Read only is used only when we want to assign the value at run time.

13. What is an interface class?

Interface is an abstract class which has only public abstract methods and the methods only have the declaration and not the definition. These abstract methods must be implemented in the inherited classes.

14. What are value types and reference types?

Value types are stored in the Stack whereas reference types stored on heap.
Value types:
[csharp] int, enum , byte, decimal, double, float, long[/csharp]

Reference Types:
[csharp] string , class, interface, object.[/csharp]

15. What are Custom Control and User Control?

Custom Controls are controls generated as compiled code (Dlls), those are easier to use and can be added to toolbox. Developers can drag and drop controls to their web forms. Attributes can be set at design time. We can easily add custom controls to Multiple Applications (If Shared

Dlls), If they are private then we can copy to dll to bin directory of web application and then add reference and can use them.
User Controls are very much similar to ASP include files, and are easy to create. User controls can't be placed in the toolbox and dragged – dropped from it. They have their design and code behind. The file extension for user controls is ascx.

16. What are sealed classes in C#?

We create sealed classes when we want to restrict the class to be inherited. Sealed modifier used to prevent derivation from a class. If we forcefully specify a sealed class as base class then a compile-time error occurs.

17. What is method overloading?

Method overloading is creating multiple methods with the same name with unique signatures in the same class. When we compile, the compiler uses overload resolution to determine the specific method to be invoke.

18. What is the difference between Array and Arraylist?

In an array, we can have items of the same type only. The size of the array is fixed. An arraylist is similar to an array but it doesn't have a fixed size.

19. Can a private virtual method be overridden?

No, because they are not accessible outside the class.

20. Describe the accessibility modifier "protected internal".

Protected Internal variables/methods are accessible within the same assembly and also from the classes that are derived from this parent class.

21. What are the differences between System.String and System.Text.StringBuilder classes?

System.String is immutable. When we modify the value of a string variable then a new memory is allocated to the new value and the previous memory allocation released. System.StringBuilder was designed to have concept of a mutable string where a variety of operations can be performed without allocation separate memory location for the modified string.

22. What's the difference between the System.Array.CopyTo() and System.Array.Clone() ?

Using Clone() method, the program creates a new array object containing all the elements in the original array and using CopyTo() method, all the elements of existing array copies into another existing array. Both the methods perform a shallow copy.

23. How can we sort the elements of the array in descending order?

Using Sort() methods followed by Reverse() method.

24. Write down the C# syntax to catch exception?

To catch an exception, we use try catch blocks. Catch block can have parameter of system.Exception type.

Eg:

[csharp]try
{
GetAllData();
}
catch(Exception ex)
{
}[/csharp]

In the above example, we can omit the parameter from catch statement.

25. What's the difference between an interface and abstract class?

Interfaces have all the methods having only declaration but no definition. In an abstract class, we can have some concrete methods. In an interface class, all the methods are public. An abstract class may have private methods.

26. What is the difference between Finalize() and Dispose() methods?

Dispose() is called when we want for an object to release any unmanaged resources with them. On the other hand Finalize() is used for the same purpose but it doesn't assure the garbage collection of an object.

27. What are circular references?

Circular reference is situation in which two or more resources are interdependent on each other causes the lock condition and make the resources unusable.

28. What are generics in C#.NET?

Generics are used to make reusable code classes to decrease the code redundancy, increase type safety and performance. Using generics, we can create collection classes. To create generic collection, System.Collections.Generic namespace should be used instead of classes such as ArrayList in the System.Collections namespace. Generics promotes the usage of parameterized types.

Collections

Keeping Stuff in C# Collection Classes

Many variables are for storing one thing, like a number or a sentence. An object can have a lot of properties, storing a lot of information about a single concept. Sometimes, though, you just have to keep track of a big list of stuff. That's where the collection classes come in.

The collection classes in C# are awesome because they can help you keep a list of items, or a group of keys and values, and provide searching, proctoring, and general maintenance tools for free. That's a lot easier than writing the two hundredth 'sort customers by last name' method this month.

C# has basic collection classes, generic collection classes, and concurrent collection classes. These last ones are for thread safe operations in a multiprocessor environment. All of these collection classes are useful if you know where and when to use them.

Regular collections

Regular collections store untyped stuff for you. If you don't know exactly what you're going to store, regular collections are for you. Most everything in these classes are stored as Object.

Class	Description
ArrayList	This is just a straightforward collection of objects whose size is initially declared, and that size is increased as needed (but not decreased).
HashTable	Just a two-column table of keys and values, where the pairs are linked for sorting a retrieval.
Queue	A first in, last out (FIFO) list of items. Great for storing program process stuff.
Stack	Like a Queue, but Last In First Out (LIFO).

Generic collections

C# 2.0 introduced generic collections. Generic collections allow you to store typed collections of items, as declared at runtime.

Class	Description
Dictionary<TKey, TValue>	Just a HastTable, except typed, so you don't have to store Object, Object. TKey and TValue are types, so you can store a list of <Int32, String> if you want.
List<T>	This is the new Array. It's a typed list of items. By far, the most common collection class used. It's the default. If you need a list of people, it's a List<Person>.
Queue<T>	Similar to the untyped Queue, but, well, typed!
SortedList<TKey, Value>	This awesome Dictionary implements IComparable so that it can be sorted.
Stack<T>	Just like the untyped Stack, except typed.

Concurrent collection classes

Finally, the Concurrent classes make sure that the objects you store in a list are available in any memory space, no matter what processor. It's a bigger deal than you think.

Class	Description

BlockingCollection<T>	It's a thread-safe class that will prevent one thread from altering a field that is accessible by another thread. Brilliant stuff, really.
ConcurrentDictionary<TKey, TValue>	Speaking of brilliant, this set of key/value pairs can be accessed by multiple threads.
ConcurrentQueue<T>	Remember that FIFO stack? Yeah. This one is generic AND thread safe.
ConcurrentStack<T>	The orchestrated collections are useful. They're beneficial for memory, for one thing. Anyway, this one is LIFO and thread-safe, too.

29. What is an object pool in .NET?

An object pool is a container having objects ready to be used. It tracks the object that is currently in use, total number of objects in the pool. This reduces the overhead of creating and re-creating objects.

30. List down the commonly used types of exceptions in .Net?

ArgumentException, ArgumentNullException , ArgumentOutOfRangeException, ArithmeticException, DivideByZeroException ,OverflowException , IndexOutOfRangeException ,InvalidCastException ,InvalidOperationException , IOEndOfStreamException , NullReferenceException , OutOfMemoryException , StackOverflowException etc.

31. What are Custom Exceptions?

Sometimes there are some errors that need to be handled as per user requirements. Custom exceptions are used for them and are used defined exceptions.

32. What are delegates?

Delegates are same as function pointers in C++ but the only difference is that they are type safe unlike function pointers. Delegates are required because they can be used to write much more generic type safe functions.

33. How do you inherit a class into another class in C#?

Colon is used as inheritance operator in C#. Just place a colon and then the class name.
[csharp] public class DerivedClass : BaseClass[/csharp]

34. What is the base class in .net from which all the classes are derived from?
[csharp]System.Object[/csharp]

35. What is the difference between method overriding and method overloading?
In method overriding, we change the method definition in the derived class that changes the method behavior. Method overloading is creating a method with the same name within the same class having different signatures.

36. What are the different ways a method can be overloaded?
Methods can be overloaded using different data types for parameter, different order of parameters, and different number of parameters.

37. Why can't you specify the accessibility modifier for methods inside the interface?
In an interface, we have virtual methods that do not have method definition. All the methods are there to be overridden in the derived class. That's why they all are public.

38. How can we set class to be inherited, but prevent the method from being over-ridden?
Declare the class as public and make the method sealed to prevent it from being overridden.

39. What happens if the inherited interfaces have conflicting method names?
Implement is up to you as the method is inside your own class. There might be problem when the methods from different interfaces expect different data, but as far as compiler cares you're okay.

40. What is the difference between a Struct and a Class?

Structs are value-type variables and classes are reference types. Structs stored on the stack, causes additional overhead but faster retrieval. Structs cannot be inherited.

Most importantly, a **struct** unlike a **class**, is a value type. So, while instances of a**class** are **stored** in the heap, instances of a **struct** are **stored** in the stack.

41. How to use nullable types in .Net?

Value types can take either their normal values or a null value. Such types are called nullable types.

[csharp]Int? someID = null;

If(someID.HasVAlue)

{

}

[/csharp]

42. How we can create an array with non-default values?

We can create an array with non-default values using Enumerable.Repeat.

43. What is difference between is and as operators in c#?

“is” operator is used to check the compatibility of an object with a given type and it returns the result as Boolean.

“as” operator is used for casting of object to a type or a class.

44. What’s a multicast delegate?

A delegate having multiple handlers assigned to it is called multicast delegate. Each handler is assigned to a method.

45. What are indexers in C# .NET?

Indexers are known as smart arrays in C#. It allows the instances of a class to be indexed in the same way as array.

Eg:

[csharp]public int this[int index] // Indexer declaration[/csharp]

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

“Throw” statement preserves original error stack whereas “throw ex” have the stack trace from their throw point. It is always advised to use “throw ex” because it provides more accurate error information.

47. What are C# attributes and its significance?

C# provides developers a way to define declarative tags on certain entities eg. Class, method etc. are called attributes. The attribute’s information can be retrieved at runtime using Reflection.

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

In singleton pattern, a class can only have one instance and provides access point to it globally.

Eg:

[csharp]

Public sealed class Singleton

{

Private static readonly Singleton _instance = new Singleton();

}

[/csharp]

49. What is the difference between directcast and ctype?

DirectCast is used to convert the type of an object that requires the run-time type to be the same as the specified type in DirectCast.

Ctype is used for conversion where the conversion is defined between the expression and the type.

50. Is C# code is managed or unmanaged code?

C# is managed code because Common language runtime can compile C# code to Intermediate language.

LINQ using the .ToList() ---- This is a Collection a List<T>

Types of Collections

Lists

Dictionaries

Sets

Index: The position in order index = 0 zero-based indexing ----

List Types in .NET:

T[] List<T>

Collection<T>

ReadOnlyCollection<T>

ObservableCollection<T>

IList<T> ----- Contract for index-based lists

Good for memory use Efficient for accessing elements

Dictionary<TKey, TValue>

IDictionary<TKey, TValue>

Hash Table is a Type Dictionary

var employees = new Dictionary<string, Employee>();

Dictionary is slower than a List with an Index.

Sets

Union Distinct operations are used to combine collections to make new collections and that is what a SET is or HashSet ISet

Sets DO NOT HAVE KEYS You only enumerate a set

Enumerating this using a foreach loop

All Collections allow Enumerating
Looking up items is not allowed in Sets - Linked lists, Stacks, Queues

Collection Operations Writing ----
Add an element
Remove an element
Lists Insert an element (replace an element)

LINQ is the great for these KINDS of derived operations (Language Integrated Query)

C# is a strongly typed language ----
List<int>
List<Employee>
List<Control>
List<T>

Generics did not exist till 2005

Arrays are strongly typed “ALWAYS BEEN that way”

int[] arrayOfInts = new int[];

2005 - 20010 I was doing SharePoint --- Bad Decision ----- I missed out on Generics, LINQ and Concurrent Collections

History 2002 - .NET

.NET 4.0 added Readonly Interfaces

Array = using System;

Old .NET 1.0 Collections = using System.Collections;
using System.Collections.Specialized

Core generic collections = using System.Collections.Generic;
using System.Collections.ObjectModel;

Concurrent collections = using System.Collections.Concurrent;

Immutable collections = using System.Collections.Immutable;

Arrays only weakness - Fixed Size

Arrays special syntax in C#:
int[] iHaveSquare Brackets Implemented inside the CLR itself

Arrays are implemented by .NET execution Engine

Fore Each iteration variable is “READ ONLY”!!!

.NET Types
Arrays and Collections = Reference ---------- Managed Heap

Value Types are on the Stack

Pretty Cool Array.binarySearch(sortedDays, “Sunday”);

Collection Interfaces

So, strings are immutable. What this means in application is that modifying a string creates new string data on the heap and modifies the pointer on the stack to point to the new location. The data on the heap that is not referenced from the stack will be cleared at garbage collection time, but until then it’s wasteful.

In comparison, StringBuilders are mutable. Modifications/additions to the stringbuilder will update the data in the heap instead of making copies of it each time.

```
StringBuilder sb = new StringBuilder;

sb.Add(“test”);

sb.Add(” and test2″);
```

That code will create a pointer on the stack and a data value on the heap, update that data value to add the second string and that’s it. Doing the same with strings would have left an abandoned string data value on the heap of value “test” waiting on the GC.

ASP.NET MVC

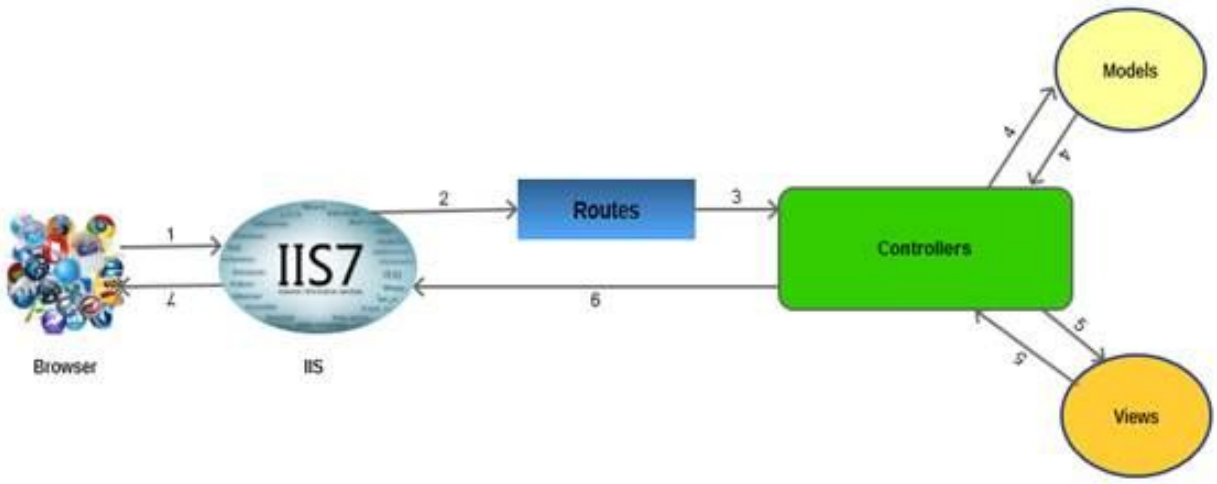
What is MVC?

MVC is a framework pattern that splits an application's implementation logic into three component roles: models, views, and controllers.

- Model: The business entity on which the overall application operates. Many applications use a persistent storage mechanism (such as a database) to store data. MVC does not specifically mention the data access layer because it is understood to be encapsulated by the Model.
- View: The user interface that renders the Model into a form of interaction.
- Controller: Handles a request from a View and updates the Model that results in a change of the Model's state.

To implement MVC in .NET, we need mainly three classes (`View`, `Controller` and the `Model`).

Explain MVC Architecture?



The architecture is self explanatory. The browser (as usual) sends a request to IIS, IIS searches for the route defined in MVC application and passes the request to the controller as per route, the controller communicates with the model and passes the populated model (entity) to View

(front end), Views are populated with model properties, and are rendered on the browser, passing the response to browser through IIS via controllers which invoked the particular View.

MVC Comment out a line

@* *@

Default Routing of MVC

```
routes.MapRoute(
            "Default",                                          // Route name
            "{controller}/{action}/{id}",                      // URL with parameters
            new { controller = "Home", action = "Index", id = "" }  // Parameter defaults
        );
```

Defined in the Global.asax.cs file

controller/action/id

What are the new features of MVC2?

ASP.NET MVC 2 was released in March 2010. Its main features are:

- Introduction of UI helpers with automatic scaffolding with customizable templates
- Attribute-based model validation on both client and server
- Strongly typed HTML helpers
- Improved Visual Studio tooling
- There were also lots of API enhancements and “pro” features, based on feedback from developers building a variety of applications on ASP.NET MVC 1, such as:
    - Support for partitioning large applications into areas
    - Asynchronous controllers support
    - Support for rendering subsections of a page/site using `Html.RenderAction`

- Lots of new helper functions, utilities, and API enhancements

What are the new features of MVC3?

ASP.NET MVC 3 shipped just 10 months after MVC 2 in Jan 2011. Some of the top features in MVC 3 included:

- The Razor view engine
- Support for .NET 4 Data Annotations
- Improved model validation
- Greater control and flexibility with support for dependency resolution and global action filters
- Better JavaScript support with unobtrusive JavaScript, jQuery Validation, and JSON binding
- Use of NuGet to deliver software and manage dependencies throughout the platform

What are the new features of MVC4?

Following are the top features of MVC4:

- ASP.NET Web API
- Enhancements to default project templates
- Mobile project template using jQuery Mobile
- Display Modes
- Task support for Asynchronous Controllers
- Bundling and minification

Explain “page lifecycle” of an ASP.NET MVC?

Following processes are performed by ASP.NET MVC page:

1. App initialization
2. Routing
3. Instantiate and execute controller

4. Locate and invoke controller action
5. Instantiate and render view

Advantages of MVC Framework?

1. Provides a clean separation of concerns between UI (Presentation layer), model (Transfer objects/Domain Objects/Entities) and Business Logic (Controller)
2. Easy to UNIT Test
3. Improved reusability of views/model. One can have multiple views which can point to the same model and vice versa
4. Improved structuring of the code

What do you mean by Separation of Concerns?

As per Wikipedia, 'the process of breaking a computer program into distinct features that overlap in functionality as little as possible'. MVC design pattern aims to separate content from presentation and data-processing from content.

Where do we see Separation of Concerns in MVC?

Between the data-processing (Model) and the rest of the application.
When we talk about Views and Controllers, their ownership itself explains separation. The views are just the presentation form of an application, it does not have to know specifically about the requests coming from controller. The Model is independent of View and Controllers, it only holds business entities that can be passed to any View by the controller as required for exposing them to the end user. The controller is independent of Views and Models, its sole purpose is to handle requests and pass it on as per the routes defined and as per the need of rendering views. Thus our business entities (model), business logic (controllers) and presentation logic (views) lie in logical/physical layers independent of each other.

What is Razor View Engine?generations

Razor is the first major update to render HTML in MVC3. Razor was designed specifically as a view engine syntax. It has one main focus: code-focused templating for HTML generation. Here's how that same markup would be generated using Razor:

Collapse | Copy Code

```
@model MvcMusicStore.Models.Genre
@{ViewBag.Title = "Browse Albums";}
<div class="genre">
<h3><em>@Model.Name</em> Albums</h3>
<ul id="album-list">
@foreach (var album in Model.Albums)
{
<li>
<a href="@Url.Action("Details", new { id = album.AlbumId })">
<img alt="@album.Title" src="@album.AlbumArtUrl" />
<span>@album.Title</span>
</a>
</li>
}
</ul>
</div>
```

The Razor syntax is easier to type, and easier to read. Razor doesn't have the XML-like heavy syntax. of the Web Forms view engine.

What is Unobtrusive JavaScript?

Unobtrusive JavaScript is a general term that conveys a general philosophy, similar to the term REST (Representational State Transfer). The high-level description is that unobtrusive JavaScript doesn't intermix JavaScript code in your page markup. For example, rather than hooking in via event attributes like `onclick` and `onsubmit`, the unobtrusive JavaScript attaches to elements by their ID or class, often based on the presence of other attributes (such as HTML5 data- attributes).

It's got semantic meaning, and all of it — the tag structure, element attributes, and so on — should have a precise meaning. Strewing JavaScript gunk across the page to facilitate interaction (I'm looking at you, `__doPostBack`!) harms the content of the document.

What is JSON Binding?

MVC 3 included JavaScript Object Notation (JSON) binding support via the new `JsonValueProviderFactory`, enabling the action methods to accept and model-bind data in JSON format. This is especially useful in advanced Ajax scenarios like client templates and data binding that need to post data back to the server.

What is Dependency Resolution?

MVC 3 introduced a new concept called a dependency resolver, which greatly simplified the use of dependency injection in your applications. This made it easier to decouple application components, making them more configurable and easier to test.
Support was added for the following scenarios:

- Controllers (registering and injecting controller factories, injecting controllers)
- Views (registering and injecting view engines, injecting dependencies into view pages)
- Action filters (locating and injecting filters)
- Model binders (registering and injecting)
- Model validation providers (registering and injecting)
- Model metadata providers (registering and injecting)
- Value providers (registering and injecting)

What are Display Modes in MVC4?

Display modes use a convention-based approach to allow selecting different views based on the browser making the request. The default view engine first looks for views with names ending with *.Mobile.cshtml* when the browser's user agent indicates a known mobile device. For example, if we have a generic view titled *Index.cshtml* and a mobile view titled

Index.Mobile.cshtml, MVC 4 will automatically use the mobile view when viewed in a mobile browser.

Additionally, we can register your own custom device modes that will be based on your own custom criteria — all in just one code statement. For example, to register a WinPhone device mode that would serve views ending with.*WinPhone.cshtml* to Windows Phone devices, you'd use the following code in the `Application_Start` method of your *Global.asax*:

Collapse | Copy Code

```
DisplayModeProvider.Instance.Modes.Insert(0, new DefaultDisplayMode("WinPhone")
{
ContextCondition = (context => context.GetOverriddenUserAgent().IndexOf
("Windows Phone OS", StringComparison.OrdinalIgnoreCase) >= 0)
});
```

What is AuthConfig.cs in MVC4?

AuthConfig.cs is used to configure security settings, including sites for OAuth login.

What is BundleConfig.cs in MVC4?

BundleConfig.cs in MVC4 is used to register bundles used by the bundling and minification system. Several bundles are added by default, including jQuery, jQueryUI, jQuery validation, Modernizr, and default CSS references.

What is FilterConfig.cs in MVC4?

This is used to register global MVC filters. The only filter registered by default is the `HandleErrorAttribute`, but this is a great place to put other filter registrations.

What is RouteConfig.cs in MVC4?

RouteConfig.cs holds the granddaddy of the MVC *config* statements, Route configuration.

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace MvcApplication1
{
    // Note: For instructions on enabling IIS6 or IIS7 classic mode,
    // visit http://go.microsoft.com/?LinkId=9394801

    public class MvcApplication : System.Web.HttpApplication
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                "Default",                                              // Route name
                "{controller}/{action}/{id}",                           // URL with parameters
                new { controller = "Home", action = "Index", id = "" }  // Parameter defaults
            );

        }

        protected void Application_Start()
        {
            RegisterRoutes(RouteTable.Routes);
        }
    }
}
```

What is WebApiConfig.cs in MVC4?

Used to register Web API routes, as well as set any additional Web API configuration settings.

What's new in adding controller in MVC4 application?

Previously (in MVC3 and MVC2), the Visual Studio Add Controller menu item only displayed when we right-clicked on the *Controllers* folder. However, the use of the *Controllers* folder was purely for organization. (MVC will recognize any class that implements the `IController` interface as a `Controller`, regardless of its location in your application.) The MVC 4 Visual Studio tooling has been modified to display the Add Controller menu item for any folder in your MVC project. This allows us to organize your controllers however you would like, perhaps separating them into logical groups or separating MVC and Web API controllers.

What are the software requirements of ASP.NET MVC4 application?

MVC 4 runs on the following Windows client operating systems:

- Windows 7
- Windows 8
- Windows 10

It runs on the following server operating systems:

- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2 and above

MVC 4 development tooling is included with Visual Studio 2012 and can be installed on Visual Studio 2010 SP1/Visual Web Developer 2010 Express SP1.

What are the various types of Application Templates used to create an MVC application?

The various templates are as follows:

1. The Internet Application template: This contains the beginnings of an MVC web application — enough so that you can run the application immediately after creating it

and see a few pages. This template also includes some basic account management functions which run against the ASP.NET Membership.

2. The Intranet Application template: The Intranet Application template was added as part of the ASP.NET MVC 3 Tools Update. It is similar to the Internet Application template, but the account management functions run against Windows accounts rather than the ASP.NET Membership system.
3. The Basic template: This template is pretty minimal. It still has the basic folders, CSS, and MVC application infrastructure in place, but no more. Running an application created using the Empty template just gives you an error message.
4. Why use Basic template? The Basic template is intended for experienced MVC developers who want to setup and configure things exactly how they want them.
5. The Empty template: The Basic template used to be called the Empty template, but developers complained that it wasn't quite empty enough. With MVC 4, the previous Empty template was renamed Basic, and the new Empty template is about as empty as we can get. It has the assemblies and basic folder structure in place, but that's about it.
6. The Mobile Application template: The Mobile Application template is preconfigured with jQuery Mobile to jump-start creating a mobile only website. It includes mobile visual themes, a touch-optimized UI, and support for Ajax navigation.
7. The Web API template: ASP.NET Web API is a framework for creating HTTP services. The Web API template is similar to the Internet Application template but is streamlined for Web API development. For instance, there is no user account management functionality, as Web API account management is often significantly different from standard MVC account management. Web API functionality is also available in the other MVC project templates, and even in non-MVC project types.

What are the default Top level directories created when adding MVC4 application?

Default Top level Directories are:

Collapse | Copy Code

```
DIRECTORY            PURPOSE
/Controllers         To put Controller classes that handle URL requests
/Models              To put classes that represent and manipulate data and business
objects
/Views               To put UI template files that are responsible for rendering output
like HTML.
/Scripts             To put JavaScript library files and scripts (.js)
/Images              To put images used in your site
/Content             To put CSS and other site content, other than scripts and images
/Filters             To put filter code.
/App_Data            To store data files you want to read/write
/App_Start           To put configuration code for features like Routing, Bundling, Web
API.
```

What is namespace of ASP.NET MVC?

ASP.NET MVC namespaces as well as classes are located in assembly `System.Web.Mvc`.
Note: Some of the content has been taken from various books/articles.

What is System.Web.Mvc namespace?

This namespace contains classes and interfaces that support the MVC pattern for ASP.NET Web applications. This namespace includes classes that represent controllers, controller factories, action results, views, partial views, and model binders.

What is System.Web.Mvc.Ajax namespace?

`System.Web.Mvc.Ajax` namespace contains classes that supports Ajax scripting in an ASP.NET MVC application. The namespace includes support for Ajax scripts and Ajax option settings as well.

What is System.Web.Mvc.Async namespace?

`System.Web.Mvc.Async` namespace contains classes and interfaces that support asynchronous actions in an ASP.NET MVC application.

What is System.Web.Mvc.Html namespace?

`System.Web.Mvc.Html` namespace contains classes that help render HTML controls in an MVC application. This namespace includes classes that support forms, input controls, links, partial views, and validation.

What is ViewData, ViewBag and TempData?

MVC provides us `ViewData`, `ViewBag` and `TempData` for passing data from controller, view and in next requests as well. `ViewData` and `ViewBag` are similar to some extent but `TempData` performs additional roles.

Types of way to transfer data

ViewData

ViewBag

TempData

ViewModel

What are the roles and similarities between ViewData and ViewBag?

- Maintains data when moving from controller to view
- Passes data from controller to respective view
- Their value becomes null when any redirection occurs, because their role is to provide a way to communicate between controllers and views. It's a communication mechanism within the server call.

What are the differences between ViewData and ViewBag? (taken from a blog)

- `ViewData` is a dictionary of objects that is derived from `ViewDataDictionary` class and accessible using `strings` as keys.
- `ViewBag` is a dynamic property that takes advantage of the new dynamic features in C# 4.0.
- `ViewData` requires typecasting for complex data type and checks for null values to avoid error.
- `ViewBag` doesn't require typecasting for complex data type.

NOTE: Although there might not be a technical advantage to choosing one format over the other, there are some critical differences to be aware of between the two syntaxes.
One obvious difference is that `ViewBag` works only when the key being accessed is a valid C# identifier. For example, if you place a value in `ViewData["KeyWith Spaces"]`, you can't access that value using `ViewBag` because the code won't compile.
Another key issue to be aware of is that dynamic values cannot be passed in as parameters to extension methods. The C# compiler must know the real type of every parameter at compile time in order for it to choose the correct extension method.
If any parameter is dynamic, compilation will fail. For example, this code will always fail: `@Html.TextBox("name", ViewBag.Name)`. To work around this, either use `ViewData["Name"]` or cast the value to a specific type: `(string) ViewBag.Name`.

What is TempData?

`TempData` is a dictionary derived from the `TempDataDictionary` class and stored in short lives session. It is a `string` key and `object` value.
It keeps the information for the time of an HTTP Request. This means only from one page to another. It helps to maintain data when we move from one controller to another controller or from one action to other action. In other words, when we redirect `Tempdata` helps to maintain data between those redirects. It internally uses session variables. Temp data use during the

current and subsequent request only means it is used when we are sure that the next request will be redirecting to next view. It requires typecasting for complex data type and checks for null values to avoid error. Generally it is used to store only one time messages like error messages, validation messages.

How can you define a dynamic property with the help of viewbag in ASP.NET MVC?

Assign a key name with syntax, `ViewBag.[Key]=[ Value]` and value using equal to operator.

For example, you need to assign list of students to the dynamic `Students` property of `ViewBag`.

Collapse | Copy Code

```
List<string> students = new List<string>();
countries.Add("Akhil");
countries.Add("Ekta");
ViewBag.Students = students;
//Students is a dynamic property associated with ViewBag.
```

Note: Some of the content has been taken from various books/articles.

What is ViewModel (taken from stackoverflow)?

Accepted a view model represents data that you want to have displayed on your view/page. Let's say that you have an `Employee` class that represents your employee domain model and it contains the following 4 properties:

Collapse | Copy Code

```
public class Employee : IEntity
{
    public int Id { get; set; }  // Employee's unique identifier
    public string FirstName { get; set; }  // Employee's first name
    public string LastName { get; set; }  // Employee's last name
    public DateTime DateCreated { get; set; }  // Date when employee was created
}
```

View models differ from domain models in that view models only contain the data (represented by properties) that you want to use on your view. For example, let's say that you want to add a new employee record, your view model might look like this:

Collapse | Copy Code

```
public class CreateEmployeeViewModel
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
}
```

As you can see, it only contains 2 of the properties of the employee domain model. Why is this you may ask? `Id` might not be set from the view, it might be auto generated by the `Employee` table. And `DateCreated` might also be set in the stored procedure or in the service layer of your application. So `Id` and `DateCreated` is not needed in the view model.

When loading the view/page, the create action method in your employee controller will create an instance of this view model, populate any fields if required, and then pass this view model to the view:

Collapse | Copy Code

```
public class EmployeeController : Controller
{
    private readonly IEmployeeService employeeService;

    public EmployeeController(IEmployeeService employeeService)
    {
        this.employeeService = employeeService;
    }

    public ActionResult Create()
    {
        CreateEmployeeViewModel viewModel = new CreateEmployeeViewModel();

        return View(viewModel);
    }
```

```
    public ActionResult Create(CreateEmployeeViewModel viewModel)
    {
        // Do what ever needs to be done before adding the employee to the database
    }
}
```

Your view might look like this (assuming you are using ASP.NET MVC3 and razor):

Collapse | Copy Code

```
@model MyProject.Web.ViewModels.ProductCreateViewModel

<table>
    <tr>
        <td><b>First Name:</b></td>
        <td>@Html.TextBoxFor(x => x.FirstName, new { maxlength = "50", size = "50" })
            @Html.ValidationMessageFor(x => x.FirstName)
        </td>
    </tr>
    <tr>
        <td><b>Last Name:</b></td>
        <td>@Html.TextBoxFor(x => x.LastName, new { maxlength = "50", size = "50" })
            @Html.ValidationMessageFor(x => x.LastName)
        </td>
    </tr>
</table>
```

Validation would thus be done only on `FirstName` and `LastName`. Using Fluent Validation, you might have validation like this:

Collapse | Copy Code

```
public class CreateEmployeeViewModelValidator :
AbstractValidator<CreateEmployeeViewModel>
{
    public CreateEmployeeViewModelValidator()
    {
        RuleFor(x => x.FirstName)
            .NotEmpty()
```

```
                .WithMessage("First name required")
                .Length(1, 50)
                .WithMessage("First name must not be greater than 50 characters");

        RuleFor(x => x.LastName)
                .NotEmpty()
                .WithMessage("Last name required")
                .Length(1, 50)
                .WithMessage("Last name must not be greater than 50 characters");
    }
}
```

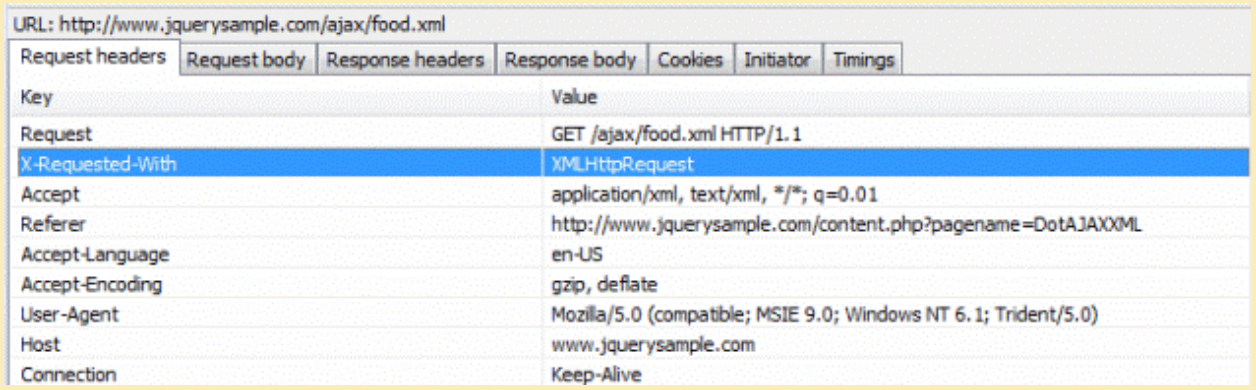
The key thing to remember is that the view model only represents the data that you want use. You can imagine all the unneccessary code and validation if you have a domain model with 30 properties and you only want to update a single value. Given this scenario, you would only have this one value/property in the view model and not the whole domain object.

How do you check for AJAX request with C# in MVC.NET?

The solution is independent of MVC.NET framework and is global across server side technologies. Most modern AJAX applications utilize `XmlHTTPRequest` to send async request to the server. Such requests will have distinct request header:

Collapse | Copy Code

```
X-Requested-With = XMLHTTPREQUEST
```



MVC.NET provides helper function to check for ajax requests which internally inspects X-Requested-With request header to set IsAjax flag.

What are Scaffold templates?

These templates use the Visual Studio T4 templating system to generate a view based on the model type selected. Scaffolding in ASP.NET MVC can generate the boilerplate code we need for create, read, update, and delete (CRUD) functionality in an application. The scaffolding templates can examine the type definition for, and then generate a controller and the controller's associated views. The scaffolding knows how to name controllers, how to name views, what code needs to go in each component, and where to place all these pieces in the project for the application to work.

What are the types of Scaffolding Templates?

Various types are as follows:

Collapse | Copy Code

```
SCAFFOLD            DESCRIPTION
Empty               Creates empty view. Only the model type is specified using the
model syntax.
Create              Creates a view with a form for creating new instances of the model.
                    Generates a label and input field for each property of the model
type.
Delete              Creates a view with a form for deleting existing instances of the
model.
                    Displays a label and the current value for each property of the
model.
Details             Creates a view that displays a label and the value for each
property of the
                    model type.
Edit                Creates a view with a form for editing existing instances of the
model.
                    Generates a label and input field for each property of the model
type.
List                Creates a view with a table of model instances. Generates a column
                    for each property of the model type. Make sure to pass an
                    IEnumerable<YourModelType> to this view from your action method.
```

```
                    The view also contains links to actions for performing the
                    create/edit/delete operation.
```

Show an example of difference in syntax in Razor and WebForm View?

Collapse | Copy Code

```
Razor <span>@model.Message</span>
Web Forms <span><%: model.Message %></span>
```

Code expressions in Razor are always HTML encoded. This Web Forms syntax also automatically HTML encodes the value.

What are Code Blocks in Views?

Unlike code expressions, which are evaluated and outputted to the response, blocks of code are simply sections of code that are executed. They are useful for declaring variables that we may need to use later.

Razor

Collapse | Copy Code

```
@{
int x = 123;
string y = ?because.?;
}
```

Web Forms

Collapse | Copy Code

```
<%
int x = 123;
string y = "because.";
%>
```

What is HelperPage.IsAjax Property?

`HelperPage.IsAjax` gets a value that indicates whether Ajax is being used during the request of the Web page.

- Namespace: `System.Web.WebPages`
- Assembly: *System.Web.WebPages.dll*

However, the same can be achieved by checking requests header directly:

Collapse | Copy Code

```
Request["X-Requested-With"] == "XmlHttpRequest".
```

Explain combining text and markup in Views with the help of an example?

This example shows what intermixing text and markup looks like using Razor as compared to Web Forms:

Razor

Collapse | Copy Code

```
@foreach (var item in items) {
<span>Item @item.Name.</span>
}
```

Web Forms

Collapse | Copy Code

```
<% foreach (var item in items) { %>
<span>Item <%: item.Name %>.</span>
<% } %>
```

Explain Repository Pattern in ASP.NET MVC?

In simple terms, a repository basically works as a mediator between our business logic layer and our data access layer of the application. Sometimes, it would be troublesome to expose the data access mechanism directly to business logic layer, it may result in redundant code for

accessing data for similar entities or it may result in a code that is hard to test or understand. To overcome these kinds of issues, and to write an Interface driven and test driven code to access data, we use Repository Pattern. The repository makes queries to the data source for the data, thereafter maps the data from the data source to a business entity/domain object, finally and persists the changes in the business entity to the data source. According to MSDN, a repository separates the business logic from the interactions with the underlying data source or Web service. The separation between the data and business tiers has three benefits:

- It centralizes the data logic or Web service access logic.
- It provides a substitution point for the unit tests.
- It provides a flexible architecture that can be adapted as the overall design of the application evolves.

In Repository, we write our whole business logic of CRUD operations with the help of Entity Framework classes, that will not only result in meaningful test driven code but will also reduce our controller code of accessing data.

How can you call a JavaScript function/method on the change of Dropdown List in MVC?

Create a JavaScript method:

Collapse | Copy Code

```
<script type="text/javascript">
function selectedIndexChanged() {
}
</script>
Invoke the method:
<%:Html.DropDownListFor(x => x.SelectedProduct,
new SelectList(Model.Users, "Value", "Text"),
"Please Select a User", new { id = "ddlUsers",
onchange="selectedIndexChanged()" })%>
```

Explain Routing in MVC?

A route is a URL pattern that is mapped to a handler. The handler can be a physical file, such as an *.aspx* file in a Web Forms application. Routing module is responsible for mapping incoming browser requests to particular MVC controller actions.
Routing within the ASP.NET MVC framework serves two main purposes:

- It matches incoming requests that would not otherwise match a file on the file system and maps the requests to a controller action.
- It constructs outgoing URLs that correspond to controller actions.

How route table is created in ASP.NET MVC?

When an MVC application first starts, the `Application_Start()` method in *global.asax* is called. This method calls the `RegisterRoutes()` method. The `RegisterRoutes()` method creates the route table for MVC application.

What are Layouts in ASP.NET MVC Razor?

Layouts in Razor help maintain a consistent look and feel across multiple views within our application. As compared to Web Forms, layouts serve the same purpose as master pages, but offer both a simpler syntax and greater flexibility.
We can use a layout to define a common template for your site (or just part of it). This template contains one or more placeholders that the other views in your application provide content for. In some ways, it's like an abstract base class for your views.
E.g. declared at the top of view as:

Collapse | Copy Code

```
@{
        Layout = "~/Views/Shared/SiteLayout.cshtml";
}
```

What is ViewStart?

For group of views that all use the same layout, this can get a bit redundant and harder to maintain.

The *_ViewStart.cshtml* page can be used to remove this redundancy. The code within this file is executed before the code in any view placed in the same directory. This file is also recursively applied to any view within a subdirectory.

When we create a default ASP.NET MVC project, we find there is already a *_ViewStart .cshtml* file in the *Views*directory. It specifies a default layout:

Collapse | Copy Code

```
@{
Layout = "~/Views/Shared/_Layout.cshtml";
}
```

Because this code runs before any view, a view can override the `Layout` property and choose a different one. If a set of views shares common settings, the *_ViewStart.cshtml* file is a useful place to consolidate these common view settings. If any view needs to override any of the common settings, the view can set those values to another value.

Note: Some of the content has been taken from various books/articles.

What are HTML Helpers?

HTML helpers are methods we can invoke on the HTML property of a view. We also have access to URL helpers (via the `Url` property), and AJAX helpers (via the Ajax property). All these helpers have the same goal: to make views easy to author. The URL helper is also available from within the controller. Most of the helpers, particularly the HTML helpers, output HTML markup. For example, the `BeginForm` helper is a helper we can use to build a robust form tag for our search form, but without using lines and lines of code:

Collapse | Copy Code

```
@using (Html.BeginForm("Search", "Home", FormMethod.Get)) {
<input type="text" name="q" />
```

```
<input type="submit" value="Search" />
}
```

What is Html.ValidationSummary?

The `ValidationSummary` helper displays an unordered list of all validation errors in the `ModelState` dictionary. The Boolean parameter you are using (with a value of `true`) is telling the helper to exclude property-level errors. In other words, you are telling the summary to display only the errors in `ModelState` associated with the model itself, and exclude any errors associated with a specific model property. We will be displaying property-level errors separately. Assume you have the following code somewhere in the controller action rendering the edit view:

Collapse | Copy Code

```
ModelState.AddModelError("", "This is all wrong!");
ModelState.AddModelError("Title", "What a terrible name!");
```

The first error is a model-level error, because you didn't provide a key (or provided an empty key) to associate the error with a specific property. The second error you associated with the `Title` property, so in your view it will not display in the validation summary area (unless you remove the parameter to the helper method, or change the value to `false`). In this scenario, the helper renders the following HTML:

Collapse | Copy Code

```
<div class="validation-summary-errors">
<ul>
<li>This is all wrong!</li>
</ul>
</div>
```

Other overloads of the `ValidationSummary` helper enable you to provide header text and set specific HTML attributes.

NOTE: By convention, the `ValidationSummary` helper renders the CSS class validation-summary-errors along with any specific CSS classes you provide. The default MVC project template includes some styling to display these items in red, which you can change in *styles.css*.

What are Validation Annotations?

Data annotations are attributes you can find in `System.ComponentModel.DataAnnotations` namespace. These attributes provide server-side validation, and the framework also supports client-side validation when you use one of the attributes on a model property. You can use four attributes in the `DataAnnotations` namespace to cover common validation scenarios,
Required, String Length, Regular Expression, Range.

What HtmlEncoder.Default.Encode

Used for ?

- to protect the app from malicious input (namely JavaScript).

Interpolated String example

```
public string Welcome(string name, int numTimes = 1)
{
    return HtmlEncoder.Default.Encode($"Hello {name}, NumTimes is: {numTimes}");
}
```

What is Html.Partial?

The `Partial` helper renders a partial view into a `string`. Typically, a `partial` view contains reusable markup you want to render from inside multiple different views. `Partial` has four overloads:

Collapse | Copy Code

```
public void Partial(string partialViewName);
public void Partial(string partialViewName, object model);
public void Partial(string partialViewName, ViewDataDictionary viewData);
public void Partial(string partialViewName, object model,
ViewDataDictionary viewData);
```

What is Html.RenderPartial?

The `RenderPartial` helper is similar to `Partial`, but `RenderPartial` writes directly to the response output stream instead of returning a `string`. For this reason, you must place `RenderPartial` inside a code block instead of a code expression. To illustrate, the following two lines of code render the same output to the output stream:

Collapse | Copy Code

```
@{Html.RenderPartial("AlbumDisplay "); }
@Html.Partial("AlbumDisplay ")
```

If they are the same, then which one to use?

In general, you should prefer `Partial` to `RenderPartial` because `Partial` is more convenient (you don't have to wrap the call in a code block with curly braces). However, `RenderPartial` may result in better performance because it writes directly to the response stream, although it would require a lot of use (either high site traffic or repeated calls in a loop) before the difference would be noticeable.

How do you return a partial view from controller?

Collapse | Copy Code

```
return PartialView(options); //options could be Model or View name
```

What are different ways of returning a View?

There are different ways for returning/rendering a view in MVC Razor. E.g. return `View()`, return`RedirectToAction()`, return `Redirect()` and return `RedirectToRoute()`.

jQuery

Q1. What is jQuery?

Ans: jQuery is fast, lightweight and feature-rich client side JavaScript Library/Framework which helps in to traverse HTML DOM, make animations, add Ajax interaction, manipulate the page content, change the style and provide cool UI effect. It is one of the most popular client side library and as per a survey it runs on every second website.

Q2. Why do we use jQuery?

Ans: Due to following advantages.

- Easy to use and learn.
- Easily expandable.
- Cross-browser support (IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+)
- Easy to use for DOM manipulation and traversal.
- Large pool of built in methods.
- AJAX Capabilities.
- Methods for changing or applying CSS, creating animations.
- Event detection and handling.
- Tons of plug-ins for all kind of needs.

Q3. How JavaScript and jQuery are different?

Ans: JavaScript is a language While jQuery is a library built in the JavaScript language that helps to use the JavaScript language.

Q4. Is jQuery replacement of Java Script?

Ans: No. jQuery is not a replacement of JavaScript. jQuery is a different library which is written on top of JavaScript. jQuery is a lightweight JavaScript library that emphasizes interaction between JavaScript and HTML.

Q5. Is jQuery a library for client scripting or server scripting?

Ans. Client side scripting.

Q6. Is jQuery a W3C standard?

Ans: No. jQuery is not a W3C standard.

Q7. What is the basic need to start with jQuery?

Ans: To start with jQuery, one need to make reference of it's library. The latest version of jQuery can be downloaded from jQuery.com.

Q8. Which is the starting point of code execution in jQuery?

Ans: The starting point of jQuery code execution is $(document).ready() function which is executed when DOM is loaded.

Q9. What does dollar sign ($) means in jQuery?

Ans: Dollar Sign is nothing but it's an alias for JQuery. Take a look at below jQuery code.

```
$(document).ready(function(){
});
```

Over here $ sign can be replaced with "jQuery" keyword.

Collapse | Copy Code

```
jQuery(document).ready(function(){
});
```

Q10. Can we have multiple document.ready() function on the same page?

Ans: YES. We can have any number of document.ready() function on the same page.

Q11. Can we use our own specific character in the place of $ sign in jQuery?

Ans: Yes. It is possible using jQuery.noConflict().

Q12. Is it possible to use other client side libraries like MooTools, Prototype along with jQuery?

Ans: Yes.

Q13. What is jQuery.noConflict?

Ans: As other client side libraries like MooTools, Prototype can be used with jQuery and they also use $() as their global function and to define variables. This situation creates conflict as `$()` is used by jQuery and other library as their global function. To overcome from such situations, jQuery has introduced `jQuery.noConflict()`.

Collapse | Copy Code

```
jQuery.noConflict();
// Use jQuery via jQuery(...)
jQuery(document).ready(function(){
   jQuery("div").hide();
});
```

You can also use your own specific character in the place of $ sign in jQuery.

Collapse | Copy Code

```
var $j = jQuery.noConflict();
// Use jQuery via jQuery(...)
$j(document).ready(function(){
   $j("div").hide();
});
```

Q14. Is there any difference between body onload() and document.ready() function?

Ans: `document.ready()` function is different from body `onload()` function for 2 reasons.

1. We can have more than one `document.ready()` function in a page where we can have only one body `onload` function.
2. `document.ready()` function is called as soon as DOM is loaded where `body.onload()` function is called when everything gets loaded on the page that includes DOM, images and all associated resources of the page.

Q15. What is the difference between .js and .min.js?

Ans: jQuery library comes in 2 different versions Production and Deployment. The deployment version is also known as minified version. So .min.js is basically the minified version of jQuery library file. Both the files are same as far as functionality is concerned. but .min.js is quite small in size so it loads quickly and saves bandwidth.

Q16. Why there are two different version of jQuery library?

Ans: jQuery library comes in 2 different versions.

1. Production
2. Deployment

The production version is quite useful at development time as jQuery is open source and if you want to change something then you can make those changes in production version. But the deployment version is minified version or compressed version so it is impossible to make changes in it. Because it is compressed, so its size is very less than the production version which affects the page load time.

Q17. What is a CDN?

Ans: A content delivery network or content distribution network (CDN) is a large distributed system of servers deployed in multiple data centers across the Internet. The goal of a CDN is to serve content to end-users with high availability and high performance.

Q18. Which are the popular jQuery CDN? and what is the advantage of using CDN?

Ans: There are 3 popular jQuery CDNs.

1. 1. Google.
2. 2. Microsoft
3. 3. jQuery.

Advantage of using CDN.

- It reduces the load from your server.
- It saves bandwidth. jQuery framework will load faster from these CDN.

- The most important benefit is it will be cached, if the user has visited any site which is using jQuery framework from any of these CDN

Q19. How to load jQuery from CDN?

Ans: Below is the code to load jQuery from all 3 CDNs.

Code to load jQuery Framework from Google CDN

Collapse | Copy Code

```
<script type="text/javascript"
    src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
```

Code to load jQuery Framework from Microsoft CDN

Collapse | Copy Code

```
<script type="text/javascript"
    src="http://ajax.microsoft.com/ajax/jquery/jquery-1.9.1.min.js">
</script>
```

Code to load jQuery Framework from jQuery Site(EdgeCast CDN)

Collapse | Copy Code

```
<script type="text/javascript"
    src="http://code.jquery.com/jquery-1.9.1.min.js">
</script>
```

Q20. How to load jQuery locally when CDN fails?

Ans: It is a good approach to always use CDN but sometimes what if the CDN is down (rare possibility though) but you never know in this world as anything can happen.

Below given jQuery code checks whether jQuery is loaded from Google CDN or not, if not then it references the jQuery.js file from your folder.

Collapse | Copy Code

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script type="text/javascript">
if (typeof jQuery == 'undefined')
{
  document.write(unescape("%3Cscript src='Scripts/jquery.1.9.1.min.js'
type='text/javascript'%3E%3C/script%3E"));
}
</script>
```

It first loads the jQuery from Google CDN and then check the jQuery object. If jQuery is not loaded successfully then it will references the jQuery.js file from hard drive location. In this example, the jQuery.js is loaded from Scripts folder.

Q21. What are selectors in jQuery and how many types of selectors are there?

Ans: To work with an element on the web page, first we need to find them. To find the html element in jQuery we use selectors. There are many types of selectors but basic selectors are:

- Name: Selects all elements which match with the given element Name.
- #ID: Selects a single element which matches with the given ID
- .Class: Selects all elements which match with the given Class.
- Universal (*): Selects all elements available in a DOM.
- Multiple Elements E, F, G: Selects the combined results of all the specified selectors E, F or G.
- Attribute Selector: Select elements based on its attribute value.

Q22. How do you select element by ID in jQuery?

Ans: To select element use ID selector. We need to prefix the id with "#" (hash symbol). For example, to select element with ID "txtName", then syntax would be,

Collapse | Copy Code

```
$('#txtName')
```

Q23. What does $("div") will select?

Ans: This will select all the div elements on page.

Q24. How to select element having a particular class (".selected")?

Ans: `$('.selected')`. This selector is known as class selector. We need to prefix the class name with "." (dot).

Q25. What does $("div.parent") will select?

Ans: All the div element with parent class.

Q26. What are the fastest selectors in jQuery?

Ans: ID and element selectors are the fastest selectors in jQuery.

Q27. What are the slow selectors in jQuery?

Ans: class selectors are the slow compare to ID and element.

Q28. How jQuery selectors are executed?

Ans: Your last selectors is always executed first. For example, in below jQuery code, jQuery will first find all the elements with class "`.myCssClass`" and after that it will reject all the other elements which are not in "p#elmID".

Collapse | Copy Code

```
$("p#elmID .myCssClass");
```

Q29. Which is fast document.getElementByID('txtName') or $('#txtName').?

Ans: Native JavaScipt is always fast. jQuery method to select txtName "`$('#txtName')`" will internally makes a call to `document.getElementByID('txtName')`. As jQuery is written on top of JavaScript and it internally uses JavaScript only So JavaScript is always fast.

Q30. Difference between $(this) and 'this' in jQuery?

Ans: this and `$(this)` refers to the same element. The only difference is the way they are used. 'this' is used in traditional sense, when 'this' is wrapped in `$()` then it becomes a jQuery object and you are able to use the power of jQuery.

Collapse | Copy Code

```
$(document).ready(function(){
    $('#spnValue').mouseover(function(){
        alert($(this).text());
  });
});
```

In below example, this is an object but since it is not wrapped in `$()`, we can't use jQuery method and use the native JavaScript to get the value of span element.

Collapse | Copy Code

```
$(document).ready(function(){
    $('#spnValue').mouseover(function(){
        alert(this.innerText);
```

```
  });
});
```

Q31. How do you check if an element is empty?

Ans: There are 2 ways to check if element is empty or not. We can check using ":empty" selector.

Collapse | Copy Code

```
$(document).ready(function(){
    if ($('#element').is(':empty')){
        //Element is empty
  }
});
```

And the second way is using the "$.trim()" method.

Collapse | Copy Code

```
$(document).ready(function(){
    if($.trim($('#element').html())=='') {
        //Element is empty
  }
});
```

Q32. How do you check if an element exists or not in jQuery?

Ans: Using jQuery length property, we can ensure whether element exists or not.

Collapse | Copy Code

```
$(document).ready(function(){
    if ($('#element').length > 0){
        //Element exists
  });
});
```

Q33. What is the use of jquery .each() function?

Ans: The `$.each()` function is used to iterate over a jQuery object. The `$.each()` function can be used to iterate over any collection, whether it is an object or an array.

Q34. What is the difference between jquery.size() and jquery.length?

Ans: jQuery `.size()` method returns number of element in the object. But it is not preferred to use the `size()` method as jQuery provide `.length` property and which does the same thing. But the `.length` property is preferred because it does not have the overhead of a function call.

Q35. What is the difference between $('div') and $(' ') in jQuery?

Ans: $(' ') : This creates a new div element. However this is not added to DOM tree unless you don't append it to any DOM element.

$('div') : This selects all the div element present on the page.

Q36. What is the difference between parent() and parents() methods in jQuery?

Ans: The basic difference is the `parent()` function travels only one level in the DOM tree, where parents() function search through the whole DOM tree.

Q37. What is the difference between eq() and get() methods in jQuery?

Ans: `eq()` returns the element as a jQuery object. This method constructs a new jQuery object from one element within that set and returns it. That means that you can use jQuery functions on it.

`get()` return a DOM element. The method retrieve the DOM elements matched by the jQuery object. But as it is a DOM element and it is not a jQuery-wrapped object. So jQuery functions can't be used. Find out more here.

Q38. How do you implement animation functionality?

Ans: The .animate() method allows us to create animation effects on any numeric CSS property. This method changes an element from one state to another with CSS styles. The CSS property value is changed gradually, to create an animated effect.

Syntax is:

Collapse | Copy Code

```
(selector).animate({styles},speed,easing,callback)
```

- styles: Specifies one or more CSS properties/values to animate.
- duration: Optional. Specifies the speed of the animation.
- easing: Optional. Specifies the speed of the element in different points of the animation. Default value is "swing".
- callback: Optional. A function to be executed after the animation completes.

Simple use of animate function is,

Collapse | Copy Code

```
$("btnClick").click(function(){
  $("#dvBox").animate({height:"100px"});
});
```

Q39. How to disable jQuery animation?

Ans: Using jQuery property "jQuery.fx.off", which when set to true, disables all the jQuery animation. When this is done, all animation methods will immediately set elements to their final state when called, rather than displaying an effect.

Q40. How do you stop the currently-running animation?

Ans: Using jQuery "`.stop()`" method.

Q41. What is the difference between .empty(), .remove() and .detach() methods in jQuery?

Ans: All these methods `.empty()`, `.remove()` and `.detach()` are used for removing elements from DOM but they all are different.

`.empty()`: This method removes all the child element of the matched element where remove() method removes set of matched elements from DOM.

`.remove()`: Use `.remove()` when you want to remove the element itself, as well as everything inside it. In addition to the elements themselves, all bound events and jQuery data associated with the elements are removed.

`.detach()`: This method is the same as `.remove()`, except that `.detach()` keeps all jQuery data associated with the removed elements. This method is useful when removed elements are to be reinserted into the DOM at a later time.

Find out more here

Q42. Explain .bind() vs .live() vs .delegate() vs .on()

Ans: All these 4 jQuery methods are used for attaching events to selectors or elements. But they all are different from each other.

`.bind()`: This is the easiest and quick method to bind events. But the issue with bind() is that it doesn't work for elements added dynamically that matches the same selector. `bind()` only attach events to the current elements not future element. Above that it also has performance issues when dealing with a large selection.

`.live()`: This method overcomes the disadvantage of `bind()`. It works for dynamically added elements or future elements. Because of its poor performance on large pages, this method is deprecated as of jQuery 1.7 and you should stop using it. Chaining is not properly supported using this method.

`.delegate()`: The .delegate() method behaves in a similar fashion to the .live() method, but instead of attaching the selector/event information to the document, you can choose where it is anchored and it also supports chaining.

`.on()`: Since live was deprecated with 1.7, so new method was introduced named ".on()". This method provides all the goodness of previous 3 methods and it brings uniformity for attaching event handlers.

Find out more here

Q43. What is wrong with this code line "$('#myid.3').text('blah blah!!!');"

Ans: The problem with above statement is that the selectors is having meta characters and to use any of the meta-characters (such as !"#$%&'()*+,./:;<=>?@[\]^`{|}~) as a literal part of a name, it must be escaped with with two backslashes: \\. For example, an element with id="foo.bar", can use the selector $("#foo\\.bar").
So the correct syntax is,

Collapse | Copy Code

```
$('#myid\\.3').text('blah blah!!!');
```

Q44. How to create clone of any object using jQuery?

Ans: jQuery provides `clone()` method which performs a deep copy of the set of matched elements, meaning that it copies the matched elements as well as all of their descendant elements and text nodes.

Collapse | Copy Code

```
$(document).ready(function(){
  $('#btnClone').click(function(){
     $('#dvText').clone().appendTo('body');
     return false;
  });
});
```

Q45. Does events are also copied when you clone any element in jQuery?

Ans: As explained in previous question, using `clone()` method, we can create clone of any element but the default implementation of the `clone()` method doesn't copy events unless you tell the `clone()` method to copy the events. The `clone()` method takes a parameter, if you pass true then it will copy the events as well.

Collapse | Copy Code

```
$(document).ready(function(){
   $("#btnClone").bind('click', function(){
     $('#dvClickme').clone(true).appendTo('body');
  });
```

Q46. What is difference between prop and attr?

Ans: `attr()`: Get the value of an attribute for the first element in the set of matched elements. Whereas, `.prop()`: (Introduced in jQuery 1.6) Get the value of a property for the first element in the set of matched elements.

Attributes carry additional information about an HTML element and come in name="value" pairs. Where Property is a representation of an attribute in the HTML DOM tree. once the browser parse your HTML code ,corresponding DOM node will be created which is an object thus having properties.

`attr()` gives you the value of element as it was defines in the html on page load. It is always recommended to use `prop()` to get values of elements which is modified via javascript/jquery , as it gives you the original value of an element's current state. Find out more here.

Q47. What is event.PreventDefault?

Ans: The `event.preventDefault()` method stops the default action of an element from happening. For example, Prevents a link from following the URL.

Q48. What is the difference between event.PreventDefault and event.stopPropagation?

Ans: `event.preventDefault()`: Stops the default action of an element from happening. `event.stopPropagation()`: Prevents the event from bubbling up the DOM tree, preventing any parent handlers from being notified of the event. For example, if there is a link with a click method attached inside of a DIV or FORM that also has a click method attached, it will prevent the DIV or FORM click method from firing.

Q49. What is the difference between event.PreventDefault and "return false"?

Ans: `e.preventDefault()` will prevent the default event from occurring, `e.stopPropagation()` will prevent the event from bubbling up and return false will do both.

Q50. What is the difference between event.stopPropagation and event.stopImmediatePropagation?

Ans: `event.stopPropagation()` allows other handlers on the same element to be executed, while`event.stopImmediatePropagation()` prevents every event from running. For example, see below jQuery code block.

Collapse | Copy Code

```
$("p").click(function(event){
  event.stopImmediatePropagation();
});
$("p").click(function(event){
  // This function won't be executed
  $(this).css("background-color", "#f00");
});
```

If `event.stopPropagation` was used in previous example, then the next click event on p element which changes the css will fire, but in case `event.stopImmediatePropagation()`, the next p click event will not fire.

Q51. How to check if number is numeric while using jQuery 1.7+?

Ans: Using "`isNumeric()`" function which was introduced with jQuery 1.7.

Q52. How to check data type of any variable in jQuery?

Ans: Using $.type(Object) which returns the built-in JavaScript type for the object.

Q53. How do you attach a event to element which should be executed only once?

Ans: Using jQuery `one()` method. This attaches a handler to an event for the element. The handler is executed at most once per element. In simple terms, the attached function will be called only once.

Collapse | Copy Code

```
$(document).ready(function() {
    $("#btnDummy").one("click", function() {
        alert("This will be displayed only once.");
    });
});
```

Q54. Can you include multiple version of jQuery? If yes, then how they are executed?

Ans: Yes. Multiple versions of jQuery can be included in same page.

Q55. In what situation you would use multiple version of jQuery and how would you include them?

Ans: Well, it is quite possible that the jQuery plugins which are used are dependent on older version but for your own jQuery code, you would like to use newer version. So because of this dependency, multiple version of jQuery may required sometimes on single page.

Below code shows how to include multiple version of jQuery.

Collapse | Copy Code

```
<script type='text/javascript' src='js/jquery_1.9.1.min.js'></script>

<script type='text/javascript'>
 var $jq = jQuery.noConflict();
</script>

<script type='text/javascript' src='js/jquery_1.7.2.min.js'></script>
```

By this way, for your own jQuery code use "$jq", instead of "$" as "$jq" refers to jQuery 1.9.1, where "$" refers to 1.7.2.

Q56. Is it possible to hold or delay document.ready execution for sometime?

Ans: Yes, its possible. With Release of jQuery 1.6, a new method "`jQuery.holdReady(hold)`" was introduced. This method allows to delay the execution of `document.ready()` event. `document.ready()` event is called as soon as your DOM is ready but sometimes there is a situation when you want to load additional JavaScript or some plugins which you have referenced.

Collapse | Copy Code

```
$.holdReady(true);
$.getScript("myplugin.js", function() {
     $.holdReady(false);
});
```

Q57. What is chaining in jQuery?

Ans: Chaining is one of the most powerful feature of jQuery. In jQuery, Chaining means to connect multiple functions, events on selectors. It makes your code short and easy to manage and it gives better performance. The chain starts from left to right. So left most will be called first and so on.

Collapse | Copy Code

```
$(document).ready(function(){
    $('#dvContent').addClass('dummy');
    $('#dvContent').css('color', 'red');
    $('#dvContent').fadeIn('slow');
});
```

The above jQuery code sample can be re-written using chaining. See below.

Collapse | Copy Code

```
$(document).ready(function(){
    $('#dvContent').addClass('dummy')
          .css('color', 'red')
          .fadeIn('slow');
});
```

Not only functions or methods, chaining also works with events in jQuery. Find out more here.

Q58. How does caching helps and how to use caching in jQuery?

Ans: Caching is an area which can give you awesome performance, if used properly and at the right place. While using jQuery, you should also think about caching. For example, if you are using any element in jQuery more than one time, then you must cache it. See below code.

Collapse | Copy Code

```
$("#myID").css("color", "red");
//Doing some other stuff......
$("#myID").text("Error occurred!");
```

Now in above jQuery code, the element with `#myID` is used twice but without caching. So both the times jQuery had to traverse through DOM and get the element. But if you have saved this in a variable then you just need to reference the variable. So the better way would be,

Collapse | Copy Code

```
var $myElement = $("#myID").css("color", "red");
//Doing some other stuff......
$myElement.text("Error occurred!");
```

So now in this case, jQuery won't need to traverse through the whole DOM tree when it is used second time. So in jQuery, Caching is like saving the jQuery selector in a variable. And using the variable reference when required instead of searching through DOM again.

Q59. You get "jquery is not defined" or "$ is not defined" error. What could be the reason?

Ans: There could be many reasons for this.

- You have forgot to include the reference of jQuery library and trying to access jQuery.
- You have include the reference of the jQuery file, but it is after your jQuery code.
- The order of the scripts is not correct. For example, if you are using any jQuery plugin and you have placed the reference of the plugin js before the jQuery library then you will mvc
- face this error.

Find out more here.

Q60. How to write browser specific code using jQuery?

Ans: Using jQuery.browser property, we can write browser specific code. This property contains flags for the useragent, read from navigator.userAgent. This property was removed in jQuery 1.9.

Q61. Can we use jQuery to make ajax request?

Ans: Yes. jQuery can be used for making ajax request.

Q62. What are various methods to make ajax request in jQuery?

Ans: Using below jQuery methods, you can make ajax calls.

- `load()` : Load a piece of html into a container DOM
- `$.getJSON()`: Load JSON with GET method.
- `$.getScript()`: Load a JavaScript file.
- `$.get()`: Use to make a GET call and play extensively with the response.
- `$.post()`: Use to make a POST call and don't want to load the response to some container DOM.
- `$.ajax()`: Use this to do something on XHR failures, or to specify ajax options (e.g. cache: true) on the fly.

Find out more here.

Q63. Is there any advantage of using $.ajax() for ajax call against $.get() or $.post()?

Ans: By using jQuery `post()` / jQuery `get()`, you always trust the response from the server and you believe it is going to be successful all the time. Well, it is certainly not a good idea to trust the response. As there can be n number of reason which may lead to failure of response.

Where `jQuery.ajax()` is jQuery's low-level AJAX implementation. `$.get` and `$.post` are higher-level abstractions that are often easier to understand and use, but don't offer as much functionality (such as error callbacks). Find out more here.

Q64. What are deferred and promise object in jQuery?

Ans: Deferred and promise are part of jQuery since version 1.5 and they help in handling asynchronous functions like Ajax. Find out more here.

Q65. Can we execute/run multiple Ajax request simultaneously in jQuery? If yes, then how?

Ans: Yes, it is possible to execute multiple Ajax request simultaneously or in parallel. Instead of waiting for first ajax request to complete and then issue the second request is time consuming. The better approach to speed up things would be to execute multiple ajax request simultaneously.

Using jQuery `.when()` method which provides a way to execute callback functions based on one or more objects, usually Deferred objects that represent asynchronous events. Find out more here.

Q66. Can you call C# code-behind method using jQuery? If yes,then how?

Ans: Yes. We can call C# code-behind function via $.ajax. But for do that it is compulsory to mark the method as WebMethod.

Q67. Which is the latest version of jQuery library?

Ans: The latest version (when this post is written) of jQuery is 1.10.2 or 2.0.3. jQuery 2.x has the same API as jQuery 1.x, but does not support Internet Explorer 6, 7, or 8.

Q68. Does jQuery 2.0 supports IE?

Ans: No. jQuery 2.0 has no support for IE 6, IE 7 and IE 8.

Q69. What are source maps in jQuery?

Ans: In case of jQuery, Source Map is nothing but mapping of minified version of jQuery against the un-minified version. Source map allows to debug minified version of jQuery library. Source map feature was release with jQuery 1.9. Find out more here.

Q70. How to use migrate jQuery plugin?

Ans: with release of 1.9 version of jQuery, many deprecated methods were discarded and they are no longer available. But there are many sites in production which are still using these deprecated features and it's not possible to replace them overnight. So jQuery team provided with jQuery Migrate plugin that makes code written prior to 1.9 work with it.

So to use old/deprecated features, all you need to do is to provide reference of jQuery Migrate Plugin. Find out more here.

Q71. Is it possible to get value of multiple CSS properties in single statement?

Ans: Well, before jQuery 1.9 release it was not possible but one of the new feature of jQuery 1.9 was `.css()` multi-property getter.

Collapse | Copy Code

```
var propCollection = $("#dvBox").css([ "width", "height", "backgroundColor" ]);
```

In this case, the `propCollection` will be an array and it will look something like this.

Collapse | Copy Code

```
{
  width: "100px",
  height: "200px",
  backgroundColor: "#FF00FF"
}
```

Q72. How do you stop the currently-running animation, remove all queued animations, and complete all animations for the matched elements?

Ans: It can be done via calling `.stop([clearQueue ] [, jumpToEnd ])` method and by passing both the parameters as true.

Q73. What is finish method in jQuery?

Ans: The .finish() method stops all queued animations and places the element(s) in their final state. This method was introduced in jQuery 1.9.

Q74. What is the difference between calling stop(true,true) and finish method?

Ans: The .finish() method is similar to .stop(true, true) in that it clears the queue and the current animation jumps to its end value. It differs, however, in that .finish() also causes the CSS property of all queued animations to jump to their end values, as well.

Q75. Consider a scenario where things can be done easily with javascript, would you still prefer jQuery?

Ans: No. If things can be done easily via CSS or JavaScript then You should not think about jQuery. Remember, jQuery library always comes with xx kilobyte size and there is no point of wasting bandwidth.

Q76. Can we use protocol less URL while referencing jQuery from CDNs?

Ans: Yes. Below code is completely valid.

Collapse | Copy Code

```
<script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
```

Q77. What is the advantage of using protocol less URL while referencing jQuery from CDNs?

Ans: It is quite useful when you are moving from HTTP to HTTPS url. You need to make sure that correct protocol is used for referencing jQuery library as pages served via SSL should contain no references to content served through unencrypted connections.

"protocol-less" URL is the best way to reference third party content that's available via both HTTP and HTTPS. When a URL's protocol is omitted, the browser uses the underlying document's protocol instead. Find out morehere.

Q78. What is jQuery plugin and what is the advantage of using plugin?

Ans: A plug-in is piece of code written in a standard JavaScript file. These files provide useful jQuery methods which can be used along with jQuery library methods. jQuery plugins are quite

useful as its piece of code which is already written by someone and re-usable, which saves your development time.

Q79. What is jQuery UI?

Ans: jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library that can be used to build interactive web applications.

Q80. What is the difference between jQuery and jQuery UI?

Ans: jQuery is the core library. jQueryUI is built on top of it. If you use jQueryUI, you must also include jQuery.

Note: If you have any questions to add to this list then please put it comments. We will be glad to add them in this list. We will be keep on updating this list with new questions and share the updates on our Facebook orTwitter channel. If you are not following us then request you to please follow and stay updated.

ADO.NET

Introduction

In this section, we will touch base on one of the important concepts in .NET database access. You can download my .NET interview questions PDF from http://www.questpond.com/SampleDotNetInterviewQuestionBook.zip. I have also put all the Design Patterns text in video format and uploaded it athttp://www.questpond.com/FreeDesign1.htm. You can visit http://www.questpond.com and download the complete architecture interview questions PDF which covers SOA, UML, Design Patterns, Togaf, OOPs etc.
You can download the software architecture interview questions PDF here: Download Software Architecture Interview Questions.

Previous parts of my interview questions series for architects:

- Part 1 - SoftArchInter1.aspx
- Part 2 - SoftArch2.aspx
- Part 3 - SoftArch3.aspx
- Part 4 - SoftArch4.aspx

UML interview questions Part 1: SoftArch5.aspx
Happy job hunting.....

(B) What is the namespace in which .NET has the data functionality class?

Following are the namespaces provided by .NET for data management:

System.Data

This contains the basic objects used for accessing and storing relational data, such as `DataSet`, `DataTable`, and `DataRelation`. Each of these is independent of the type of data source and the way we connect to it.

System.Data.OleDB

It contains the objects that we use to connect to a data source via an OLE-DB provider, such as `OleDbConnection`, `OleDbCommand`, etc. These objects inherit from the common base classes, and so have the same properties, methods, and events as the `SqlClient` equivalents.

System.Data.SqlClient

This contains the objects that we use to connect to a data source via the Tabular Data Stream (TDS) interface of Microsoft SQL Server (only). This can generally provide better performance as it removes some of the intermediate layers required by an OLE-DB connection.

System.XML

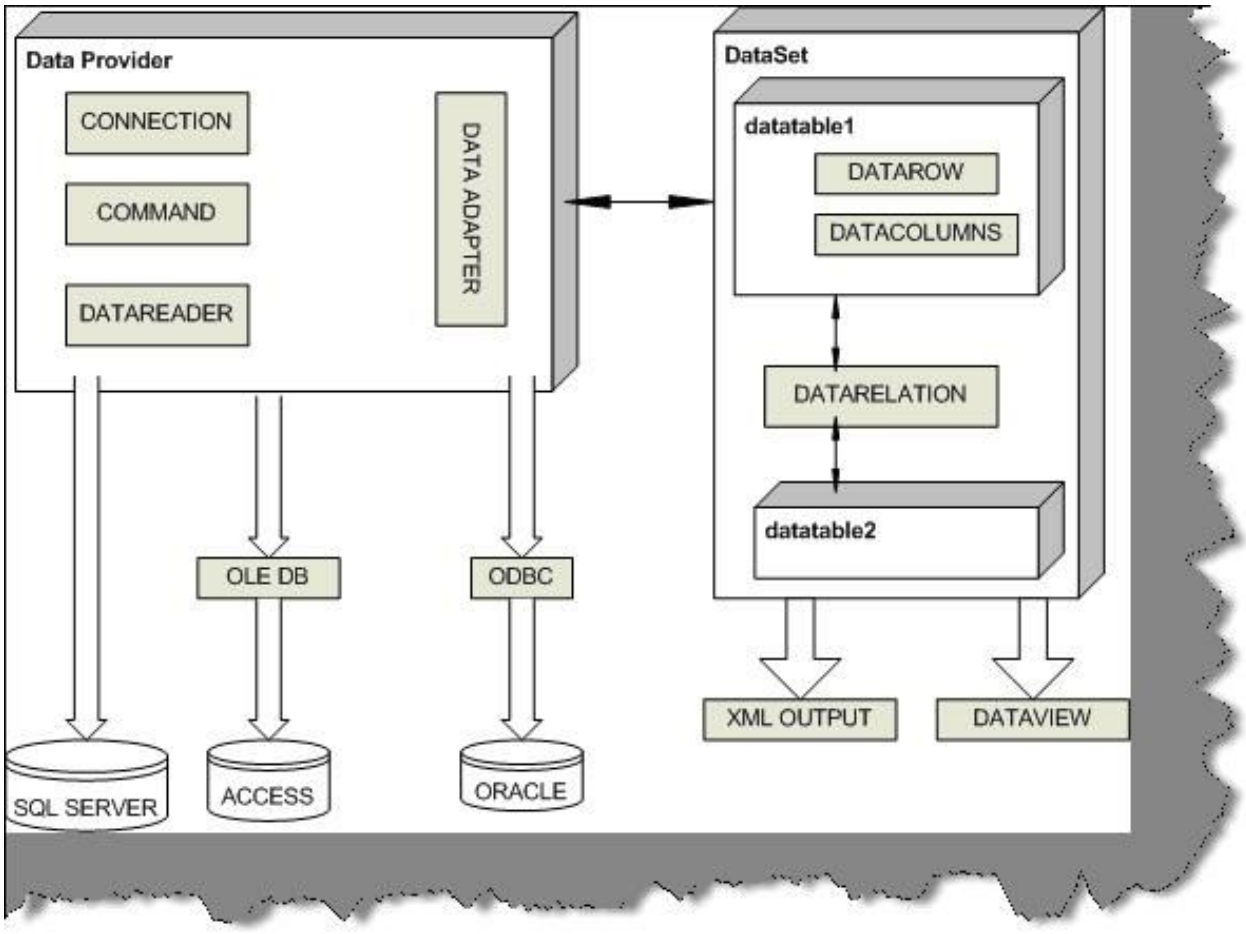
This contains the basic objects required to create, read, store, write, and manipulate XML documents according to W3C recommendations.

(B) Can you give an overview of the ADO.NET architecture?

The most important concept in ADO.NET architecture is “Data Provider”. Data Provider provides access to data sources (SQL Server, Access, Oracle). In short, it provides an object to achieve

functionalities like opening and closing connection, retrieve data, and update data. In the below figure, you can see the four main sections of a data provider:

- Connection.
- Command object (this is the object for using Stored Procedures).
- Data Adapter (this object acts as a bridge between the data store and the dataset).
- Data Reader (this object reads data from the data store in forward only mode).
- DataSet object represents disconnected and cached data. If you see the diagram, it is not in direct connection with the data store (SQL Server, Oracle, etc.) rather it talks with the data adapter, who is responsible for filling the dataset. The dataset can have one or more datatables and relations.



- ***Figure: 9.1 ADO.NET architecture***
- The DataView object is used to sort and filter data in the datatable.

Note: This is one of the poular questions in .NET interviews. Just paste the picture in your mind and during the interview try to refer to the image.

(B) What are the two fundamental objects in ADO.NET?

`DataReader` and `DataSet` are the two fundamental objects in ADO.NET.

(B) What is the difference between DataSet and DataReader?

Following are some major differences between `DataSet` and `DataReader`:

- `DataReader` provides forward-only and read-only access to data, while the `DataSet` object can hold more than one table (in other words, more than one row set) from the same data source as well as the relationships between them.
- `DataSet` is a disconnected architecture while `DataReader` is a connected architecture.
- `DataSet` can persist contents while `DataReader` cannot persist contents, they are forward only.

(I) What are the major difference between classic ADO and ADO.NET?

Following are some major differences:

- In ADO, we have a `Recordset` and in ADO.NET we have a `DataSet`.
- In `Recordset`, we can only have one table. If we want to accommodate more than one table, we need to do inner join and fill the `Recordset`. A `DataSet` can have multiple tables.
- All data is persisted in XML as compared to classic ADO where data is persisted in binary format.

(B) What is the use of the Connection object?

They are used to connect data to a `Command` object.

- An `OleDbConnection` object is used with an OLE-DB provider.

- A `SqlConnection` object uses Tabular Data Services (TDS) with MS SQL Server.

(B) What is the use of Command objects?

They are used to connect a `Connection` object to a DataReader or `DataSet`. Following are the methods provided by a `Command` object:

- `ExecuteNonQuery`
- Executes the command defined in the `CommandText` property against the connection defined in the`Connection` property for a query that does not return any row (an UPDATE, DELETE, or INSERT). Returns an Integer indicating the number of rows affected by the query.
- `ExecuteReader`
- Executes the command defined in the `CommandText` property against the connection defined in the `Connection` property. Returns a "reader" object that is connected to the resulting row set within the database, allowing the rows to be retrieved.
- `ExecuteScalar`
- Executes the command defined in the `CommandText` property against the connection defined in the `Connection` property. Returns only a single value (effectively the first column of the first row of the resulting row set, any other returned columns and rows are discarded). It is fast and efficient when only a "singleton" value is required.

(B) What is the use of a data adapter?

These objects connect one or more `Command` objects to a `DataSet` object. They provide logic that would get data from the data store and populates the tables in the `DataSet`, or pushes the changes in the `DataSet` back into the data store.

- An `OleDbDataAdapter` object is used with an OLE-DB provider
- A `SqlDataAdapter` object uses Tabular Data Services with MS SQL Server.

(B) What are basic methods of a DataAdapter?

These are the most commonly used methods of a `DataAdapter`:

- `Fill`
- Executes the Select command to fill the `DataSet` object with data from the data source. It an also be used to update (refresh) an existing table in a `DataSet` with changes made to the data in the original data source if there is a primary key in the table in the `DataSet`.

- `FillSchema`
- Uses the `SelectCommand` to extract just the schema for a table from the data source, and creates an empty table in the `DataSet` object with all the corresponding constraints.
- `Update`
- Calls the respective `InsertCommand`, `UpdateCommand`, or `DeleteCommand` for each inserted, **updated**, or deleted row in the `DataSet` so as to update the original data source with the changes made to the content of the `DataSet`. This is a little like the `UpdateBatch` method provided by the ADO `Recordset` object, but in the `DataSet`, it can be used to update more than one table.

(B) What is a DataSet object?

The `DataSet` provides the basis for disconnected storage and manipulation of relational data. We fill it from a data store, work with it while disconnected from that data store, then reconnect and flush changes back to the data store if required.

(B) What are the various objects in a DataSet?

`DataSet` has a collection of `DataTable` objects within the `Tables` collection. Each `DataTable` object contains a collection of `DataRow` objects and a collection of `DataColumn` objects. There are also collections for primary keys, constraints, and default values used in this table, which is called as constraint collection, and the parent and child relationships between the tables. Finally, there is a `DefaultView` object for each table. This is used to create a`DataView` object based on the table, so that the data can be searched, filtered, or otherwise manipulated while displaying the data.
Note: Look back again to the main diagram for the ADO.NET architecture for visualizing this answer in pictorial form.

(B) How can we connect to Microsoft Access, FoxPro, and Oracle etc.?

Microsoft provides the `System.Data.OleDb` namespace to communicate with databases like Access, Oracle, etc. In short, any OLE DB-compliant database can be connected using the `System.Data.OldDb` namespace.
Note: A small sample of OLEDB is provided in *WindowsAppOleDb* which uses *Nwind.mdb* in the *bin* directory to display data in a listbox.

```
Private Sub loadData()
    Dim strPath As String
    strPath = AppDomain.CurrentDomain.BaseDirectory
    Dim objOLEDBCon As New OleDbConnection("Provider=Microsoft.Jet." & _
        "OLEDB.4.0;Data Source =" & strPath & "Nwind.mdb")
    Dim objOLEDBCommand As OleDbCommand
    Dim objOLEDBReader As OleDbDataReader
    Try
        objOLEDBCommand = New OleDbCommand("Select FirstName from Employees")
        objOLEDBCon.Open()
        objOLEDBCommand.Connection = objOLEDBCon
        objOLEDBReader = objOLEDBCommand.ExecuteReader()
        Do While objOLEDBReader.Read()
            lstNorthwinds.Items.Add(objOLEDBReader.GetString(0))
        Loop
    Catch ex As Exception
    Throw ex
    Finally
        objOLEDBCon.Close()
    End Try
End Sub
```

The main code is in the `LoadData` method which actually loads the data in the list box.
Note: This source code has the connectionstring hard coded in the program itself which is not a good programming practice. For Windows applications, the best place to store the connectionstring is "*App.config*". Also note that the `AppDomain.CurrentDomain.BaseDirectory` function gives the current path of the running exe which is "*BIN*" and the MDB file is in that directory. Also note, the `Finally` block executes irrespective of if there is an error or not. This ensures that all the connections to the datastore are freed. It is a best practice to put all clean up statements in a `Finally` block thus ensuring that the resources are deallocated properly.

(B) How do we connect to SQL Server, which namespace do we use?

Below is the code required. We will try to understand the code in a more detailed manner. For this sample, we will also need a SQL table setup, which I have imported, using the DTS wizard.

Collapse | Copy Code

```
Private Sub LoadData()
    ' note :- with and end with makes your code more readable
    Dim strConnectionString As String
    Dim objConnection As New SqlConnection
    Dim objCommand As New SqlCommand
    Dim objReader As SqlDataReader
    Try
    ' this gets the connectionstring from the app.config file.
```

```
    ' note if this gives error see where the MDB file is stored
    ' in your pc and point to thastrConnectionString =
AppSettings.Item("ConnectionString")
    ' take the connectiostring and initialize the connection object
    With objConnection
        .ConnectionString = strConnectionString
        .Open()
    End With
    objCommand = New SqlCommand("Select FirstName from Employees")
    With objCommand
        .Connection = objConnection
        objReader = .ExecuteReader()
    End With
    ' looping through the reader to fill the list box
    Do While objReader.Read()
        lstData.Items.Add(objReader.Item("FirstName"))
    Loop
    Catch ex As Exception
        Throw ex
    Finally
    objConnection.Close()
End Try
```

Collapse | Copy Code

```
<appSettings>
    <add key="Connectionstring" value="Server=ERMBOM1-IT2;User
ID=sa;Database=Employees"/>
</appSettings>
```

Note: The above code is provided in CD in the folder *WindowsAppSqlClient*. Comments in the code explain a lot but we will iterate through the code later. `LoadData` is the main method which loads the data from SQL Server. Before running this code, you have to install SQL Server in your machine. As we are dealing with SQLClient, we need to setup a database in SQL Server. For this sample, I have imported *Nwind.mdb* in the "*SampleAccessDatabase*" folder in CD into SQL Server. Depending on the computer you will have to change the connectionstring in the *Web.config* file.

For setting up the sample SQL table, we can use the DTS import wizard to import the table. See the below figure which uses Microsoft Access as the data source. While importing the database, give the database name as "Employees".

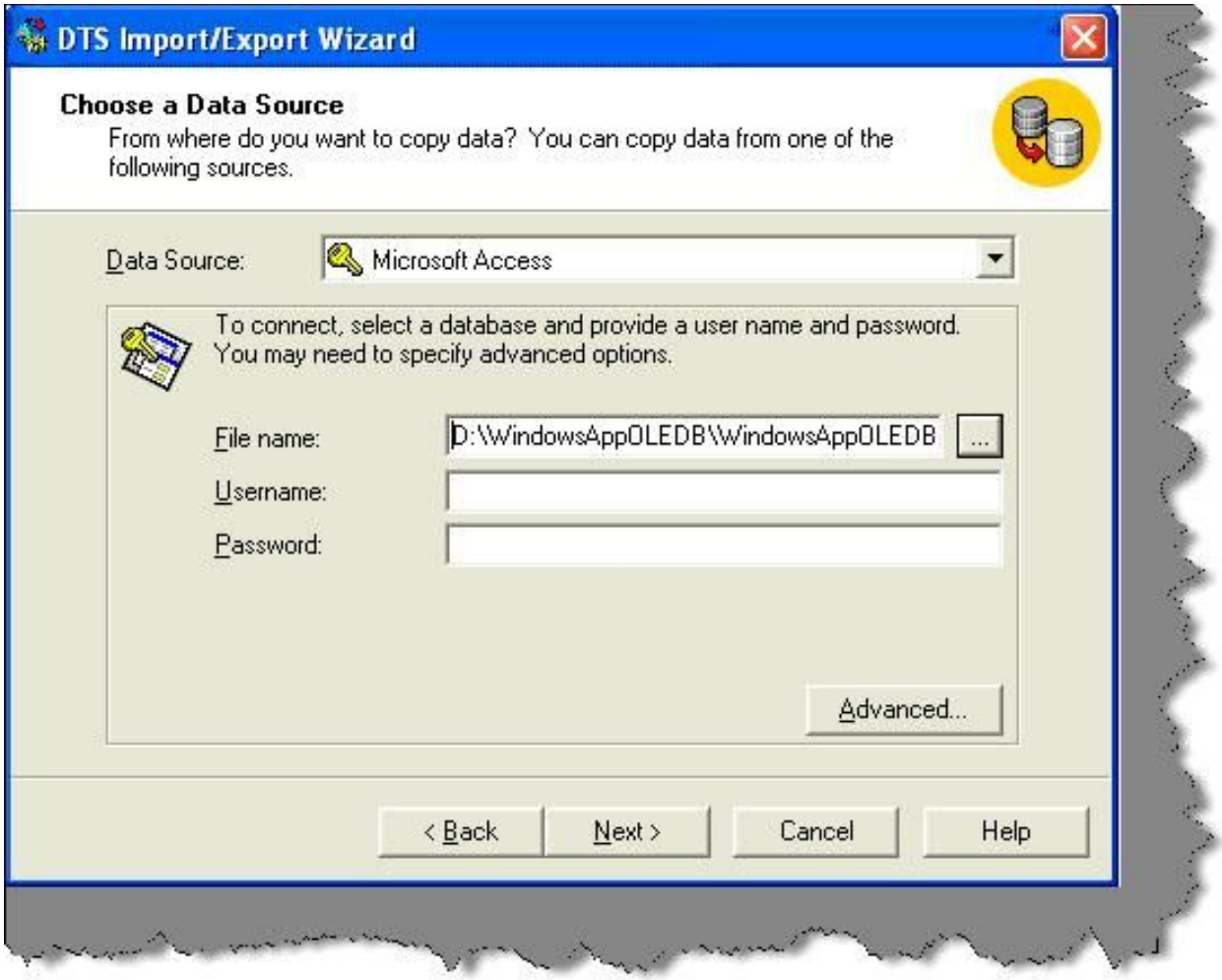


Figure: 9.2 Loading "Nwind.mdb" in SQL SERVER for the sample

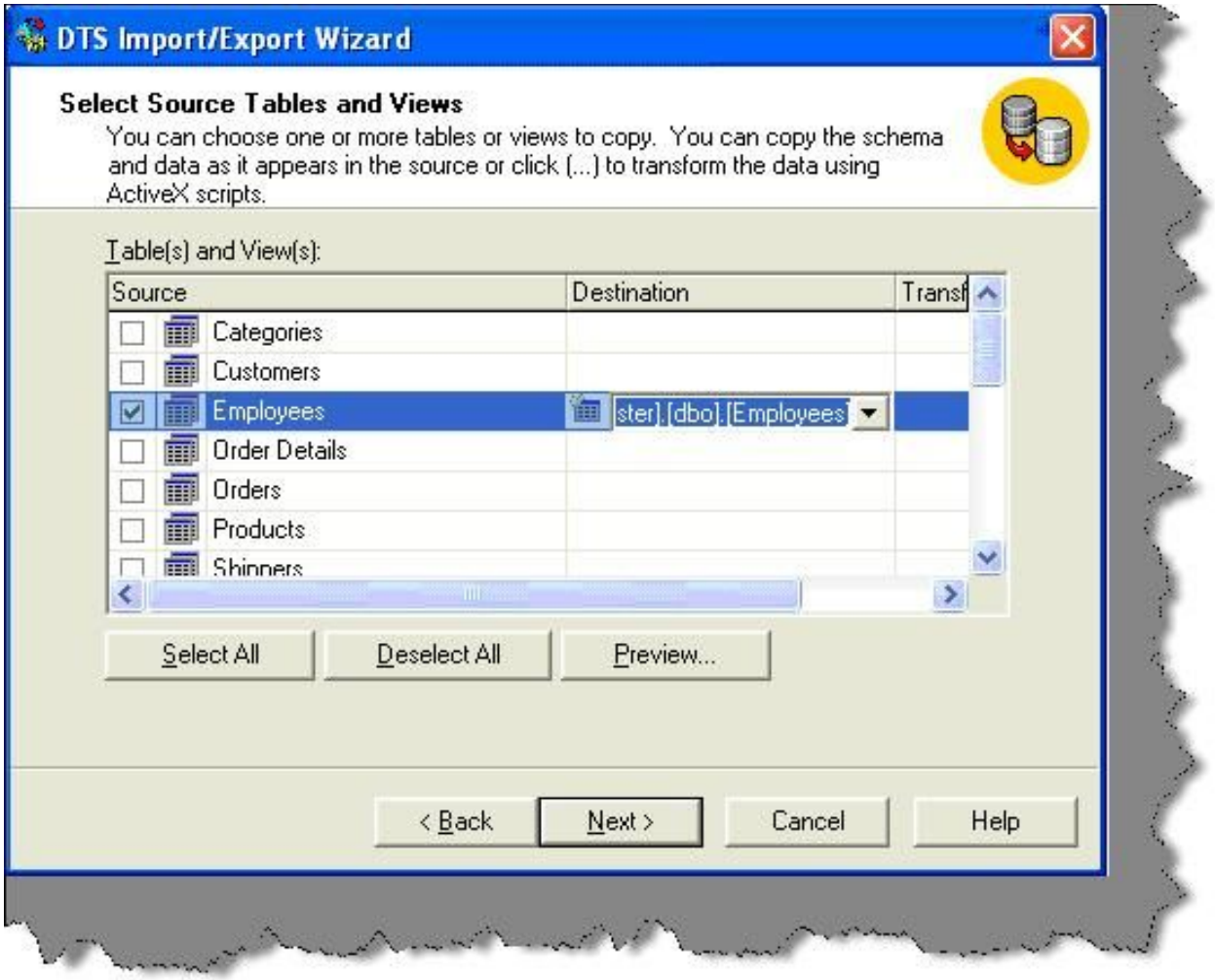


Figure: 9.3 Load only the Employee table.

To make it simple, we will only import the employee table as that is the only thing needed in our sample code.

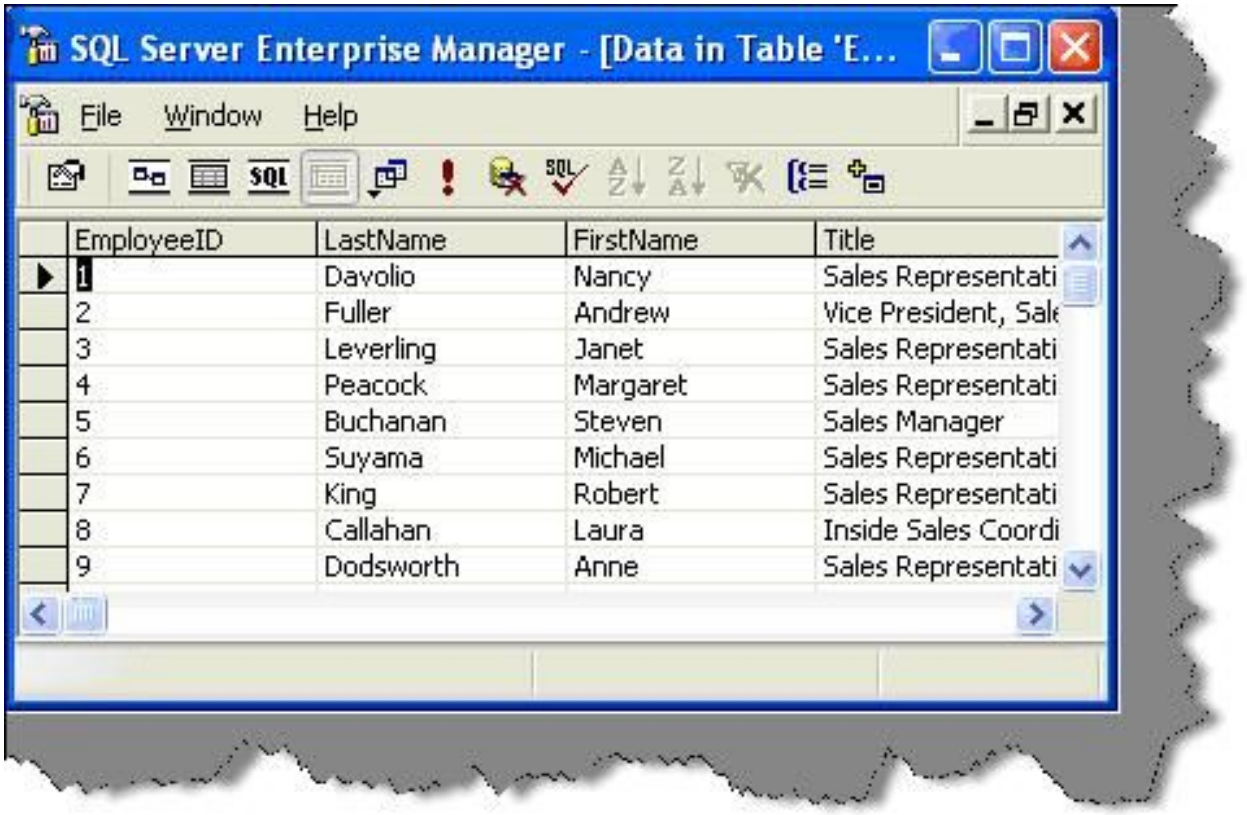


Figure: 9.4 View of loaded Employee table

Now from interview point of view, definitely you are not going to say the whole source code which is given in the book. The interviewer expects a broader answer of the steps needed to connect to SQL Server. You only have to explain the `LoadData` method in a broader way. Following are the steps to connect to SQL Server:

- First imports the namespace "`System.Data.SqlClient`".
- Create a connection object as shown in the `LoadData` method.
  - Collapse | Copy Code

```
With objConnection
    .Connection String = strConnectionString
    .Open ()
End With
```

- Create the command object with the SQL. Also, assign the created connection object to the command object and execute the reader.
  - Collapse | Copy Code

```
ObjCommand = New SqlCommand ("Select First Name from Employees")

With objCommand
    .Connection = objConnection
    Breeder = .Execute Reader ()
End With
```

- Finally loop through the reader and fill the list box. If VB programmers are expecting the move next command, it has been replaced by `Read()` which returns `True` if there is

any data to be read. If `Read()` returns `False` that means that it's the end of the data reader and there is no more data to be read.

- Collapse | Copy Code

```
Do while objReader.Read ()
    lstData.Items.Add (objReader.Item ("First Name"))
Loop
```

- Do not forget to close the connection object.
- **Note**: In `LoadData`, you will see that the connectionstring is stored in the *Web.config* file and is loaded using `AppSettings.Item("ConnectionString")`. While running this sample live on your database, do not forget to change this connectionstring according to your machine name and SQL Server or else the source code will not run.

Old Web forms Page Life Cycle

(1) PreInit The entry point of the page life cycle is the pre-initialization phase called "PreInit". This is the only event where programmatic access to master pages and themes is allowed. You can dynamically set the values of master pages and themes in this event. You can also dynamically create controls in this event.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
protected void Page_PreInit(object sender, EventArgs e)

{        //  Use this event for the following:          //  Check the IsPostBack property to determine whether this is the first time the page is being processed.       //  Create or re-create dynamic controls.        //  Set a master page dynamically.      //  Set the Theme property dynamically.
}
```

---

(2)Init This event fires after each control has been initialized, each control's UniqueID is set and any skin settings have been applied. You can use

this event to change initialization values for controls. The "Init" event is fired first for the most bottom control in the hierarchy, and then fired up the hierarchy until it is fired for the page itself.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected void Page_Init(object sender, EventArgs e)
{// Raised after all controls have been initialized and any skin settings have been applied. Use this event to read or initialize control properties.}
```

---

(3)InitComplete Raised once all initializations of the page and its controls have been completed. **Till now the viewstate values are not yet loaded, hence you can use this event to make changes to view state that you want to make sure are persisted after the next postback**

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected void Page_InitComplete(object sender, EventArgs e)
{       // Raised by the  Page object. Use this event for processing tasks that require all initialization be complete. }
```

---

(4)PreLoad Raised after the page loads view state for itself and all controls, and after it processes postback data that is included with the Request instance

(1)Loads ViewState : ViewState data are loaded to controls

Note : The page viewstate is managed by ASP.NET and is used to persist information over a page roundtrip to the server. Viewstate information is saved as a string of name/value pairs and contains information such as control text or value. The viewstate is held in the value property of a hidden <input> control that is passed from page request to page request.

(2)Loads Postback data : postback data are now handed to the page controls

Note : During this phase of the page creation, form data that was posted to the server (termed postback data in ASP.NET) is processed against each control that requires it. Hence, the page fires the LoadPostData event and parses through the page to find each control and updates the control state with the correct postback data. ASP.NET updates the correct control by matching the control's unique ID with the name/value pair in the NameValueCollection. This is one reason that ASP.NET requires unique IDs for each control on any given page.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected override void OnPreLoad(EventArgs e)
{        // Use this event if you need to perform processing on your page or control before the  Load event.        // Before the Page instance raises this event, it loads view state for itself and all controls, and then processes any postback data included with the Request instance.}
```

---

(5)Load The important thing to note about this event is the fact that by now, the page has been restored to its previous state in case of postbacks. Code inside the page load event typically checks for PostBack and then sets control properties appropriately. This method is typically used for most code, since this is the first place in the page lifecycle that all values are restored. Most code checks the value of IsPostBack to avoid unnecessarily resetting state. You may also wish to call Validate and check the value of IsValid in this method. You can also create dynamic controls in this method.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected void Page_Load(object sender, EventArgs e)
{        // The  Page calls the  OnLoad event method on the  Page, then recursively does the same for each child control, which does the same for each of its child controls until the page and all controls are loaded.        // Use the OnLoad event method to set properties in controls and establish database connections.}
```

---

(6)Control (PostBack) event(s)ASP.NET now calls any events on the page or its controls that caused the PostBack to occur. This might be a button's click event or a dropdown's selectedindexchange event, for example.These are the events, the code for which is written in your code-behind class(.cs file).

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected void Button1_Click(object sender, EventArgs e)
{        // This is just an example of control event.. Here it is button click event that caused the postback}
```

---

(7)LoadComplete This event signals the end of Load.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected void Page_LoadComplete(object sender, EventArgs e)
{        // Use this event for tasks that require that all other controls on the page be loaded.}
```

---

(8)PreRender Allows final changes to the page or its control. This event takes place after all regular PostBack events have taken place. This event takes place before saving ViewState, so any changes made here are saved.For example : After this event, you cannot change any property of a button or change any viewstate value. Because, after this event, SaveStateComplete and Render events are called.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected override void OnPreRender(EventArgs e)
```

```
{        // Each data bound control whose DataSourceID property is
set calls its DataBind method.        // The PreRender event occurs
for each control on the page. Use the event to make final changes to
the contents of the page or its controls.}
```

---

(9)SaveStateComplete Prior to this event the view state for the page and its controls is set. Any changes to the page's controls at this point or beyond are ignored.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected override void OnSaveStateComplete(EventArgs e)
{        // Before this event occurs,  ViewState has been saved for
the page and for all controls. Any changes to the page or controls at
this point will be ignored.        // Use this event perform tasks
that require view state to be saved, but that do not make any changes
to controls.}
```

---

(10)Render This is a method of the page object and its controls (and not an event). At this point, ASP.NET calls this method on each of the page's controls to get its output. The Render method generates the client-side HTML, Dynamic Hypertext Markup Language (DHTML), and script that are necessary to properly display a control at the browser.

Note: Right click on the web page displayed at client's browser and view the Page's Source. You will not find any aspx server control in the code. Because all aspx controls are converted to their respective HTML representation. Browser is capable of displaying HTML and client side scripts.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
// Render stage goes here. This is not an event
```

---

(11)UnLoad This event is used for cleanup code. After the page's HTML is rendered, the objects are disposed of. During this event, you should destroy any objects or references you have created in building the page. At this point, all processing has occurred and it is safe to dispose of any remaining objects, including the Page object. Cleanup can be performed on-

(a) Instances of classes i.e. objects
(b) Closing opened files
(c) Closing database connections.

EXAMPLE : Override the event as given below in your code-behind cs file of your aspx page

```
protected void Page_UnLoad(object sender, EventArgs e)
{        // This event occurs for each control and then for the page.
In controls, use this event to do final cleanup for specific
controls, such as closing control-specific database connections.
// During the unload stage, the page and its controls have been
rendered, so you cannot make further changes to the response stream.
```

```
//If you attempt to call a method such as the Response.Write method,
the page will throw an exception.     }
-----------------------------------------------------------------
 For More Reference : Follow the below links
(1)http://www.15seconds.com/issue/020102.htm(2)
http://msdn.microsoft.com/en-us/library/ms178472.aspx(3)http://www.devlifestyle
.net/blogs/articles/archive/2009/05/24/asp-net-internals-viewstate-and-page-lif
e-cycle.aspx Thank You...
```

WCF

Key things to WCF are the names spaces System.Runtime.Serialization and System.ServiceModel

[DataContract]
public class Invoice

[DataMember] // Used to be included in the message and if it is not annotated it is not included in the
//message
private DateTime InvoiceDate;

ASP.NET Web Services

Accessing Methods over HTTP using XML SOAP

HTTP - Transfer Protocol
XML - Format the data is stored
SOAP - Protocol to call the services
WSDL - Web services to make them discoverable

HTTP
- GET
- POST
- HEAD
- DELETE

HTTP - Coarse Scott Allen
XML - Dan Sullivans

WSDL - Web Service Definition Language

Adding the Attribute [WebMethod(Description = “Adds an amount to the total.”)]

In order to use Session you have to inherit from : WebService

async keyword and all web services have methods with async method versions

Async, await

How do you catch exceptoins? With FaultException

[ScriptService] attribute that allows AJAX to be used with the .aspx page

```
<asp:ScriptManager runat=”server”>
<Services>
<asp:scriptreference Path=”path to asmx” />
</Services
</asp:ScriptManager>
```

SQL Server

Function is a database object in Sql Server. Basically it is a set of sql statements that accepts only input parameters, perform actions and return the result. Function can return only single value or a table. We can’t use function to Insert, Update, Delete records in the database table(s). For more about stored procedure and function refer the article Difference between Stored Procedure and Function

Types of Function

- **System Defined Function**
- These functions are defined by Sql Server for different purpose. We have two types of system defined function in Sql Server
    - **Scalar Function**

- Scalar functions operates on a single value and returns a single value. Below is the list of some useful Sql Server Scalar functions.

    - System Scalar Function

- Scalar Function
- Description
- abs(-10.67)
- This returns absolute number of the given number means 10.67.
- rand(10)
- This will generate random number of 10 characters.
- round(17.56719,3)
- This will round off the given number to 3 places of decimal means 17.567
- upper('dotnet')
- This will returns upper case of given string means 'DOTNET'
- lower('DOTNET')
- This will returns lower case of given string means 'dotnet'
- ltrim(' dotnet')
- This will remove the spaces from left hand side of 'dotnet' string.
- convert(int, 15.56)
- This will convert the given float value to integer means 15.
- **Aggregate Function**

- Aggregate functions operates on a collection of values and returns a single value. Below is the list of some useful Sql Server Aggregate functions.
    - System Aggregate Function
- Aggregate Function
- Description
- max()
- This returns maximum value from a collection of values.
- min()
- This returns minimum value from a collection of values.
- avg()
- This returns average of all values in a collection.
- count()
- This returns no of counts from a collection of values.

User Defined Function

- These functions are created by user in system database or in user defined database. We three types of user defined functions.

Scalar Function

- User defined scalar function also returns single value as a result of actions perform by function. We return any datatype value from function.

```
--Create a table
CREATE TABLE Employee
(
EmpID int PRIMARY KEY,
FirstName varchar(50) NULL,
```

```
LastName varchar(50) NULL,

Salary int NULL,

Address varchar(100) NULL,

)

--Insert Data

Insert into Employee(EmpID,FirstName,LastName,Salary,Address)
Values(1,'Mohan','Chauahn',22000,'Delhi');

Insert into Employee(EmpID,FirstName,LastName,Salary,Address)
Values(2,'Asif','Khan',15000,'Delhi');

Insert into Employee(EmpID,FirstName,LastName,Salary,Address)
Values(3,'Bhuvnesh','Shakya',19000,'Noida');

Insert into Employee(EmpID,FirstName,LastName,Salary,Address)
Values(4,'Deepak','Kumar',19000,'Noida');

--See created table

Select * from Employee
```

- Results | Messages

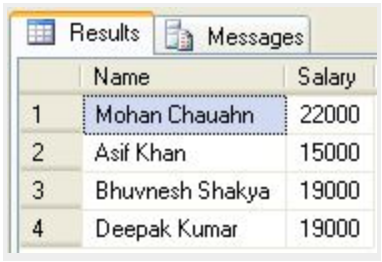
	EmpID	FirstName	LastName	Salary	Address
1	1	Mohan	Chauahn	22000	Delhi
2	2	Asif	Khan	15000	Delhi
3	3	Bhuvnesh	Shakya	19000	Noida
4	4	Deepak	Kumar	19000	Noida

```
--Create function to get emp full name

Create function fnGetEmpFullName
```

```
(
@FirstName varchar(50),
@LastName varchar(50)
)
returns varchar(101)
As
Begin return (Select @FirstName + ' '+ @LastName);
end
```

- 


```
--Calling the above created function
Select
dbo.fnGetEmpFullName(FirstName,LastName)
as Name, Salary from Employee
```

- 

	Name	Salary
1	Mohan Chauahn	22000
2	Asif Khan	15000
3	Bhuvnesh Shakya	19000
4	Deepak Kumar	19000

- # Inline Table-Valued Function
- User defined inline table-valued function returns a table variable as a result of actions perform by function. The value of table variable should be derived from a single SELECT statement.

```
--Create function to get employees
Create function fnGetEmployee()
returns Table
As
```

```
return (Select * from Employee)
```

- 


```
--Now call the above created function
Select * from fnGetEmployee()
```

- 

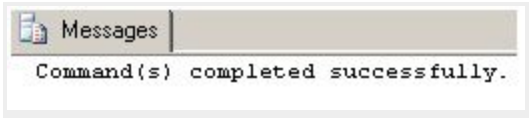
	EmpID	FirstName	LastName	Salary	Address
1	1	Mohan	Chauahn	22000	Delhi
2	2	Asif	Khan	15000	Delhi
3	3	Bhuvnesh	Shakya	19000	Noida
4	4	Deepak	Kumar	19000	Noida

- ## Multi-Statement Table-Valued Function

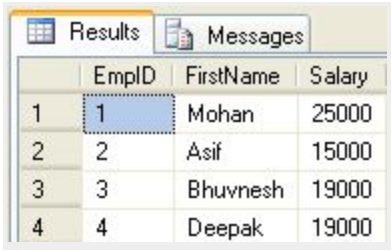
- User defined multi-statement table-valued function returns a table variable as a result of actions perform by function. In this a table variable must be explicitly declared and defined whose value can be derived from a multiple sql statements.

```
--Create function for EmpID,FirstName and
Salary of Employee
Create function fnGetMulEmployee()
returns @Emp Table
(
EmpID int,
FirstName varchar(50),
Salary int
)
As
begin
```

```
Insert into @Emp Select e.EmpID,e.FirstName,e.Salary from Employee e;

--Now update salary of first employee

update @Emp set Salary=25000 where EmpID=1;

--It will update only in @Emp table not in Original Employee table

return

end
```

- 


```
--Now call the above created function

Select * from fnGetMulEmployee()
```

- 

	EmpID	FirstName	Salary
1	1	Mohan	25000
2	2	Asif	15000
3	3	Bhuvnesh	19000
4	4	Deepak	19000

```
--Now see the original table. This is not affected by above function update command

Select * from Employee
```

- 

	EmpID	FirstName	LastName	Salary	Address
1	1	Mohan	Chauahn	22000	Delhi
2	2	Asif	Khan	15000	Delhi
3	3	Bhuvnesh	Shakya	19000	Noida
4	4	Deepak	Kumar	19000	Noida

Note

1. Unlike Stored Procedure, Function returns only single value.
2. Unlike Stored Procedure, Function accepts only input parameters.
3. Unlike Stored Procedure, Function is not used to Insert, Update, Delete data in database table(s).
4. Like Stored Procedure, Function can be nested up to 32 level.
5. User Defined Function can have upto 1023 input parameters while a Stored Procedure can have upto 2100 input parameters.
6. User Defined Function can't returns XML Data Type.
7. User Defined Function doesn't support Exception handling.
8. User Defined Function can call only Extended Stored Procedure.
9. User Defined Function doesn't support set options like set ROWCOUNT etc.

Summary

In this article I try to explain the types of function in sql server with example. I hope after reading this article you will be aware about function in Sql Server. I would like to have feedback from my blog readers. Please post your feedback, question, or comments about this article.

Entity Framework

1) What is Entity Framework?

Entity Framework is an Object Relational Mapping (ORM) from Microsoft that will enable the developers to work with domain specific objects, which eliminates the extra code being written in the data access layer.

2) Why to use Entity Framework?

Writing ADO.NET code and managing it is a tedious job. To avoid this, Microsoft has provided a solution - Entity Framework. Entity Framework reduces a great deal of code by enabling to work with relational data as domain specific objects.

3) What is the difference between Entity Framework and ADO.NET?

ADO.NET	Entity Framework
ADO.NET is faster.	"Entity Framework will be around the ADO.NET, which means ADO.NET is faster than Entity Framework."
We need to write so much code to talk to database.	Easy to use. As an Entity Framework will talk to database without much code involved.
Performance is better than Entity Framework.	Performance is not good compared to ADO.NET.

4) What is the difference between Entity Framework and LINQ to SQL?

LINQ to SQL	Entity Framework
Supports only for SQL database.	"Supports databases like SQL, MySQL, DB2 etc."
Allows only one-to-one mapping between the entity classes and the relational tables.	Allows one-to-one, one-to-many & many-to-many mappings between the Entity classes and the relational tables.
.dbml file is generated for maintaining the relationships.	Relationships are maintained in 3 different files - .csdl, .msl and .ssdl
No Support for Complex Types	Supports Complex Types

Supports Rapid application development with SQL Server.	Supports Rapid application development with SQL Server, DB2, MySQL etc.

5) What are the components of Entity Framework Architecture?

Below are the components of Entity Framework –

- Entity Data Model (EDM)
- LINQ to Entities
- Entity SQL
- Object Service
- Entity Client Data Provider
- ADO.Net Data Provider

6) What are the parts of Entity Data Model (EDM)?

Below are the parts of Entity Data Model –

- Conceptual Model
- Mapping
- Storage Model

7) How to create Entity Data Model (EDM)?

Add New Item -> ADO.NET Entity Data Model, which generates file with .edmx extension.

8) What does .edmx consists of?

.edmx file is a XML file and it has Conceptual Model, Storage Model and Mapping details i.e,

- SSDL (Store schema definition language)
- CSDL (Conceptual schema definition language)

- MSL (Mapping specification language)

9) What is Conceptual Model?

Conceptual Models are the model classes which contain the relationships. These are independent of the database design.

10) What is Storage Model?

Storage Models are our database design models, which contains database tables, views, stored procs and keys with relationships.

11) What is Mapping?

The Mapping will have the information on how the Conceptual Models are mapped to Storage Models.

12) What is LINQ to Entities?

LINQ to Entities is a query language which we used to write queries against the object models and the query result will return the entities defined in the Conceptual Model.

13) What is Entity SQL?

Entity SQL is a query language is like LINQ to Entities. This is a bit complex compared to LINQ to Entities. A developer who is using this should learn this separately.

14) What is the role of Entity Client Data Provider?

Responsibility of Entity Client Data Provider is to convert the LINQ to Entities or Entity SQL queries to a SQL query, which is understood by the underlying database. This finally communicates with ADO.NET Data Provider which in turn used to talk to the database.

15) What is the meaning of Pluralize and Singularize in Entity Framework?

Pluralize and Singularize gives the meaningful naming conventions for objects.

We will get this option while adding an edmx file. On selecting this option Entity Framework will adhere to Singular or Plural coding conventions.

16) What is DB Context

17) What is the role of DB Context?

DBContext will be responsible for Insert, Update and Delete functionalities of the entities. It acts like a bridge between the database and entity classes.

18) What is Object Context?

Object Context manages all the database operations, like database connection, and manages various entities of the Entity Model. DB Context is a wrapper around Object Context.

19) What is the difference between ObjectContext and DbContext?

Conceptually, both these are here for the same reason.

- **DBContext is wrapper around Object Context.**

- **Prior to EF 4.1, EDM used to use Object Context as base class for Context classes.**
- **There were some difficulties been observed in Object Context so now Db Context is introduced in EF 6.0 and this is used for all the development models –**
- **Database First, Model First and Code First.**
- **Object Context supports complied queries, but DB Context not.**
- **Object Context supports self-tracking entities, but DB Context does not.**
- **DB Context is thread safe, but Object Context not.**

20) What is Entity Set?

Entity Set holds the Entity Types and this most of the times compared with a database table.

21) What is Association Set?

Association Set is used to define the relationship between Entity Sets.

22) What is Entity Container?

It is a wrapper for Association Sets and Entity Sets and this is a critical to query a model.

23) What is Entity Graph?

Entity Graph is when one entity has a relation with other entity.

24) What is T4 Templates?

T4 Template (Text Template Transformation Toolkit) will generate the C# code based on edmx XML file. (.tt extension)

25) What are the types of Entities in Entity Framework?

- **POCO Entity**
- **Dynamic Proxy**

26) What is POCO Entity?

POCO Entity is Plain old CLR objects. These can be used as domain entities with our models.

27) What are the different approaches supported in the Entity Framework to create Entity Model?

Below are the different approaches supported in the Entity Framework to create Entity Model –

- **Database First**
- **Model First**
- **Code First**

28) What is the Database First Approach?

This approach is suitable, if we have a database already created and ready to use it. Using the existing database, we can create the Entity Models.

29) What is the Model First Approach?

This approach is suitable, when we prefer to create the Entity Models first and derive the database from the Entity Models.

30) What is the Code First Approach?

This approach is suitable, when we prefer to create the Domain classes first and derive the database from the Domain classes.

31) What are the advantages of Database First Approach?

Below are the advantages of Database First Approach –

- **Easy to create entity models if there is an existing database.**
- **Preferred approach for data intensive applications.**

32) What are the disadvantages of Database First Approach?

Below are the disadvantages of Database First Approach –

- **Once we create a edmx file from an existing database, huge pile of code is generated.**
- **If we want to add the additional functionality to the models generated, we need to extend the models.**

33) What are the advantages of Model First Approach?

Below are the advantages of Model First Approach –

- **Model first approach gives the flexibility to design the Entity Models independently and gives an option to improve at later stages.**
- **Model classes can be created by drawing it in the edmx designer, so no much of database is required.**

34) What are the advantages of Code First Approach?

Below are the advantages of Code First Approach –

- **Based on business objects we can decide the database structure.**
- **We can decide which classes need to be serialized and can specify the collection to eager load.**
- **Good for smaller applications.**

35) What are the disadvantages of Code First Approach?

Below are the disadvantages of Code First Approach –

- **All database related stuffs should be included in the code.**
- **For Stored Procs, we need to use the Fluent APIs to write it in code.**
- **Not good for data intensive applications.**

36) What are the Entity States supported in Entity Framework?

Below are the States supported in Entities during lifetime –

- **Added**
- **Deleted**
- **Modified**
- **Un Changed**
- **Detached**

mx file, it will have the list of entities and context class which will be derived from DB Context class.

---

37) What is Connected Scenario in Entity Framework?

Connected Scenario is when the Entity is retrieved from the database and some operations like modification is done in the same context.

38) What is Disconnected Scenario in Entity Framework?

Disconnected Scenario is when the Entity is retrieved from the database and some operations like modification is done in the different context. In this scenario context does not understand the changes done so we have to specify the changes done outside of context.

39) How we can increase the performance of Entity Framework?

Below are the points we can consider increasing the performance –

- Disable the Change Tracking if it's not required.
- Use the compiled query whenever required.
- Avoid using Views
- Fetch the required data from database.

40) What are the ways we can load the related entities in Entity Framework?

Below are the ways to load the entities in Entity Framework –

- Lazy Loading
- Eager Loading
- Explicit Loading

41) What is Lazy Loading in Entity Framework?

Lazy Loading is the default behavior of Entity Framework, wherein the dependent/related entities are loaded once they are accessed for the first time.

42) Give an example for Lazy Loading?

Consider a scenario of Student and Courses. Student can enroll to multiple Courses.

Filter the Student based on Student Id.

```
Student s = dbContext.Students.FirstOrDefault(a => a. StudentId == sId);
```

Load all the Courses related to Student.

```
List<Course> courses = s.Courses;
```

43) What is Eager Loading in Entity Framework?

Eager Loading will load the dependent/related entities at once. Unlike Lazy loading, Eager loading will do only one database call and get all the dependent entities.

44) Give an example for Eager Loading?

Consider a same Student scenario -

```
Student s = dbContext. Students.Include(s => s. Courses).FirstOrDefault(s => s. StudentId == sId);
```

45) What is Explicit Loading in Entity Framework?

Be default Entity Framework supports Lazy loading, but we can disable the Lazy loading and we can still load the dependent/related entities by calling “Load” method.

46) Give an example for Explicit Loading?

Continuing Student Scenario -

```
Student s = dbContext. Students.FirstOrDefault(a => a. StudentId == sId);
```

```
dbContext.Entry(s).Reference(s => s. Courses).Load();
```

47) Which type of loading is good in which scenario?

- Use Eager Loading, when the relations are not too much so that we can get the data in one database query.
- Use Eager Loading, if you are sure that we are going to use the dependent/related entities.
- If there are one-to-many relations use Lazy loading and if the dependent/related entities are not required down the line.
- When Lazy Loading is turned off, use Explicit loading when you are not sure whether dependent/related entities are not required down the line.

48) Can we use Stored Procedures in Entity Framework?

Yes.

49) Do Entity Framework handle the Change Tracking?

Yes.

50) How Entity Framework handles the Change Tracking?

Entity Framework supports automatic change tracking of all the loaded entities through context class.

51) What is the minimum requirement for Change Tracking in Entity Framework?

If Entity Framework need to handle Change Tracking, each entity should have Entity Key (Primary Key).