目录

[C# Interview 4](#_Toc430208363)

[!!!Can you store different types in an array in c#? 4](#_Toc430208364)

[What is jagged array? 4](#_Toc430208365)

[!!!What happens if finally block throws an exception 4](#_Toc430208366)

[!!!Difference between IS, AS and CASTING keyword 4](#_Toc430208367)

[!!!Array vs ArrayList vs List vs LinkedList 4](#_Toc430208368)

[Method overriding VS Method overloading vs overhiding 4](#_Toc430208369)

[Static Class 5](#_Toc430208370)

[Static vs Instance Method 5](#_Toc430208371)

[What is constructor and destructor? 5](#_Toc430208372)

[Can you use ‘this’ in static method? 5](#_Toc430208373)

[Static constructor 5](#_Toc430208374)

[What has access to a protected variable? 5](#_Toc430208375)

[When to use protected: 5](#_Toc430208376)

[Struct vs Class 5](#_Toc430208377)

[Value type vs Reference type 6](#_Toc430208378)

[1) Value types 6](#_Toc430208379)

[Abstract class: 6](#_Toc430208380)

[Interface: 6](#_Toc430208381)

[Abstract class VS Interface 6](#_Toc430208382)

[Why and where to use abstract class? 7](#_Toc430208383)

[Advantage of using interfaces? 7](#_Toc430208384)

[Can an abstract class have a constructor? 7](#_Toc430208385)

[What’s the use of a constructor in an abstract class 7](#_Toc430208386)

[Can you call an abstract method from an abstract class constructor 7](#_Toc430208387)

[Difference between ‘==’ and Equals? 7](#_Toc430208388)

[Difference between Convert.ToString and ToString 7](#_Toc430208389)

[Difference between String and StringBuilder 7](#_Toc430208390)

[Partial classes 7](#_Toc430208391)

[Partial Method 7](#_Toc430208392)

[Optional Parameter 7](#_Toc430208393)

[Dictionary in C# 8](#_Toc430208394)

[What is the difference between finally and  finalize() ? 8](#_Toc430208395)

[Difference between "throw" and "throw ex" in .NET? 8](#_Toc430208396)

[What is an exception filter? 8](#_Toc430208397)

[Name a reason to use an exception filter. 8](#_Toc430208398)

[What are code contract? 8](#_Toc430208399)

[What is an anonymous method? 9](#_Toc430208400)

[What is Func<T, TResult> in C#? 9](#_Toc430208401)

[What are Extension Methods? 9](#_Toc430208402)

[Which method is used to enforce garbage collection in .NET? 9](#_Toc430208403)

[What is the difference between dispose() and finalize ()? 9](#_Toc430208404)

[When use using keyword in c#? 9](#_Toc430208405)

[What is Garbage collection? 9](#_Toc430208406)

[How to fast page loading in asp.net? 9](#_Toc430208407)

[What is cross page posting in asp.net? 10](#_Toc430208408)

[Difference between Server.Transfer() vs Response.Redirect() 10](#_Toc430208409)

[What is test-driven development (TDD) ? 10](#_Toc430208410)

[What is the advantage of using TDD? 10](#_Toc430208411)

[What is a good TDD? 11](#_Toc430208412)

[What is Assert? 11](#_Toc430208413)

[(FluentMvc: sut.WithCallTo(x => x.Apply())).ShouldRenderRedirect… 11](#_Toc430208414)

[Difference between REST and SOAP 11](#_Toc430208415)

[Difference between var and int/string 11](#_Toc430208416)

[Why must one use JSON over XML? 11](#_Toc430208417)

[What things that Xml over Json 11](#_Toc430208418)

[What is Daily Scrum / Standup? 11](#_Toc430208419)

[What are 3 questions in Daily Scrum / Standup? 11](#_Toc430208420)

[How to call different Get Method In WebApi 12](#_Toc430208421)

[What is Aspect oriented programming(AOP)? 12](#_Toc430208422)

[What is OOP? 12](#_Toc430208423)

[What are the differences between OOP and AOP ? 12](#_Toc430208424)

[Dependency Injection vs Inversion of Control(IOC) 12](#_Toc430208425)

[SQL Interview 17](#_Toc430208426)

[Difference between SCOPE\_IDENTITY() AND @@IDENTITY 17](#_Toc430208427)

[Different between Primary key and Foreign key 17](#_Toc430208428)

[Different Join 17](#_Toc430208429)

[Different between Truncate and Delete 18](#_Toc430208430)

[What is an index? What are the types of indexes? How many clustered indexes can be created on a table? I create a separate index on each column of a table. what are the advantages and disadvantages of this approach? 18](#_Toc430208431)

[Unique & Non-Unique indexes 18](#_Toc430208432)

[When to use Index? 18](#_Toc430208433)

[Primary Key vs Unique Key 18](#_Toc430208434)

[Define candidate key, alternate key, composite key 19](#_Toc430208435)

[How does a transaction work? 19](#_Toc430208436)

[What is the difference between a HAVING CLAUSE and a WHERE CLAUSE? 19](#_Toc430208437)

[DML triggers 19](#_Toc430208438)

[Disadvantage of trigger 19](#_Toc430208439)

[What are cursors? Explain different types of cursors. What are the disadvantages of cursors? 19](#_Toc430208440)

[CTE 20](#_Toc430208441)

[ROW\_NUMBER() OVER(Partition By value,…) order by clause() 20](#_Toc430208442)

[PARTITION BY Clause 20](#_Toc430208443)

[Normalization 20](#_Toc430208444)

[De-normalization 20](#_Toc430208445)

[BCNF 20](#_Toc430208446)

[Deterministic and Nondeterministic Functions 20](#_Toc430208447)

[Javascript Interview 21](#_Toc430208448)

[What are closures and how are they used? 21](#_Toc430208449)

[Can you use x === “object” to test if x is an object? 21](#_Toc430208450)

[What happens when you don’t declare a variable in Javascript? 21](#_Toc430208451)

[What is the difference between == and === ? 21](#_Toc430208452)

[What datatypes are supported in Javascript? 21](#_Toc430208453)

[Javascript object adding 21](#_Toc430208454)

[How to convert ‘20’ to 20 in Javascript 21](#_Toc430208455)

[What is the difference between window.onload and the jQuery $(document).ready() method? 21](#_Toc430208456)

[What is NaN? 21](#_Toc430208457)

[Explain the meaning of the keyword ‘this’ in JavaScript functions 21](#_Toc430208458)

[How to check this.object? 22](#_Toc430208459)

[What is the difference between undefined value and null value? 22](#_Toc430208460)

[How to create a class in javascript? 22](#_Toc430208461)

[New keyword in javascript 23](#_Toc430208462)

[Javascript synchronous 23](#_Toc430208463)

[Write an simple function to determine string is palindrome 23](#_Toc430208464)

[What is the significance of including 'use strict' at the beginning of a JavaScript source file? 23](#_Toc430208465)

[How do you reduce the load time of the javascript (.js files) 24](#_Toc430208466)

[How to overload method in javascript? 24](#_Toc430208467)

[Explain Directives in Angular? 24](#_Toc430208468)

[Explain what is scope in Angular.js ? 24](#_Toc430208469)

[Explain what is services in Angular.js ? 24](#_Toc430208470)

[Explain what Angular JS routes does ? 25](#_Toc430208471)

[What happened when you open 25](#_Toc430208472)

C# Interview

!!!Can you store different types in an array in c#? Yes. Create an array of type object.

What is jagged array? A jagged array is an array of arrays.

!!!What happens if finally block throws an exception**?**

1.the execution propagates up, and should be handled at a higher level. If not, crashed

2. The finnaly block execution stops at the point where the exception is thrown.

!!!Difference between IS, AS and CASTING keyword**?**

The **[IS](https://msdn.microsoft.com/en-us/library/scekt9xw.aspx)** [operator](https://msdn.microsoft.com/en-us/library/scekt9xw.aspx) checks if an object can be cast to a specific type.

Example: if (someObject is StringBuilder) ...

The **[AS](https://msdn.microsoft.com/en-us/library/cscsdfbt.aspx)** [operator](https://msdn.microsoft.com/en-us/library/cscsdfbt.aspx) attempts to cast an object to a specific type, and returns null if it fails.

Example: StringBuilder b = someObject as StringBuilder; if (b != null) ...

|  |
| --- |
| **Casting**: The [cast operator](https://msdn.microsoft.com/en-us/library/ms173105.aspx) attempts to cast an object to a specific type, and throws an exeption if it fails.  Example: StringBuilder b = (StringBuilder)someObject. |

!!!Array vs ArrayList vs List vs LinkedList

Array: strong type with fixed length, particular type of int, String or Object, no type casting required. Can store primitives.

ArrayList: dynamic in increase or decrease the size. Can store all datatype value, cannot store premitives.

List<T> is basically backed by an array which is usually bigger than the current number of items. The elements are put in an array, and a new array is created when the old one runs out of space. This is fast for access by index, but slow at removing or inserting elements within the list or at the start.

LinkedList: is a doubly-linked list - each node knows its previous entry and its next one. This is fast for inserting after/before a particular node (or the head/tail), but slow at access by index. **LinkedList<T> will usually take more memory than List<T> because it needs space for all those next/previous references** - and the data will probably have less locality of reference, as each node is a separate object.

Method overriding VS Method overloading vs overhiding

**Overloading** means we will declare methods with same name but different signatures because of this we will perform different tasks with same method name. This overloading also called as compile time polymorphism or early binding.

**Overriding** means a base class reference variable pointing to a child class object, will invoke the overridden method in the Child class by using inheritance principle and using “polyIt’s also called as run time polymorphism or late binding or dynamic polymorphism.

**Overhiding** means a base class reference variable pointing to a child class object, will invoke the hidden method in the Base class.

Static Class:

1. They only contain static members.  
2. They cannot be instantiated.  
3. They are sealed.  
4. They cannot contain Instance Constructors or simply constructors as we know that they are associated with objects and operates on data when an object is created.

**In C#, static classes can be thought of as being both abstract and sealed**, **meaning that cannot be created or derived from.** Static classes (*and methods*) work especially well as "factories" as you can use them to create instances of other classes however there will only be a single static "factory" that will actually create the classes.

Static vs Instance Method**:** Static method is invoked using the class name, where as an instance method is invoked using an instance of the class.

What is constructor and destructor?

Constructors are special methods, used when instantiating a class. A constructor can never return anything, which is why you don't have to define a return type for it.

A destructor, a method called once an object is disposed, can be used to cleanup resources used by the object.It’s used for garbage collection for free the object you no longer use.

Can you use ‘this’ in static method?

No, the keyword 'this' returns a reference to the current instance of the class containing it. Static methods (or any static member) do not belong to a particular instance. They exist without creating an instance of the class.

Static constructor**:** Access modifiers(public, private) cannot be applied on Static constructor, it is always a public default constructor which is used to initialize static fields of the class. Using keyword static in front of constructor name, is called only once no matter how many instance you create and called before instance constructor.

What has access to a protected variable?

1. using derived class object

2. using this keyword

3. using base keyword

When to use protected:

Use it when you need to do some internal stuff that is not exposed in public API but still needs to be overridden by subclasses.

Your class is designed for inheritance - You expect the users of your library to inherit from the class that you are designing. Very often the class will be abstract.

Struct vs Class

1. A struct is a value type where as a class is a reference type.
2. Structs are stored on stack, where as classes are stored on the heap.
3. Structs can’t have destructors, but classes can have desctructors.
4. Struct can’t inherit from another class where class can.
5. Both structs and classes can inherit from an interface.
6. A class or struct cannot inherit from another struct.Struct are sealed types.
7. When you copy a struct into another struct, a new copy of that struct gets created and modifications on one struct will not affect the values contained by the other struct. When you copy a class into another class, we only get a copy of the reference variable. Both the reference variables point to the same object on the heap.

Value type vs Reference type

1. Value typeshold their value in memory where they are declared, but reference types hold a reference to an object in memory.
2. **Value types** are destroyed immediately after the scope is lost, whereas for **reference types** only the reference variable is destroyed after the scope is lost. The object is later destroyed by garbage collector.

Abstract class:

1. An abstract class can only be used as base class.
2. An abstract class can have implementation for some of its members except abstract method.
3. An abstract class cannot be sealed.
4. A non-abstract class derived from an abstract class must provide implementations for all inherited abstract members.(using override!)

Interface:

1. Just like classes interfaces also contains properties, methods, delegates or events, but only declarations and no implementations.
2. If a class or a struct inherits from an interface, it must provide implementation for all interface members.
3. Interface can inherit from other interfaces.
4. A class that inherits this interface must provide implementation for all interface members in the entire interface inheritance chain.
5. A class inherits from 2 or more interfaces and both the interfaces have the same method name.

Abstract class VS Interface

1. Abstract classes can have implementations for some of its members(not abstract Methods), but the interface can’t have implementation for any of its members.
2. Interfaces cannot have fields where as an abstract class can have fields.(int, string)
3. An interface can inherit from another interface only and cannot inherit from an abstract class, where as an abstract class can inherit from another abstract class or another interface.
4. Abstract class members can have access modifiers whereas interface members cannot have access modifiers.(no public, private for interface)

Why and where to use abstract class?

**Why:** Prevent us from accidently create a base class object.

**When:** we would create an abstract class, when want to move the common functionality of 2 or more related classes into a base class and when.

Advantage of using interfaces? Interfaces allow us to develop loosely coupled system, interfaces are very useful for Dependency Injection. Interfaces make unit testing and mocking easier.

Can an abstract class have a constructor? Yes, it’s used to initialize fields of the abstract class.

What’s the use of a constructor in an abstract class**?**

When an instance of derived class is created, the parent abstract class constructor is automatically called. Abstract classes can’t be directly instantiated, usually use protected access modifier with abstract class constructor, this prevents duplicate code.

Can you call an abstract method from an abstract class constructor**?** Yes; **Usage:** If you want the abstract method to be invoked automatically whenever an instance of the class that is derived from the abstract class is created, then we would call it in the constructor of abstract class.

Difference between ‘==’ and Equals?

“==” compares object references. Equals compare object contents, string datatypes always does content comparison.

Difference between Convert.ToString and ToString

Convert.ToString() handles null and return empty string, while ToString() doesn’t, and throws a NULL Reference exception.

Difference between String and StringBuilder

System.String is immutable(previous string remain in memory, cannot be changed, another new string object get created, and first string object value be carried over.), StringBuilder is Mutable, as StringBuilder objects are mutable, they offer better performance than string object of type System.String, when heavy string manipulation is involved.

Partial classes

Partial classes allow us to split a class into 2 or more files. All those parts are then combined into a single class, when the application is compiled. The partial keyword can also be used to split a struct or an interface over two or more files.

Partial Method

A partial class or a struct can contain partial methods.

A partial method is created using the partial keyword.

A partial method declaration consists of two parts.

**The definition**(only the method signature)

Optional Parameter

There are 4 ways that can be used to make method parameters optional

1. Use parameter arrays
2. Method overloading
3. Specify parameter defaults
4. Use Optional Attribute that is present in System.RunTime.InteropServices namespace.

Dictionary in C#

A Dictionary is a collection of (key, value) pairs.

When creating a dictionary, we need to specify the type for key and value.

Dictionary provides fast lookups for value using keys.

Keys in the dictionary must be unique.

1.foreach(KeyValuePair<int, Customer> customerKeyValuePair in dictionaryCustomers) 🡪 output customerKeyValuePair.Value.Attribute

2.foreach(int key in dictionaryCustomers.Keys) 🡪 output key

3.dictionaryCustomers.ContainsKey(104) 🡪 true of false

4.dictionaryCustomers.TryGetValue(102, out outputCustomer)

5.Convert a list to a Dictionary – use ToDictionary() method (Dictionary<int, Customer> dictionary = listCustomer.ToDictionary(x => x.Id);)

What is the difference between finally and  finalize() ?

**finally**:- It is a block associated with try catch to maintain cleanup code. Finally block will be executed always irrespective of whether exception is raised or not raised or whether the exception is handle or not handle.

**finalize()**:- It is a method, Garbage collector always calls this method just before destroying any object to perform cleanup activities.

Difference between "throw" and "throw ex" in .NET?

If you use "throw" statement, it preserve original error stack information.

If you use"throw ex" statement, stack trace of the exception will be replaced with a stack trace starting at the re-throw point.

What is an exception filter?

An exception filter will let you specify conditions along with a catch block which will only executed if condition satisfied.

Name a reason to use an exception filter.

Reason: because exceptions are preferable to catch and re-throw as they keep the stack structure, if the exception later causes the stack to dump, we would see the original source of it rather than the last place where it was re-thrown.

What are code contract?

Code contracts are used when you want to call and run a method under certain pre-conditions, post-conditions and running conditions (invariants).

public int Add(int num1, int num2)

{

Contract.Requires((num1 > 0) && (num2 > 0)); // precondition

Contract.Ensures(Contract.Result<int>() > 0); // post condition

return num1 + num2;

}

What is an anonymous method?  
In simple terms, anonymous method is a method without a name.

What is Func<T, TResult> in C#?  
In simple terms, Func<T, TResult> is just a generic delegate. For example, Func<Employee, string> is a delegate that represents a function expecting Employee object as an input parameter and returns a string.

What are Extension Methods?

Extension methods enable you to "add" methods to existing types without creating a new derived type, recompiling, or otherwise modifying the original type.

public static class StringHelper{

public static string ChangeFirstLetterCase(this string inputString)}

**Suppose you have a third party dll**, it doesn’t have the method you are looking for.

Create a static class with static method, this + parameter of this Extention Third party object.

Which method is used to enforce garbage collection in .NET?

System.GC.Collect ( ) method.

What is the difference between dispose() and finalize ()?

Although Dispose and Finalize both methods are used by CLR to perform garbage collection of runtime objects of .NET applications but there is a difference between them.

The Finalize method is called automatically by the runtime while the Dispose method is called by the programmer.

When use using keyword in c#?

When the SomeType class implements [IDisposable](http://msdn.microsoft.com/en-us/library/system.idisposable.aspx). File Stream, Email server, "using" statement is to ensure that the object is disposed as soon as it goes out of scope, and it doesn't require explicit code to ensure that this happens.

What is Garbage collection?

Garbage collection is used to prevent memory leaks during execution of programs.

How to fast page loading in asp.net?

Find out which is slow, is it the **application** or the **database** :

1. If the page is executing SQL queries or stored procedures, run those on the database and check how long they take to run. If the queries are taking most of the time, then you know you have to tune the queries for better performance. To tune the queries, there are several ways and I have listed some of them below.  
   a) Check if there are indexes to help the query  
   b) Select only the required columns, avoid **Select \*.**  
   c) Check if there is a possibility to reduce the number of joins  
   d) If possible use **NO LOCK** on your select statements  
   e) Check if there are cursors and if you can replace them with joins  
**2.** If the queries are running fast, then we know it is the **application code** that is causing the slowness. Isolate the page event that is causing the issue by turning tracing on. To turn tracing on, set **Trace="true"** in the page directive. Once you have tracing turned on you should see trace information at the bottom of the page as shown in the image below. In this case Page Load event is taking the maximum time. So we know, the code in **Page\_Load** event is causing the issue. Once you look at the code, you should be able to nail down the issue.

What is cross page posting in asp.net?

By default, button controls in ASP. NET pages post back to the same page that contains the button, where you can write an event handler for the post. In most cases this is the desired behavior, but occasionally you will also want to be able to post to another page in your application. The Server. Transfer method can be used to move between pages, however the URL does not change.

Difference between Server.Transfer() vs Response.Redirect()

Both Response.Redirect and Server.Transfer methods are used to transfer a user from one web page to another web page.

The **Response.Redirect** method redirects a request to a new URL and specifies the new URL while the **Server.Transfer** method for the current request, terminates execution of the current page and starts execution of a new page using the specified URL path of the page.

**Response.Redirect** sends an HTTP request to the browser, then the browser sends that request to the web server, then the web server delivers a response to the web browser. HTTP200->HTTP302(FOUND)🡪HTTP200

**Server.Tarnsfer** sends a request directly to the web server and the web server delivers the response to the browser.(HTTP200->HTTP200)

What is test-driven development (TDD) ?

TDD is a method of software development in which unit testing is repeatly done on source code. TDD is an iterative software development process where you first write the test with the idea that it must fail. This is a different approach to the traditional development where you write the application functionality first and then write test cases.

What is the advantage of using TDD?

1. Write new business code only if an automated unit test has failed: Business application requirements drive the tests and tests drive the actual functional code. Each test should test only one business concept. Any change in the business requirements will impact pre and post conditions of the test. Talking about pre and post conditions, following design by contract methodology helps achieving TDD. In design by contract, you specify the pre and post conditions that act as contracts of a method, which provides a specification to write your tests against.
2. Eliminate duplication from the code: A particular business concept should be implemented only once within the application code. Code for checking an account balance should be centralized to only one place within the application code. This makes your code decoupled, more maintainable and reusable.

What is a good TDD?

1. You should be able to run each test in isolation and in any order.
2. The test code should not have any duplicate business logic.
3. You should test for all the pre and post conditions as well as exceptions.
4. Each test should concentrate on one business requirement as mentioned earlier.

What is Assert?

(FluentMvc: sut.WithCallTo(x => x.Apply())).ShouldRenderRedirect…

**Asserts** tell the test runner whether a test has passed or failed.

Assert.That(Calculator.Add(1,2), Is.EqualTo(3))

First parameter is actual production code, second is expect value.

Older NUnit asserts?

Assert.AreEqual(3, Calculator.Add(1,2));

Difference between REST and SOAP

1) REST is more simple and easy to use than SOAP

2) REST uses HTTP protocol for producing or consuming web services while SOAP uses XML.

3) REST is lightweight as compared to SOAP and preferred choice in mobile devices and PDA's.

4) REST supports different format like text, JSON and XML while SOAP only support XML.

5) REST web services call can be cached to improve performance.

Difference between var and int/string

Var is an implicit type. It aliases any type. The aliased type is determined by the C# compiler. This has no performance penalty.

Why must one use JSON over XML?

It is faster and lighter than XML as on the wire data format

XML data is typeless while JSON objects are typed

JSON types: Number, Array, Boolean, String

XML data are all string

Data is readily available as JSON object is in your JavaScript

Fetching values is as simple as reading from an object property in your JavaScript code

What things that Xml over Json

XML is good for situations in which...

You need message validation

You're using XSLT

Your messages include a lot of marked-up text

You need to interoperate with environments that don't support JSON

You need attributes or namespacing

[What is Daily Scrum / Standup?](http://www.questions-interviews.com/software-development-testing-models/scrum.aspx#what_is_Daily_Scrum_Standup)  
Each day during the sprint, a project status meeting occurs. This is called a daily scrum, or the daily standup.

[What are 3 questions in Daily Scrum / Standup?](http://www.questions-interviews.com/software-development-testing-models/scrum.aspx#what_are_3_questions_in_Daily_Scrum_Standup)  
The three questions in Daily Scrum / Standup are:

What have you done since yesterday?

What are you planning to do today?

Any impediments/stumbling blocks?

How to call different Get Method In WebApi

By default the route configuration follows RESTFul conventions meaning that it will accept only the Get, Post, Put and Delete action names (look at the route in global.asax => by default it doesn't allow you to specify any action name => it uses the HTTP verb to dispatch

If you want to have different action names than the standard ones you could modify your route definition in global.asax:

Routes.MapHttpRoute(

name: "DefaultApi",

routeTemplate: "api/{controller}/{action}/{id}",

defaults: new { action = "get", id = RouteParameter.Optional }

);

What is Aspect oriented programming(AOP)?

Aspect-Oriented Programming (AOP) complements OOP (Object Oriented Programming) by allowing the developer to dynamically modify the static OO model to create a system that can grow to meet new requirements. AOP allows you to dynamically modify your static model consisting mainly of business logic to include the code required to fulfill the secondary requirements

What is OOP?

OOP is a technique to develop logical modules, such as classes that contain properties, methods, fields, and events. An object is created in the program to represent a class. Therefore, an object encapsulates all the features, such as data and behavior that are associated to a class.

What are the differences between OOP and AOP ?

**Object Oriented Programming (OOP)**

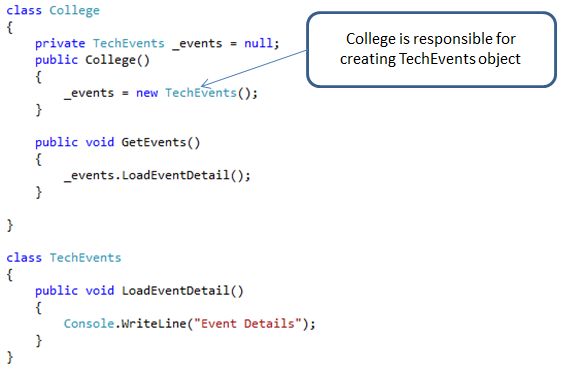
1. OOP looks at an application as a set of collaborating objects. OOP code scatters system level code like logging, security etc with the business logic code.
2. OOP nomenclature has classes, objects, interfaces etc.
3. Provides benefits such as code reuse, flexibility, improved maintainability, modular architecture, reduced development time etc with the help of polymorphism, inheritance and encapsulation.

**Aspect Oriented Programming (AOP)**

1. AOP looks at the complex software system as combined implementation of multiple concerns like business logic, data persistence, logging, security, multithread safety, error handling, and so on. Separates business logic code from the system level code. In fact one concern remains unaware of other concerns.
2. AOP nomenclature has join points, point cuts, advice, and aspects.
3. AOP implementation coexists with the OOP by choosing OOP as the base language. For example: AspectJ uses Java as the base language.
4. AOP provides benefits provided by OOP plus some additional benefits which are discussed in the next question.

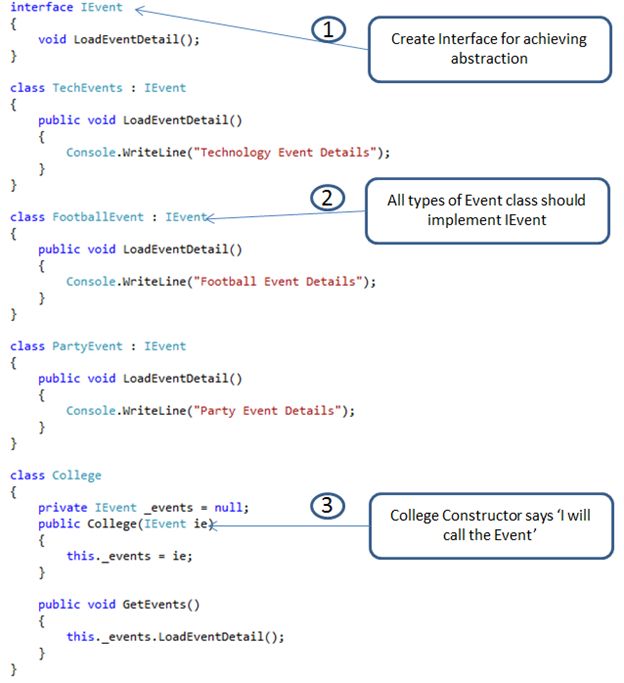
Dependency Injection vs Inversion of Control(IOC)

**Inversion of Control:**

**MVC has api like Ninject/StructureMap**  
  
Suppose I have a class say College and another class say TechEvents. As you can see in the preceding figure there are many problems that may arise:

1. **Both classes are tightly coupled with each other. I cannot have a College without TechEvents because a TechEvents object is created in a College Constructor.**
2. **If I make any changes to TechEvents, I need to compile or you can say update College too.**
3. **College controls the creation of Events. College is aware of the single event that is organized. If there is any specific event to be organized like Weekend FootballEvent or PartyEvent, then there needs to make changes to a College class as College is directly referring to Events.**

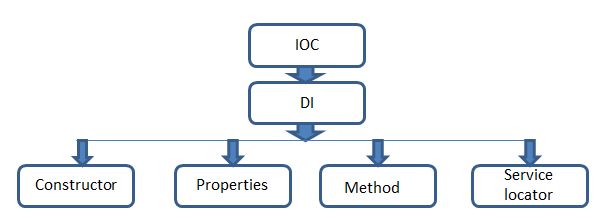
The solution to this could be to shift the control of events organization to some other place. This we call Inversion of Control (IOC), **inverting the control to some other entity instead of organizing the event in College directly itself.**

In other words, **the Main class should not have a concrete implementation of an aggregated class, rather it should depend on abstraction of that class**. The College class should depend on TechEvents class **abstraction using an interface or abstract class**.  


**Dependency Injection**

**IOC can be done using Dependency Injection (DI). It explains how to inject the concrete implementation into a class that is using abstraction, in other words an interface inside.** The main idea of dependency injection is to reduce the coupling between classes and move the binding of abstraction and concrete implementation out of the dependent class.

In Simple words, **DI is how one object know about other dependent object which is abstracted.**

There are mainly 4 ways of achieving the Dependency Injection.  


**1. Injection via Constructor**  
This methodology is already discussed above, where the object of the concrete class is passed to the constructor of the dependent class.

class College

{

        private IEvent \_events;

        public College(IEvent ie)

        {

            \_events = ie;

        }

       public void GetEvents()

        {

            this.\_events.LoadEventDetail();

        }

}

As you can see above, the event object is injected by the constructor keeping it loosely coupled. The College class will do his work and if it wants to get the events related details, it will call it in the constructor based on which event he wants to call.

College coll = new College(new FootballEvent());

**Along with this advantage, another advantage is that if there are any changes in events or added some more events then College doesn't need to care about that.**

**2. Injection via Property**  
This is the most commonly used methodology **where we inject the concrete class by creating a property whose type is of Interface.**

class College

{

        private IEvent \_events;

        public IEvent MyEvent

        {

            set

            {

                \_events = value;

            }

        }

}

As you can see above, **the setter of the MyEvent property will take a concrete object and bind it to the interface.** My class is loosely coupled from a concrete object. Now any changes to the any type of Event class will not affect my College class.

College coll = new College();

coll.MyEvent = new FootballEvent();

**3. Injection via Method**  
In this methodology, **the concrete class object is passed through the method parameter to the dependent class.**

class College

{

        private IEvent \_events;

        public void GetEvent(IEvent myevent)

        {

            this.\_events = myevent;

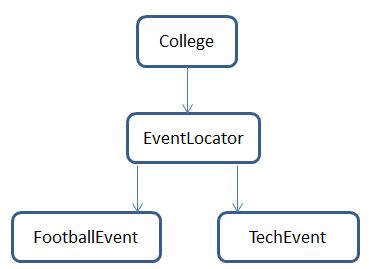
        }

}

As you can see above, **I have called the event of college using the GetEvents() method where the type of event is passed as a parameter that is of abstract type. This will help me to add or make changes to events without affecting the College, in other words both are decoupled.** This is how I can call the method.

College coll = new College();

coll.GetEvent(new FootballEvent());

**4. Injection via Service Locator**  
A service locator can act like a simple runtime mapper. **This allows code to be added at run-time without re-compiling the application and in some cases without having to even restart it.**  
  


class College

    {

        private IEvent \_events = null;

        EventLocator el = new EventLocator();

        public College(int index)

        {

            this.\_events = el.LocateEvent(index);

        }

    }

    class EventLocator

    {

        public IEvent LocateEvent(int index)

        {

            if (index == 1)

                return new FootballEvent();

            else if (index == 2)

                return new PartyEvent();

            else

                return new TechEvents();

        }

   }

In the code snippet above, y**ou can see that there is an EventLocator class between the Events and College that helps us to locate the service without knowing the concrete type.** I am just passing the index value in the constructor that in turn calls the third party to locate the event and return it to the constructor. Hence any changes to EventLocator will not affect the College class.

College coll = new College(1);

coll.GetEvents();

**Advantages of implementing this principle**

It helps in class decoupling.

Due to decoupling, the reusability of the code is increased.

Improved code maintainability and testing.

**Conclusion**

Inversion of control (IOC) talks about who is going to initiate the call

Dependency Injection (DI) talks about how one object acquires dependency on other object through abstraction.

SQL Interview

Difference between SCOPE\_IDENTITY() AND @@IDENTITY:

SCOPE\_IDENTITY() – Same session and the same scope

@@IDENTITY – Same session and across any scope

IDENT\_CURRENT(‘TableName’) – Specific table across any session and any scope.

Different between Primary key and Foreign key

**Primary Key**

Primary key uniquely identify a record in the table.

Primary Key can't accept null values.

By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index.

We can have only one Primary key in a table.

**Foreign Key**

Foreign key is a field in the table that is primary key in another table.

Foreign key can accept multiple null value.

Foreign key do not automatically create an index, clustered or non-clustered. You can manually create an index on foreign key.

We can have more than one foreign key in a table.

Different Join

**INNER JOIN**: Only matching rows between join table

**LEFT JOIN**: Only matching rows plus non-matching rows of left table

**FULL JOIN**: All rows

**SELF JOIN:** can be replace with LEFT JOIN INNER JOIN CROSS JOIN  
CROSS JOIN: A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table.

Different between Truncate and Delete

TRUNCATE doesn't generate any rollback data, which makes it lightning fast. It just deallocates the data pages used by the table.

However, if you are in a transaction and want the ability to "undo" this delete, you need to use DELETE FROM, which gives the ability to rollback.

In SQL Server, it is possible to rollback a truncate operation if you are inside a transaction and the transaction has not been committed.

What is an index? What are the types of indexes? How many clustered indexes can be created on a table? I create a separate index on each column of a table. what are the advantages and disadvantages of this approach?

Indexes in SQL Server are similar to the indexes in books. They help SQL Server retrieve the data quicker.

**Indexes are of two types**. Clustered indexes and non-clustered indexes.

When you create a clustered index on a table, all the rows in the table are stored in the order of the clustered index key. So, there can be only one clustered index per table.

Non-clustered indexes have their own storage separate from the table data storage. Non-clustered indexes are stored as B-tree structures (so do clustered indexes), with the leaf level nodes having the index key and it's row locater.

If you create an index on each column of a table, **it improves the query performance,** as the query optimizer can choose from all the existing indexes to come up with an efficient execution plan. **At the same t ime, data modification operations (such as INSERT, UPDATE, DELETE) will become slow**, as every time data changes in the table, all the indexes need to be updated. **Another disadvantage is that, indexes need disk space, the more indexes you have, more disk space is used.** Additional disk space. Clustered Index does not.

Unique & Non-Unique indexes

**Unique index** is used to enforce uniqueness of key values in the index.

**Note**: by default, **Primary Key constraint, creates a unique clustered index.**

Uniqueness is a property of an index, and **both CLUSTERED and NON-CLUSTERED indexes can be UNIQUE**.

When to use Index?

a. When there is large amount of data. For faster search mechanism indexes are appropriate.

b. To improve performance they must be created on fields used in table joins

Primary Key vs Unique Key

**PK**: Can be only one in a table, never allows null values. PK is a unique key identifier and cannot be null and must be unique.

**UK**: can be more than one unique key in one table; can have null value(only single null), can be a candidate key(FK)

Define candidate key, alternate key, composite key.

A **candidate** key is one that can identify each row of a table uniquely. Generally a candidate key becomes the primary key of the table.

**If the table has more than one candidate key, one of them will become the primary key, and the rest are called alternate keys**.

A key formed by combining at least two or more columns is called **composite** **key**.

How does a transaction work?

The **transaction** will group SQL commands into a single unit, and completes successfully only if all commands within it complete successfully. The whole thing fails if one command fails.

**e.g. with a store procedure, we have**

BEGIN TRANSACTION

/\*\*\* COMMANDS \*\*\*/

IF @@error <> 0

BEGIN

ROLLBACK TRANSACTION

END

COMMIT TRANSACTION

What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?

Having Clause is basically used only with the GROUP BY function in a query. WHERE Clause is applied to each row before they are part of the GROUP BY function in a query.

DML triggers

triggers are sets of commands that get executed when an event(Before Insert, After Insert, On Update, On delete of a row) occurs on a table, views.

Disadvantage of trigger

Triggers execute invisible to client-application application. They are not visible or can be traced in debugging code.

It is hard to follow their logic as it they can be fired before or after the database insert/update happens.

It is easy to forget about triggers and if there is no documentation it will be difficult to figure out for new developers for their existence.

Triggers run every time when the database fields are updated and it is overhead on system. It makes system run slower.

What are cursors? Explain different types of cursors. What are the disadvantages of cursors?

**Cursors** allow row-by-row prcessing of the resultsets.

**Types of cursors**: Static, Dynamic, Forward-only, Keyset-driven. See books online for more information.

**Disadvantages of cursors**: Each time you fetch a row from the cursor, it results in a network roundtrip, where as a normal SELECT query makes only one rowundtrip, however large the resultset is. Cursors are also costly because they require more resources and temporary storage.

Because a cursor is an actual object inside the database engine, there is a little overhead involved in creating the cursor and destroying it. Also, a majority of cursor operations occur in tempdb, so a heavily used tempdb will be even more overloaded with the use of cursors.

CTE

CTE is a temporary result set, that can be referenced within a SELECT, INSERT, UPDATE or DELETE statement that immediately follows the CTE.

ROW\_NUMBER() OVER(Partition By value,…) order by clause()

**Over** – specify the order of the rows.

**Order by** – Provide sort order for the records.

PARTITION BY Clause

When you specify a column or set of columns with PARTITION BY clause then it will divide the result set into record partitions and then finally ranking functions are applied to each record partition separately and the rank will restart from 1 for each record partition separately.

Normalization

Is the process of organizing data to minimize data redundancy(data duplication), which in trun ensures data consistency.

**1NF:**

Data in each column should be atomic. No multiple values, separated by comma.

The table does not contain any repeating column groups.

Identify each record uniquely using primary key.

**2NF**

The table meet all condition of 1NF

Move redundant data to a separate table

Create relationship between the tables using foreign keys.

**3NF**

Meet condition of 2NF

Does not contain columns that are not fully dependent upon the primary key.

De-normalization

Is the process of attempting to optimize the performance of a database by adding redundant data.

BCNF**:** if there are non-trivial dependencies between candidate key attributes.

Deterministic and Nondeterministic Functions

Deterministic functions always return the same result any time they are called with a specific set of input values and given the same state of the database.

Example: Square(), Power(), Sum(), AVG() and Count()

Note: All aggregate functions are deterministic functions.

Nondeterministic functions: may return different results each time they are called with a specific set of input values even if the database state they access remains the same.

Example: GETDATE() and CURRENT\_TIMESTAMP

Javascript Interview

What are closures and how are they used?

**What**: A closure takes place when a function creates an environment that binds local variables to it in such a way that they are kept alive after the function has returned.

**When**: The most common example of this is when the inner function is being used to handle an event.

Can you use x === “object” to test if x is an object?

In short, yes, but you must take into account the fact that null is considered an object in JavaScript.

Even if x is null, ‘console.log(typeof x === “object”)’ will log true instead of false.

To account for this, you must also test whether or not x is null by including the following: **console.log((x !== null) && (typeof x === “object”));**

What happens when you don’t declare a variable in Javascript?

If you don’t explicitly declare a variable, you risk creating an implied global variable.

What is the difference between == and === ?

‘==’ evaluates equality of the value, while ‘===’ evaluates  equality of type and value.

What datatypes are supported in Javascript?

Number, String, Undefined, null, Boolean

Javascript object adding

**“1”+2+3🡪123,1+2+”3”🡪33, 5 + “ cats” = “5 cats”**

How to convert ‘20’ to 20 in Javascript

**Number()**

**parseInt(“20”)**

**parseFloat()**

What is the difference between window.onload and the jQuery $(document).ready() method?

**The window.onload** method occurs after all the page elements have loaded(HTML, CSS, images), which can result in a delay.

**The $(document).ready()** method begins to run code as soon as the Document Object Model (DOM) is loaded, which should be faster and less prone to loading errors across different browsers.

What is NaN?

Nan is literally “Not-a-Number”. The only real way to test if a value is equal to NaN is with the function ‘isNaN()’.

Explain the meaning of the keyword ‘this’ in JavaScript functions

The keyword ‘this’ in JavaScript refers to the object that a function is a method of. If it’s not specified, it will default to the global object, the window.

How to check this.object?

1. Was the function called with ‘new’? if so, use that object. New keyword is able to override any rules.
2. Was the function called with ‘call’ or ‘apply’ specifying an explicit this? If so, use that object(explicit binding)
3. Was the function called via a containing/owning object (context)? If so, use that object.(implicit binding)
4. DEFAULT: global object(except strict mode, undefined)

What is the difference between undefined value and null value?

**undefined** means a variable has been declared but has not yet been assigned a value. On the other hand, **null** is an assignment value. It can be assigned to a variable as a representation of no value.

Also, undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object. Unassigned variables are initialized by JavaScript with a default value of undefined. JavaScript never sets a value to null. That must be done programmatically.

How to create a class in javascript?

1. **Using a function**

This is probably one of the most common ways. You define a normal JavaScript function and then create an object by using the new keyword, defined method internally or added method to the prototype.

function Apple (type) {

this.type = type;

this.color = "red";

}

Apple.prototype.getInfo = function() {

return this.color + ' ' + this.type + ' apple';

};

1. **Using object literals.**

So you can skip the class-like stuff and create an instance (object) immediately.

var apple = {

type: "macintosh",

color: "red",

getInfo: function () {

return this.color + ' ' + this.type + ' apple';

}

}

In this case you don't need to (and cannot) create an instance of the class, it already exists. So you simply start using this instance.

apple.color = "reddish";

alert(apple.getInfo());

1. **Singleton using a function**

var apple = new function() {

this.type = "macintosh";

this.color = "red";

this.getInfo = function () {

return this.color + ' ' + this.type + ' apple';

};

}

apple.color = "reddish";

alert(apple.getInfo());

new function(){...} does two things at the same time: define a function (an anonymous constructor function) and invoke it with new.

New keyword in javascript

It creates a new object. The type of this object, is simply object.

It sets this new object's internal, inaccessible

It executes the constructor function, using the newly created object whenever this is mentioned.

It returns the newly created object, unless the constructor function returns a non-primitive value

Javascript synchronous

Javascript is always synchronous and single-threaded meaning if you're executing a Javascript block of code on a page then no other Javascript on that page will currently be executed.

Javascript is only asynchronous in the sense that it can make, for example, AJAX calls. The code will stop executing until the call returns (successfully or in error), at which point the callback will run synchronously.

**with the async: false**

**Disadvantage**

The reason it's problematic is that this will block **all** Javascript on the page until it finishes, included all event handlers and timers.

Write an simple function to determine string is palindrome

function isPalindrome(str) {

str = str.replace(/\W/g, '').toLowerCase();

return (str == str.split('').reverse().join(''));

}

What is the significance of including 'use strict' at the beginning of a JavaScript source file?

use strict is a way to enforce stricter parsing and error handling on your JavaScript code at runtime. Code errors that would otherwise have been ignored or would have failed silently will now generate errors or throw exceptions.

How do you reduce the load time of the javascript (.js files)

a.Avoid Inline JS and CSS files(A script tag without a src, An onclick="..." attribute is called an inline event handler.);

b.Optimize Caching;

c,Place JavaScript at the end of the Document;

d.Remove Unnecessary CSS and HTML;

e.Place Style Sheets at the top of Documents;

f.Minification of JavaScript and CSS

How to overload method in javascript?

The best way to do function overloading with parameters is not to check the argument length or the types; **checking the types will just make your code slow and you have the fun of Arrays, nulls, Objects, etc.**

What most developers do is tack on an object as the last argument to their methods. This object can hold anything.

function foo(a, b, opts) {

}

foo(1, 2, {"method":"add"});

foo(3, 4, {"test":"equals", "bar":"tree"});

Then you can handle it anyway you want in your method. [Switch, if-else, etc.]

Explain Directives in Angular?

Directives are attributes decorated on the HTML elements. All directives start with the word “ng”.

ng-app : Initializes application.  
ng-model : Binds HTML controls to application data.  
ng-Controller : Attaches a controller class to view.  
ng-repeat : Bind re peated data HTML elements. Its like a for loop.  
ng-if : Bind HTML elements with condition.  
ng-show : Used to show the HTML elements.  
ng-hide : Used to hide the HTML elements.  
ng-class : Used to assign CSS class.  
ng-src : Used to pass the URL image etc.

ng-click : Click event to bind on HTML elements.

Explain what is scope in Angular.js ?

Scope refers to the application model, it acts like glue between application controller and the view.  Scopes are arranged in hierarchical structure and impersonate the DOM (Document Object Model) structure of the application.  It can watch expressions and propagate events.

Explain what is services in Angular.js ?

In angular.js services are the singleton objects or functions that are used for carrying out specific tasks.  It holds some business logic and these function can be called as controllers, directive, filters and so on.

Explain what Angular JS routes does ?

Angular js routes enable you to create different URLs for different content in your application.  Different URLs for different content enables user to bookmark URLs to specific content.  Each such bookmarkable URL in Angular.js is called a route

What happened when you open [www.google.com](http://www.google.com)?

Step 1. URL is typed in the browser.

Step 2. If requested object is in browser cache and is fresh, move on to Step  8.

Step 3. DNS lookup to find the ip address of the server. when we want to connect to ***google.com***, we actually want to reach out to a server where google web services are hosted. Google has multiple servers in multiple locations to cater to the huge volume of requests they receive per second. Thus we should let Google decide which server is best suited to our needs. Using "google.com" does the job for us. When we type "google.com", [DNS(Domain Name System)](http://en.wikipedia.org/wiki/Domain_Name_System" \t "_blank) services comes into play and resolves the URL to a proper ip address.

***Check browser cache/ Check OS cache/ Router Cache/ ISP cache***

Step 4. Browser initiates a TCP connection with the server.

Browser sends a ***GET*** request to the server

Step 5. Browser sends a HTTP request to the server.

Step 6. Server handles the incoming request.

***HTTP*** request made from browsers are handled by a special software running on server - commonly known as ***web servers*** e.g. ***Apache***, ***IIS*** etc. Web server passes on the request to the proper request handler - a program written to handle web services e.g. ***PHP, ASP.NET, Ruby, Servlets*** etc.

Step 7. Browser receives the HTTP response

HTTP response starts with the returned status code from the server. Server sets various other headers to help browser render the proper content. ***Content-Type*** tells the type of the content the browser has to show, ***Content-length*** tells the number of bytes of the response. Using the ***Content-Encoding*** header's value, browsers can decode the blob data present at the end of the response

Step 8. Browsers displays the html content

Step 9. Client interaction with server

Once a html page is loaded, there are several ways a user can interact with the server. For example, he call fill out a login form to sign in to the website. This also follows all the steps listed above, the only difference is that the HTTP request this time would be a ***POST*** instead of GET and along with that request, browser will send the form data to the server for processing (username and password in this case).

Step 10. AJAX queries

This is an asynchronous GET/POST request to which server can send a response back in a variety of ways - ***json, xml, html*** etc. AJAX requests doesn't hinder the current view of the webpage and work in the background.