TOP 500

.NET

INTERVIEW QUESTIONS PART - II

HAPPY RAWAT

PREFACE

ABOUT THE BOOK

Part II - This book contains 250 more interview questions.

ABOUT THE AUTHOR

Happy Rawat has around 15 years of experience in software development. He helps candidates in clearing technical interview in tech companies.

Topic	Part II (Number of Questions)
OOPS/ C#	38
.NET FRAMEWORK	27
SQL	25
ASP.NET MVC	52
ASP.NET WEBFORMS	35
ADO.NET & EF	10
WEB API	37
.NET CORE	17
SOLID PRINCIPLES	0
DESIGN PATTERNS	9
TOTAL	**250**

OOPS

1. WHAT IS THE DIFFERENCE BETWEEN A CLASS AND A STRUCTURE?

🞣 CLASS: User-defined blueprint from which objects are created. It consists of methods or set of instructions that are to be performed on the objects.

🞣 STRUCTURE: A structure is basically a user-defined collection of variables which are of different data types.

---

2. WHAT IS OPERATOR OVERLOADING?

Operator overloading refers to implementing operators using user-defined types based on the arguments passed along with it.

---

3. CAN YOU CALL THE BASE CLASS METHOD WITHOUT CREATING AN INSTANCE?

- Yes, you can call the base class without instantiating it if:

🞣 It is a static method.
🞣 The base class is inherited by some other subclass.

---

4. WHAT ARE VIRTUAL FUNCTIONS AND PURE VIRTUAL FUNCTIONS?

Virtual functions are functions that are present in the parent class and are overridden by the subclass. These functions are used to achieve runtime polymorphism.

Pure virtual functions or abstract functions are functions that are only declared in the base class. This means that they do not contain any definition in the base class and need to be redefined in the subclass.

---

5. WHICH OOPS CONCEPT EXPOSES ONLY THE NECESSARY INFORMATION TO THE CALLING FUNCTIONS?

Encapsulation

---

6. IS IT ALWAYS NECESSARY TO CREATE OBJECTS FROM CLASS?

No, it is possible to call a base class method if it is defined as a static method.

---

7. WHAT IS EARLY AND LATE BINDING?

Early binding refers to the task of values to variables during plan time, whereas late Binding refers to the assignment of values to variables during run time.

---

8. WHICH OOPS CONCEPT IS USED AS A REUSE MECHANISM?

Inheritance is the OOPS concept that can be used as a reuse mechanism.

9. IN TRY BLOCK IF WE ADD RETURN STATEMENT WHETHER FINALLY BLOCK IS EXECUTED?

Yes. Finally block will still be executed in presence of return statement in try block.

10. WHAT YOU MEAN BY INNER EXCEPTION?

Inner exception is a property of exception class which will give you a brief insight of the exception i.e, parent exception and child exception details.

11. LIST OUT SOME OF THE EXCEPTIONS?

Below are some of the exceptions -

- NullReferenceException
- ArgumentNullException
- DivideByZeroException
- IndexOutOfRangeException
- InvalidOperationException
- StackOverflowException etc.

12. WHAT IS THE DIFFERENCE BETWEEN STACK AND QUEUE COLLECTIONS?

A stack is a linear data structure in which elements can be inserted and deleted only from one side of the list, called the **top**. A stack follows the **LIFO** (Last In First

Out) principle, i.e., the element inserted at the last is the first element to come out.

A queue is a linear data structure in which elements can be inserted only from one side of the list called **rear**, and the elements can be deleted only from the other side called the **front**. The queue data structure follows the **FIFO** (First In First Out) principle, i.e. the element inserted at first in the list, is the first element to be removed from the list.

13. EXPLAIN JAGGED ARRAYS?

If the elements of an array is an array then it's called as jagged array. The elements can be of different sizes and dimensions.

14. EXPLAIN MULTIDIMENSIONAL ARRAYS?

A multi-dimensional array in C# is an array that contains more than one rows to store the data. The multidimensional array can be declared by adding commas in the square brackets.

15. EXPLAIN INDEXERS?

Indexers are used for allowing the classes to be indexed like arrays. Indexers will resemble the property structure but only difference is indexer's accessors will take parameters. For example,

```
class MyCollection<T>
{
 private T[] myArr = new T[100];
 public T this[int t]
```

```
{
get
{
return myArr[t];
}
set
{
myArr[t] = value;
}
}
}
```

16. WHAT IS THE DIFFERENCE BETWEEN METHODS – "SYSTEM.ARRAY.CLONE()" AND "SYSTEM.ARRAY.COPYTO()"?

- "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.

17. WHAT ARE VALUE TYPE AND REFERENCE TYPES?

Value types stored in a stack.

Here are the value types:

- Decimal, int, byte, enum, double, long, float

Reference types are stored in a heap.

Below are the list of reference types -

- Class, string, interface, object

18. CAN WE OVERRIDE PRIVATE VIRTUAL METHOD?

No. We can't override private virtual methods as it is not accessible outside the class.

19. EXPLAIN CIRCULAR REFERENCE?

This is a situation where in, multiple resources are dependent on each other and this causes a lock condition and this makes the resource to be unused.

20. EXPLAIN OBJECT POOL?

Object pool is used to track the objects which are being used in the code. So object pool reduces the object creation overhead.

21. WHAT ARE THE TYPES OF DELEGATES?

Below are the uses of delegates -

- Single Delegate
- Multicast Delegate
- Generic Delegate

22. WHAT ARE THE THREE TYPES OF GENERIC DELEGATES?

Below are the three types of generic delegates -

- Func

- Action
- Predicate

---

23. WHAT ARE THE USES OF DELEGATES?

Below are the list of uses of delegates -

- Callback Mechanism
- Asynchronous Processing
- Abstract and Encapsulate method
- Multicasting

---

24. CAN WE USE DELEGATES FOR ASYNCHRONOUS METHOD CALLS?

Yes. We can use delegates for asynchronous method calls.

---

25. WHAT IS AN ESCAPE SEQUENCE? NAME SOME STRING ESCAPE SEQUENCES IN C#.

An Escape sequence is denoted by a backslash (\). The backslash indicates that the character that follows it should be interpreted literally or it is a special character. An escape sequence is considered as a single character.

---

26. WHAT ARE REGULAR EXPRESSIONS?

Regular expression is a template to match a set of input. The pattern can consist of operators, constructs or character literals. Regex is used for string parsing and replacing the character string.

---

27. WHY TO USE LOCK STATEMENT?

Lock will make sure one thread will not intercept the other thread which is running the part of code. So lock statement will make the thread wait, block till the object is being released.

---

28. WHAT IS "EXTERN" KEYWORD?

The `extern` modifier is used to declare a method that is implemented externally. A common use of the `extern` modifier is with the `DllImport` attribute when you are using Interop services to call into unmanaged code.

```
[DllImport("avifil32.dll")]
private static extern void AVIFileInit();
```

---

29. WHAT IS "SIZEOF" OPERATOR?

The **sizeof operator** returns the number of bytes occupied by a variable of a given type. The argument to the **sizeof operator** must be the name of an unmanaged type or a type parameter that is constrained to be an unmanaged type.

```
Console.WriteLine(sizeof(byte));   // output: 1
```

---

30. CAN WE USE "THIS" INSIDE A STATIC METHOD?

No. We can't use "this" in static method.

---

31. WHAT IS THE DIFFERENCE BETWEEN CTYPE AND DIRECTCAST?

- CType is used for conversion between type and the expression.
- Directcast is used for converting the object type which requires run time type to be the same as specified type.

---

32. WHICH STRING METHOD IS USED FOR CONCATENATION OF TWO STRINGS?

"Concat" method of String class is used to concatenate two strings. For example,

string.Concat(firstStr, secStr)

---

33. WHAT IS PARSING? HOW TO PARSE A DATE TIME STRING?

Parsing converts a string into another data type.

For Example:

string text = "500";
int num = int.Parse(text);
500 is an integer. So, the Parse method converts the string 500 into its own base type, i.e int.

Follow the same method to convert a DateTime string.
string dateTime = "Jan 1, 2018";
DateTime parsedValue = DateTime.Parse(dateTime);

34. EXPLAIN PARTIAL CLASS?

Partial classes concept added in .Net Framework 2.0 and it allows us to split the business logic in multiple files with the same class name along with "partial" keyword.

35. EXPLAIN ANONYMOUS TYPE?

This is being added 3.0 version. This feature enables us to create an object at compile time. Below is the sample code for the same –

Var myTestCategory = new { CategoryId = 1, CategoryName = "Category1"};

36. EXPLAIN GET AND SET ACCESSOR PROPERTIES?

Get and Set are called Accessors. These are made use by Properties. The property provides a mechanism to read, write the value of a private field. For accessing that private field, these accessors are used.

Get Property is used to return the value of a property
Set Property accessor is used to set the value.

37. WHAT IS THE DIFFERENCE BETWEEN VAR AND DYNAMIC IN C#?

VAR TYPE - VAR are those variables which are declared without specifying the *.NET type* explicitly. In implicitly typed variable, the type of the variable is automatically deduced at compile time by the compiler from the value used to initialize the variable.

```
var a = 'f';
```

DYNAMIC TYPE - It is used to avoid the compile-time type checking. The compiler does not check the type of the dynamic type variable at compile time, instead of this, the compiler gets the type at the run time. The dynamic type variable is created using dynamic keyword.

```
dynamic val1 = "Interview";
```

38. WHAT IS COVARIANCE IN C#?

Covariance enables you to pass a derived type where a base type is expected.

Small sml = new Bigger();

.NET FRAMEWORK

39. WHAT IS JIT?

JIT stands for **Just In Time**. JIT is a compiler that converts Intermediate Language to a Native code.

The code is converted into Native language during execution. Native code is nothing but hardware specifications that can be read by the CPU. The native code can be stored so that it is accessible for subsequent calls.

40. WHAT IS MSIL?

MSIL stands for Microsoft Intermediate Language.

MSIL provides instructions for calling methods, initializing and storing values, operations such as memory handling, exception handling and so on. All .Net codes are first compiled to IL.

41. WHAT IS MEANT BY MANAGED AND UNMANAGED CODE?

The code that is managed by the CLR is called **Managed code**. This code runs inside the CLR. Hence, it is necessary to install the .Net framework in order to execute the managed code. CLR manages the memory through garbage collection and also uses the other features like CAS and CTS for efficient management of the code.

Unmanaged code is any code that does not depend on CLR for execution. It means it is developed by any other language independent of .Net framework. It uses its own runtime environment for compiling and execution.

Though it is not running inside the CLR, the unmanaged code will work properly if all the other parameters are correctly followed.

42. WHAT IS DIFFERENCE BETWEEN NAMESPACE AND ASSEMBLY?

Following are the differences between namespace and assembly:

• Assembly is physical grouping of logical units, Namespace, logically groups classes.

• Namespace can span multiple assembly.

43. WHAT IS MANIFEST?

Assembly metadata is stored in Manifest. Manifest contains all the metadata needed to do the

following things (See Figure Manifest View for more details):

• Version of assembly.

• Security identity.

• Scope of the assembly.

• Resolve references to resources and classes.

44. WHERE IS VERSION INFORMATION STORED OF AN ASSEMBLY?

Version information is stored in assembly inside the manifest.

45. IS VERSIONING APPLICABLE TO PRIVATE ASSEMBLIES?

Versioning concept is only applicable to global assembly cache (GAC) as private assembly lie in

their individual folders. This does not mean versioning is not needed , you can still version it to

have better version control on the project.

---

46. WHAT IS THE APPLICATION DOMAIN?

An Application Domain is a logical container for a set of assemblies in which an executable is hosted. As you have seen, a single process may contain multiple Application Domains, each of which is hosting a .NET executable.

---

47. EXPLAIN CAS (CODE ACCESS SECURITY).

.Net provides a security model that prevents unauthorized access to resources. CAS is a part of that security model. CAS is present in the CLR. It enables the users to set permissions at a granular level for the code.

---

48. WHAT ARE SYNCHRONOUS AND ASYNCHRONOUS OPERATIONS?

1. Synchronization is a way to create a thread-safe code where only one thread can access the resource at any given time. The asynchronous call waits for the method to complete before continuing with the program flow.
2. Synchronous programming badly affects the UI operations when the user tries to perform time-consuming operations since only one thread will be used. In Asynchronous operation, the method call will immediately return so

that the program can perform other operations while the called method completes its work in certain situations.

49. WHAT IS MULTI-TASKING?

It is a feature of modern operating systems with which we can run multiple programs at same

time example Word, Excel etc.

50. NAME SOME PROPERTIES OF THREAD CLASS.

Few Properties of thread class are:

- **IsAlive** – contains value True when a thread is Active.
- **Name** – Can return the name of the thread. Also, can set a name for the thread.
- **Priority** – returns the prioritized value of the task set by the operating system.
- **IsBackground** – gets or sets a value which indicates whether a thread should be a background process or foreground.
- **ThreadState**– describes the thread state.

51. WHAT ARE THE DIFFERENT STATES OF A THREAD?

Different states of a thread are:

- **Unstarted** – Thread is created.
- **Running** – Thread starts execution.
- **WaitSleepJoin** – Thread calls sleep, calls wait on another object and calls join on another thread.
- **Suspended** – Thread has been suspended.

- **Aborted** – Thread is dead but not changed to state stopped.
- **Stopped** – Thread has stopped.

---

52. CAN WE HAVE MULTIPLE THREADS IN ONE APP DOMAIN?

One or more threads run in an AppDomain. An AppDomain is a runtime representation of a

logical process within a physical process. Each AppDomain is started with a single thread, but

can create additional threads from any of its threads.

---

53. WHICH NAMESPACE HAS THREADING?

'Systems.Threading' has all the classes related to implement threading.

---

54. EXPLAIN *LOCK, MONITORS,* AND *MUTEX* OBJECT IN THREADING.

Lock keyword ensures that only one thread can enter a particular section of the code at any given time. In the above **Example**, lock(ObjA) means the lock is placed on ObjA until this process releases it, no other thread can access ObjA.

Mutex is also like a lock but it can work across multiple processes at a time. WaitOne() is used to lock and ReleaseMutex() is used to release the lock. But Mutex is slower than lock as it takes time to acquire and release it.

Monitor.Enter and Monitor.Exit implements lock internally. a lock is a shortcut for Monitors. lock(objA) internally calls.

Monitor.Enter(ObjA);

try

{

}

Finally {Monitor.Exit(ObjA));}

---

55. WHAT IS THREAD.SLEEP () IN THREADING?

Thread's execution can be paused by calling the Thread.Sleep method. This method takes an

integer value that determines how long the thread should sleep. Example

Thread.CurrentThread.Sleep(2000).

---

56. WHAT IS SUSPEND AND RESUME IN THREADING?

Suspend() method is called to suspend the thread. Resume() method is called to resume the suspended thread. Start() method is used to send a thread into runnable State.

---

57. WHAT THE WAY TO STOP A LONG RUNNING THREAD?

Thread.Abort() stops the thread execution at that moment itself.

---

58. WHY WE NEED MULTI-THREADING IN OUR PROJECT?

Multi-threading is running the multiple threads simultaneously. Some main advantages are:

- You can do multiple tasks simultaneously. For e.g. saving the details of user to a file while at the same time retrieving something from a web service.
- Threads are much lightweight than process. They don't get their own resources. They used the resources allocated to a process.
- Context-switch between threads takes less time than process.

59. HOW TO START A THREAD IN C#?

We have to use the Thread class provided by System.Threading namespace. In the constructor of the class, we have to pass the method name which we want to run in separate thread. After than we have to call the start method of Thread class. Below is the example.

60. WHAT IS RACE CONDITION?

A race condition happens when two or more threads want to update shared data at the same time.

61. WHAT IS LIVELOCK?

A livelock is very similar to deadlock except involved threads states are continually changing their state but still they cannot complete their work.

A real-world example of livelock occurs when two people meet in a narrow corridor, and each tries to be polite by moving aside to let the other pass, but

they end up swaying from side to side without making any progress because they both repeatedly move the same way at the same time.

62. WHAT IS IMMUTABLE OBJECT?

An immutable object is an object which states cannot be changed after creation.

Immutable objects are useful in concurrent programming as they are thread-safe.

"String" objects are examples of immutable object because we can cannot change the value of string after it is created.

63. HOW MANAGED CODE IS EXECUTED?

Managed code is a code whose execution is managed by Common Language Runtime. It gets the managed code and compiles it into machine code. After that, the code is executed. The runtime here i.e. CLR provides automatic memory management, type safety, etc.

64. WHAT IS THE DIFFERENCE BETWEEN SYSTEM EXCEPTIONS AND APPLICATION EXCEPTIONS?

In general System exceptions occurred whenever some non-recoverable or fatal error is encountered, like a database crash, bound errors etc.

While in case of Application level exceptions some error which is recoverable is encountered, for instance, the wrong type of input data, arithmetic exceptions etc.

65. WHAT IS XSD?

XSD (XML Schema Definition), a recommendation of the World Wide Web Consortium (W3C), specifies how to formally describe the elements in an Extensible Markup Language (XML) document. It can be used by programmers to verify each piece of item content in a document.

ASP.NET MVC

66. IN WHICH ASSEMBLY IS THE MVC FRAMEWORK DEFINED?

System.Web.Mvc

---

67. IS IT POSSIBLE TO SHARE A VIEW ACROSS MULTIPLE CONTROLLERS?

Yes, put the view into the shared folder. This will automatically make the view available across multiple controllers.

---

68. WHAT IS ROUTE IN MVC? WHAT IS DEFAULT ROUTE IN MVC?

A route is a URL pattern that is mapped to a handler. The handler can be a physical file, such as a .aspx file in a Web Forms application. A handler can also be a class that processes the request, such as a controller in an MVC application. To define a route, you create an instance of the Route class by specifying the URL pattern, the handler, and optionally a name for the route.

You add the route to the application by adding the Route object to the static Routes property of the RouteTable class. The Routesproperty is a RouteCollection object that stores all the routes for the application.

You typically do not have to write code to add routes in an MVC application. Visual Studio project templates for MVC include preconfigured URL routes. These are defined in the MVC Application class, which is defined in the Global.asax file.
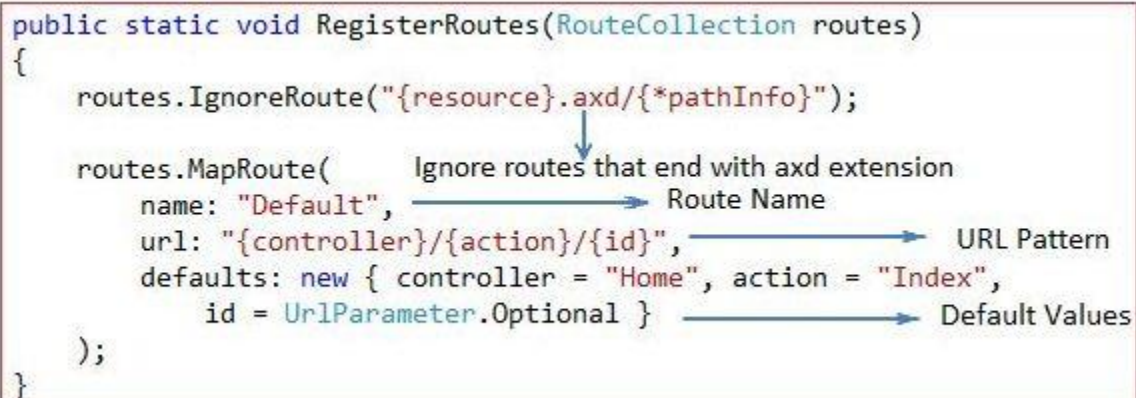
Default Route

The default ASP.NET MVC project templates add a generic route that uses the following URL convention to break the URL for a given request into three named segments.

URL: "{controller}/{action}/{id}"

This route pattern is registered via a call to the MapRoute() extension method of RouteCollection.

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

                                Ignore routes that end with axd extension
    routes.MapRoute(
        name: "Default",  ──────────────────────►  Route Name
        url: "{controller}/{action}/{id}",──────────► URL Pattern
        defaults: new { controller = "Home", action = "Index",
            id = UrlParameter.Optional } ──────────► Default Values
    );
}
```

69. WHERE ARE THE ROUTING RULES DEFINED IN AN ASP.NET MVC APPLICATION?

In Application_Start event in Global.asax

70. WHAT IS VALIDATION SUMMARY IN MVC?

The ValidationSummary helper method generates an unordered list (ul element) of validation messages that are in the ModelStateDictionary object.

The ValidationSummary can be used to display all the error messages for all the fields. It can also be used to display custom error messages.

DIFFERENCES BETWEEN RAZOR AND ASPX VIEW ENGINE IN MVC?

Razor View Engine VS ASPX View Engine

The Razor View Engine is a bit slower than the ASPX View Engine.

Conclusion

Razor provides a new view engine with a streamlined code for focused templating. Razor's syntax is very compact and improves the readability of the markup and code. By default, MVC supports ASPX (web forms) and Razor View Engine. MVC also supports third-party view engines like Spark, Nhaml, NDjango, SharpDOM and so on. ASP.NET MVC is open source.

72.WHAT ARE THE MAIN RAZOR SYNTAX RULES?

- Razor code blocks are enclosed in @{ ... }
- Inline expressions (variables and functions) start with @
- Code statements end with semicolon
- Variables are declared with the var keyword

- Strings are enclosed with quotation marks
- C# code is case sensitive
- C# files have the extension .cshtml

73.HOW DO YOU IMPLEMENT FORMS AUTHENTICATION IN MVC?

Authentication is giving access to the user for a specific service by verifying his/her identity using his/her credentials like username and password or email and password. It assures that the correct user is authenticated or logged in for a specific service and the right service has been provided to the specific user based on their role that is nothing but authorization.

ASP.NET forms authentication occurs after IIS authentication is completed. You can configure forms authentication by using forms element with in web.config file of your application. The default attribute values for forms authentication are shown below,

<system.web>

<authenticationmode="Forms">

<formsloginUrl="Login.aspx" protection="All" timeout="30" name=".ASPXAUTH" path="/" requireSSL="false" slidingExpiration="true" defaultUrl="default.aspx" cookieless="UseDeviceProfile" enableCrossAppRedirects="false" />

</authentication>

</system.web>

The FormsAuthentication class creates the authentication cookie automatically when SetAuthCookie() or RedirectFromLoginPage() methods are called. The value of authentication cookie contains a string representation of the encrypted and signed FormsAuthenticationTicket object.

74.HOW WE CAN REGISTER THE AREA IN ASP.NET MVC?

When we have created an area make sure this will be registered in "Application_Start" event in Global.asax. Below is the code snippet where area registration is done :

```
protected void Application_Start()
{
AreaRegistration.RegisterAllAreas();
}
```

75.EXPLAIN THE NEED OF DISPLAY MODE IN MVC?

DisplayModes give you another level of flexibility on top of the default capabilities we saw in the last section. DisplayModes can also be used along with the previous feature so we will simply build off of the site we just created.

Using display modes involves in 2 steps

1. We should register Display Mode with a suffix for particular browser using "DefaultDisplayMode"e class inApplication_Start() method in the Global.asax file.

2. View name for particular browser should be appended with suffix mentioned in first step.

1. Desktop browsers (without any suffix. e.g.: Index.cshtml, _Layout.cshtml).
2. Mobile browsers (with a suffix "Mobile". e.g.: Index.Mobile.cshtml,Layout.Mobile.cshtml)

If you want design different pages for different mobile device browsers (any different browsers) and render them depending on the browser requesting. To handle these requests you can register custom display modes. We can do that using DisplayModeProvider.Instance.Modes.Insert(int index, IDisplayMode item) method.

76. WHAT IS ROUTE CONSTRAINTS IN MVC?

Routing is a great feature of MVC, it provides a REST based URL that is very easy to remember and improves page ranking in search engines.

This article is not an introduction to Routing in MVC, but we will learn a few features of routing and by implementing them we can develop a very flexible and user-friendly application. So, let's start without wasting valuable time.

Add constraint to URL

This is very necessary for when we want to add a specific constraint to our URL. Say, for example we want a [URL](URL).

So, we want to set some constraint string after our host name. Fine, let's see how to implement it.

It's very simple to implement, just open the RouteConfig.cs file and you will find the routing definition in that. And modify the routing entry as in the following. We will see that we have added "abc" before.

```
17          routes.MapRoute(
18              name: "Default",
19              url: "abc/{controller}/{action}/{id}",
20              defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
21          );
22
23      }
```

Controller name, now when we browse we need to specify the string in the URL, as in the following:

localhost:2968/abc/person/

I am simple view.
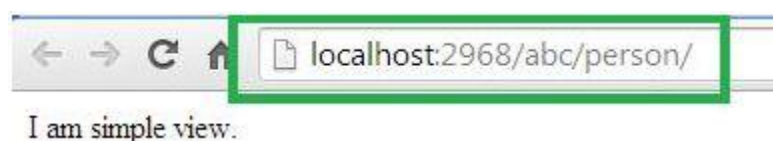
Understanding the folders

When you create a project a folder structure gets created by default under the name of your project which can be seen in solution explorer. Below i will give you a brief explanation of what these folders are for.

Model

This folder contains classes that is used to provide data. These classes can contain data that is retrived from the database or data inserted in the form by the user to update the database.

Controllers

These are the classes which will perform the action invoked by the user. These classes contains methods known as "Actions" which responds to the user action accordingly.

Views

These are simple pages which uses the model class data to populate the HTML controls and renders it to the client browser.

App_Start

Contains Classes such as FilterConfig, RoutesConfig, WebApiConfig. As of now we need to understand the RouteConfig class. This class contains the default format of the url that should be supplied in the browser to navigate to a specified page.

78. WHAT ARE THE METHODS OF HANDLING AN ERROR IN MVC?

Exception handling may be required in any application, whether it is a web application or a Windows Forms application.

ASP.Net MVC has an attribute called "HandleError" that provides built-in exception filters. The HandleError attribute in ASP.NET MVC can be applied over the action method as well as Controller or at the global level. The HandleError attribute is the default implementation of IExceptionFilter. When we create a MVC application, the HandleError attribute is added within the Global.asax.cs file and registered in the Application_Start event.

```
1. public static void RegisterGlobalFilters(GlobalFilterCollection filters)

2. {
3.     filters.Add(new HandleErrorAttribute());
4. }
5. protected void Application_Start()
6. {
7.     AreaRegistration.RegisterAllAreas();
8.     RegisterGlobalFilters(GlobalFilters.Filters);
9.     RegisterRoutes(RouteTable.Routes);
10. }
```

Important properties of HandleError attribute

The HandleError Error attribute has a couple for properties that are very useful in handling the exception.

ExceptionType

Type of exception to be catch. If this property is not specified then the HandleError filter handles all exceptions.

View

Name of the view page for displaying the exception information.

Master

Master View for displaying the exception.

Order

Order in which the action filters are executed. The Order property has an integer value and it specifies the priority from 1 to any positive integer value. 1 means highest priority and the greater the value of the integer is, the lower is the priority of the filter.

AllowMultiple

It indicates whether more than one instance of the error filter attribute can be specified.

Example

```
[HandleError(View = "Error")]

public class HomeController: Controller

{

  public ActionResult Index()

  {

    ViewBag.Message = "Welcome to ASP.NET MVC!";

    int u = Convert.ToInt32(""); // Error line

    return View();

  }

}
```

HandleError Attribute at Action Method Level,

```
[HandleError(View = "Error")]

public ActionResult Index()

{

  ViewBag.Message = "Welcome to ASP.NET MVC!";

  int u = Convert.ToInt32(""); // Error line

  return View();

}
```

Razor View Engine introduced a new layout named _ViewStart which is applied on all view automatically. Razor View Engine firstly executes the _ViewStart and then start rendering the other view and merges them.

Example of Viewstart

```
@ {
    Layout = "~/Views/Shared/_v1.cshtml";
} < !DOCTYPE html >
    < html >
    < head >
    < meta name = "viewport"
content = "width=device-width" / >
    < title > ViewStart < /title> < /head> < body >
    < /body> < /html>
```

80.HOW CAN WE DONE CUSTOM ERROR PAGE IN MVC?

The HandleErrorAttribute allows you to use a custom page for this error. First you need to update your web.config file to allow your application to handle custom errors.

1. <system.web>
2.   <customErrors mode="On">
3. </system.web>

Then, your action method needs to be marked with the atttribute.

1. [HandleError]
2. **public class** HomeController: Controller
3. {
4.   [HandleError]
5.   publicActionResultThrowException()
6.   {
7.   **throw new** ApplicationException();
8.   }
9. }

By calling the ThrowException action, this would then redirect the user to the default error page. In our case though, we want to use a custom error page and redirect the user there instead.So, let's create our new custom view page.

```
<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://r
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>CustomErrorView</title>
</head>
<body>
    <h2>
        Error</h2>
    <p>
        Controller:
        <%=((HandleErrorInfo)ViewData.Model).ControllerName %>
    </p>
    <p>
        Action:
        <%=((HandleErrorInfo)ViewData.Model).ActionName %>
    </p>
    <p>
        Message:
        <%=((HandleErrorInfo)ViewData.Model).Exception.Message %>
    </p>
    <p>
        Stack Trace:
        <%=((HandleErrorInfo)ViewData.Model).Exception.StackTrace %>
    </p>
</body>
</html>
```

Next, we simply need to update the HandleErrorAttribute on the action method.

```
1.  [HandleError]
2.  public class HomeController: Controller
3.  {
4.     [HandleError(View = "CustomErrorView")]
5.     publicActionResultThrowException()
6.     {
7.        throw new ApplicationException();
8.     }
9.  }
```

81. WHAT IS SERVER SIDE VALIDATION IN MVC?

The ASP.NET MVC Framework validates any data passed to the controller action that is executing, It populates a ModelState object with any validation failures that it finds and passes that object to the controller. Then the controller actions can query the ModelState to discover whether the request is valid and react accordingly.

I will use two approaches in this article to validate a model data. One is to manually add an error to the ModelState object and another uses the Data Annotation API to validate the model data.

Approach 1 - Manually Add Error to ModelState object

I create a User class under the Models folder. The User class has two properties "Name" and "Email". The "Name" field has required field validations while the "Email" field has Email validation. So let's see the procedure to implement the validation. Create the User Model as in the following,

```
1.  namespace ServerValidation.Models
2.  {
3.    public class User
4.    {
5.      public string Name
6.      {
7.        get;
8.        set;
9.      }
10.     public string Email
11.     {
12.       get;
13.       set;
14.     }
15.   }
16. }
```

After that I create a controller action in User Controller (UserController.cs under Controllers folder). That action method has logic for the required validation for Name and Email validation on the Email field. I add an error message on ModelState with a key and that message will be shown on the view whenever the data is not to be validated in the model.

```
1.  using System.Text.RegularExpressions;
2.  using System.Web.Mvc;
3.  namespace ServerValidation.Controllers
4.  {
5.    public class UserController: Controller
6.    {
7.      public ActionResult Index()
8.      {
9.        return View();
10.     }
```

```
11.        [HttpPost]
12.      public ActionResult Index(ServerValidation.Models.User model)
13.      {
14.
15.        if (string.IsNullOrEmpty(model.Name))
16.        {
17.          ModelState.AddModelError("Name", "Name is required");
18.        }
19.        if (!string.IsNullOrEmpty(model.Email))
20.        {
21.          string emailRegex = @ "^([a-zA-Z0-9_\-\.]+)@((\[[0-
   9]{1,3}" +
22.              @ "\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-
   ]+\" +
23.          @ ".)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$";
24.          Regex re = new Regex(emailRegex);
25.          if (!re.IsMatch(model.Email))
26.          {
27.            ModelState.AddModelError("Email", "Email is not val
   id");
28.          }
29.        } else {
30.          ModelState.AddModelError("Email", "Email is required")
   ;
31.        }
32.        if (ModelState.IsValid)
33.        {
34.          ViewBag.Name = model.Name;
35.          ViewBag.Email = model.Email;
36.        }
37.        return View(model);
38.      }
39.  }
40.}
```

Thereafter I create a view (Index.cshtml) for the user input under the User folder.

```
1.  @model ServerValidation.Models.User
2.  @ {
3.     ViewBag.Title = "Index";
4.  }
5.  @using(Html.BeginForm())
6.  {
7.     if (@ViewData.ModelState.IsValid)
8.        {
9.        if (@ViewBag.Name != null)
10.   { < b >
11.          Name: @ViewBag.Name < br / >
12.          Email: @ViewBag.Email < /b>
13.       }
14.   } < fieldset >
15.     < legend > User < /legend>  < div class = "editor-label" >
16.     @Html.LabelFor(model => model.Name) < /div> < div class = "edi
   tor-field" >
17.     @Html.EditorFor(model => model.Name)
18.   @if(!ViewData.ModelState.IsValid)
19. {
20. < span class = "field-validation-
   error" > @ViewData.ModelState["Name"].Errors[0].ErrorMessage < /spa
   n>
21.
22. }
23. < /div>  < div class = "editor-label" >
24.
25.     @Html.LabelFor(model => model.Email) < /div> < div class = "edi
   tor-field" >
26.     @Html.EditorFor(model => model.Email)
27.   @if(!ViewData.ModelState.IsValid)
28. {
29. < span class = "field-validation-
   error" > @ViewData.ModelState["Email"].Errors[0].ErrorMessage < /sp
   an>
30.   }
```

```
31. < /div> < p >
32.      < input type = "submit"
33.   value = "Create" / >
34.        < /p> < /fieldset>
35.}
```

WHAT IS THE USE OF REMOTE VALIDATION IN MVC?

Remote validation is the process where we validate specific data posting data to a server without posting the entire form data to the server. Let's see an actual scenario, in one of my projects I had a requirement to validate an email address, whetehr it already exists in the database. Remote validation was useful for that; without posting all the data we can validate only the email address supplied by the user.

Practical Explanation

Let's create a MVC project and name it accordingly, for me its "TestingRemoteValidation". Once the project is created let's create a model named UserModel that will look like:

```
1. public class UserModel
2. {
3.    [Required]
4.    public string UserName
5.    {
6.       get;
7.       set;
8.    }
9.    [Remote("CheckExistingEmail", "Home", ErrorMessage = "Email
   already exists!")]
```

```
10.   public string UserEmailAddress
11.   {
12.     get;
13.     set;
14.   }
15.}
```

Let's get some understanding of the remote attribute used, so the very first parameter "CheckExistingEmail" is the the name of the action. The second parameter "Home" is referred to as controller so to validate the input for the UserEmailAddress the "CheckExistingEmail" action of the "Home" controller is called and the third parameter is the error message. Let's implement the "CheckExistingEmail" action result in our home controller.

```
1.  public ActionResult CheckExistingEmail(string UserEmailAddress)
2.  {
3.    bool ifEmailExist = false;
4.    try
5.    {
6.       ifEmailExist = UserEmailAddress.Equals("mukeshknayak@gmail.c
   om") ? true : false;
7.       return Json(!ifEmailExist, JsonRequestBehavior.AllowGet);
8.    } catch (Exception ex)
9.    {
10.      return Json(false, JsonRequestBehavior.AllowGet);
11.   }
12.}
```

83.EXPLAIN RENDERSECTION IN MVC?

RenderSection() is a method of the WebPageBase class. Scott wrote at one point, The first parameter to the "RenderSection()" helper method specifies the name of the section we want to render at that location in the layout template. The second

parameter is optional, and allows us to define whether the section we are rendering is required or not. If a section is "required", then Razor will throw an error at runtime if that section is not implemented within a view template that is based on the layout file (that can make it easier to track down content errors). It returns the HTML content to render.

```
1. <div id="body">
2.    @RenderSection("featured", required: false)
3.    <section class="content-wrapper main-content clear-
   fix">
4.        @RenderBody()
5.    </section>
6. </div>
```

---

84. WHAT IS THE SIGNIFICANCE OF NONACTIONATTRIBUTE?

In general, all public methods of a controller class are treated as action methods. If you want prevent this default behavior, just decorate the public method with NonActionAttribute.

---

85. HOW ROUTE TABLE IS CREATED IN ASP.NET MVC?

When an MVC application first starts, the Application_Start() method is called. This method, in turn, calls the RegisterRoutes() method. The RegisterRoutes() method creates the route table.

---

86. ASP.NET MVC APPLICATION, MAKES USE OF SETTINGS AT 2 PLACES FOR ROUTING TO WORK CORRECTLY. WHAT ARE THESE 2 PLACES?

1. Web.Config File : ASP.NET routing has to be enabled here.
2. Global.asax File : The Route table is created in the application Start event handler, of the Global.asax file.

---

87. WHAT IS THE USE OF THE FOLLOWING DEFAULT ROUTE?

{resource}.axd/{*pathInfo}

This route definition, prevent requests for the Web resource files such as WebResource.axd or ScriptResource.axd from being passed to a controller.

---

88. WHAT IS THE DIFFERENCE BETWEEN ADDING ROUTES, TO A WEBFORMS APPLICATION AND TO AN MVC APPLICATION?

To add routes to a webforms application, we use MapPageRoute() method of the RouteCollection class, where as to add routes to an MVC application we use MapRoute() method.

---

89. IF I HAVE MULTIPLE FILTERS IMPLEMENTED, WHAT IS THE ORDER IN WHICH THESE FILTERS GET EXECUTED?

1. Authorization filters
2. Action filters
3. Response filters
4. Exception filters

---

90. WHICH FILTER EXECUTES FIRST IN AN ASP.NET MVC APPLICATION?

Authorization filter

91. WHAT ARE THE LEVELS AT WHICH FILTERS CAN BE APPLIED IN AN ASP.NET MVC APPLICATION?

1. Action Method
2. Controller
3. Application

92. IS IT POSSIBLE TO CREATE A CUSTOM FILTER?

Yes

93. WHAT FILTERS ARE EXECUTED IN THE END?

Exception Filters

94. IS IT POSSIBLE TO CANCEL FILTER EXECUTION?

Yes

95. WHAT TYPE OF FILTER DOES OUTPUTCACHEATTRIBUTE CLASS REPRESENTS?

Result Filter

96. WHAT ARE THE 2 POPULAR ASP.NET MVC VIEW ENGINES?

1. Razor
2. .aspx

97. WHAT SYMBOL WOULD YOU USE TO DENOTE, THE START OF A CODE BLOCK IN RAZOR VIEWS?

@

98. WHAT SYMBOL WOULD YOU USE TO DENOTE, THE START OF A CODE BLOCK IN ASPX VIEWS?

<%= %>

In razor syntax, what is the escape sequence character for @ symbol?
The escape sequence character for @ symbol, is another @ symbol

99. WHAT ARE SECTIONS?

Layout pages, can define sections, which can then be overriden by specific views making use of the layout. Defining and overriding sections is optional.

100. WHAT ARE THE FILE EXTENSIONS FOR RAZOR VIEWS?

1. .cshtml – If the programming lanugaue is C#
2. .vbhtml – If the programming lanugaue is VB

101.   HOW DO YOU SPECIFY COMMENTS USING RAZOR SYNTAX?

Razor syntax makes use of @* to indicate the begining of a comment and *@ to indicate the end.

---

102.   IS IT POSSIBLE TO COMBINE ASP.NET WEBFORMS AND ASP.MVC AND DEVELOP A SINGLE WEB APPLICATION?

Yes, it is possible to combine ASP.NET webforms and ASP.MVC and develop a single web application.

---

103.   WHAT IS THE USE OF VIEWMODEL IN MVC?

ViewModel is a plain class with properties, which is used to bind it to strongly typed view. ViewModel can have the validation rules defined for its properties using data annotations.

---

104.   EXPLAIN BUNDLE.CONFIG IN MVC4?

"BundleConfig.cs" in MVC4 is used to register the bundles by the bundling and minification system. Many bundles are added by default including jQuery libraries like – jquery.validate, Modernizr, and default CSS references.

---

105.   HOW WE CAN HANDLE THE EXCEPTION AT CONTROLLER LEVEL IN ASP.NET MVC?

Exception Handling is made simple in ASP.Net MVC and it can be done by just overriding "OnException" and set the result property of the filtercontext object (as shown below) to the view detail, which is to be returned in case of exception.

```
protected overrides void OnException(ExceptionContext filterContext)
{
}
```

106.   DOES TEMPDATA HOLD THE DATA FOR OTHER REQUEST IN ASP.NET MVC?

If Tempdata is assigned in the current request then it will be available for the current request and the subsequent request and it depends whether data in TempData read or not. If data in Tempdata is read then it would not be available for the subsequent requests.

107.   EXPLAIN KEEP METHOD IN TEMPDATA IN ASP.NET MVC?

As explained above in case data in Tempdata has been read in current request only then "Keep" method has been used to make it available for the subsequent request.

@TempData["TestData"];
TempData.Keep("TestData");

108.   EXPLAIN PEEK METHOD IN TEMPDATA IN ASP.NET MVC?

Similar to Keep method we have one more method called "Peek" which is used for the same purpose. This method used to read data in Tempdata and it maintains the data for subsequent request.

string A4str = TempData.Peek("TT").ToString();

109.   WHAT ARE CHILD ACTIONS IN ASP.NET MVC?

To create reusable widgets child actions are used and this will be embedded into the parent views. In ASP.Net MVC Partial views are used to have reusability in the application. Child action mainly returns the partial views.

110.   EXPLAIN THE TOOLS USED FOR UNIT TESTING IN ASP.NET MVC?

Below are the tools used for unit testing :

NUnit
xUnit.NET
Ninject 2
Moq

111.   CAN I USE RAZOR CODE IN JAVASCRIPT IN ASP.NET MVC?

Yes. We can use the razor code in javascript in cshtml by using <text> element.

```
< script type="text/javascript">
@foreach (var item in Model) {
< text >
//javascript goes here which uses the server values
< text >
}
< script>
```

112.   HOW CAN I RETURN STRING RESULT FROM ACTION IN ASP.NET MVC?

Below is the code snippet to return string from action method :

```
public ActionResult TestAction() {
return Content("Hello Test !!");
}
```

---

113.   HOW TO RETURN THE JSON FROM ACTION METHOD IN ASP.NET MVC?

Below is the code snippet to return string from action method :

```
public ActionResult TestAction() {
return JSON(new { prop1 = "Test1", prop2 = "Test2" });
}
```

---

114.   GIVE AN EXAMPLE FOR AUTHORIZATION FILTERS IN AN ASP.NET MVC APPLICATION?

1. RequireHttpsAttribute
2. AuthorizeAttribute

---

115.   WHAT IS DATA ANNOTATION VALIDATOR ATTRIBUTES IN MVC?

Data Annotations are nothing but certain validations that we put in our models to validate the input from the user. ASP.NET *MVC* provides a unique feature in which we can validate the models using the *Data Annotation attribute*. Import the following namespace to use *data annotations* in the application.

Example

[Required(ErrorMessage = "Please enter name"), MaxLength(30)]

[Display(Name = "Student Name")]

public string Name { **get**; **set**; }

---

116.   WHAT ARE THE EXCEPTION FILTERS IN MVC?

Exception filter in MVC provides an ability to handle the exceptions for all the controller methods at a single location. This is by creating a class, which inherits from the FilterAttribute and IExceptionFilter interface.

---

117.   WHICH APPROACH PROVIDES BETTER SUPPORT FOR TEST DRIVEN DEVELOPMENT – ASP.NET MVC OR ASP.NET WEBFORMS?

ASP.NET MVC

ASP.NET WEBFORMS

118.   WHAT'S THE USE OF RESPONSE.OUTPUT.WRITE()?

We can write formatted output using Response.Output.Write().

---

119.   IN WHICH EVENT OF PAGE CYCLE IS THE VIEWSTATE AVAILABLE?

After the Init() and before the Page_Load().

---

120.   FROM WHICH BASE CLASS ALL WEB FORMS ARE INHERITED?

Page class.

---

121.   WHICH VALIDATOR CONTROL YOU USE IF YOU NEED TO MAKE SURE THE VALUES IN TWO DIFFERENT CONTROLS MATCHED?

Compare Validator control.

---

122.   WHAT IS VIEWSTATE?

ViewState is used to retain the state of server-side objects between page post backs.

---

123. HOW LONG THE ITEMS IN VIEWSTATE EXISTS?

They exist for the life of the current page.

---

124. WHAT ARE THE DIFFERENT VALIDATORS IN ASP.NET?

1. Required field Validator
2. Range Validator
3. Compare Validator
4. Custom Validator
5. Regular expression Validator
6. Summary Validator

---

125. HOW YOU CAN ADD AN EVENT HANDLER?

Using the Attributes property of server side control.

e.g.

btnSubmit.Attributes.Add("onMouseOver","JavascriptCode();")

---

126. WHICH TYPE OF CACHING WILL BE USED IF WE WANT TO CACHE THE PORTION OF A PAGE INSTEAD OF WHOLE PAGE?

Fragment Caching: It caches the portion of the page generated by the request. For that, we can create user controls with the below code:

<%@ OutputCache Duration="120" VaryByParam="CategoryID;SelectedID"%>

---

127. CAN WE HAVE A WEB APPLICATION RUNNING WITHOUT WEB.CONFIG FILE?

Yes

---

128. CAN WE ADD CODE FILES OF DIFFERENT LANGUAGES IN APP_CODE FOLDER?

No. The code files must be in same language to be kept in App_code folder.

---

129. WHAT IS PROTECTED CONFIGURATION?

It is a feature used to secure connection string information.

---

130. WRITE CODE TO SEND E-MAIL FROM AN ASP.NET APPLICATION?

MailMessage mailMess = new MailMessage ();

mailMess.From = "abc@gmail.com";

mailMess.To = "xyz@gmail.com";

mailMess.Subject = "Test email";

mailMess.Body = "Hi This is a test mail.";

SmtpMail.SmtpServer = "localhost";

SmtpMail.Send (mailMess);

MailMessage and SmtpMail are classes defined System.Web.Mail namespace.

---

131.   HOW CAN WE PREVENT BROWSER FROM CACHING AN ASPX PAGE?

We can SetNoStore on HttpCachePolicy object exposed by the Response object's Cache property:

Response.Cache.SetNoStore ();

Response.Write (DateTime.Now.ToLongTimeString ());

---

132.   WHAT IS THE GOOD PRACTICE TO IMPLEMENT VALIDATIONS IN ASPX PAGE?

Client-side validation is the best way to validate data of a web page. It reduces the network traffic and saves server resources.

---

133.   WHAT ARE THE EVENT HANDLERS THAT WE CAN HAVE IN GLOBAL.ASAX FILE?

Application Events: Application_Start , Application_End, Application_AcquireRequestState, Application_AuthenticateRequest, Application_AuthorizeRequest, Application_BeginRequest, Application_Disposed, Application_EndRequest, Application_Error, Application_PostRequestHandlerExecute, Application_PreRequestHandlerExecute,Application_PreSendRequestContent,

Application_PreSendRequestHeaders, Application_ReleaseRequestState, Application_ResolveRequestCache, Application_UpdateRequestCache

Session Events: Session_Start,Session_End

---

134.   WHICH PROTOCOL IS USED TO CALL A WEB SERVICE?

HTTP Protocol

---

135.   EXPLAIN ROLE BASED SECURITY?

Role Based Security used to implement security based on roles assigned to user groups in the organization.

Then we can allow or deny users based on their role in the organization. Windows defines several built-in groups, including Administrators, Users, and Guests.

<AUTHORIZATION>< authorization >

< allow roles="Domain_Name\Administrators" / >   < !-- Allow Administrators in domain. -- >

< deny users="*" / >   < !-- Deny anyone else. -- >

< /authorization >

---

136.   HOW CAN WE APPLY THEMES TO AN ASP.NET APPLICATION?

We can specify the theme in web.config file. Below is the code example to apply theme:

<configuration>

```
<system.web>

<pages theme="Windows7" />

</system.web>

</configuration>
```

---

137.   WHAT IS REDIRECTPERMANENT IN ASP.NET?

RedirectPermanent Performs a permanent redirection from the requested URL to the specified URL. Once the redirection is done, it also returns 301 Moved Permanently responses.

---

138.   EXPLAIN THE WORKING OF PASSPORT AUTHENTICATION.

First of all it checks passport authentication cookie. If the cookie is not available then the application redirects the user to Passport Sign on page. Passport service authenticates the user details on sign on page and if valid then stores the authenticated cookie on client machine and then redirect the user to requested page

---

139.   WHAT ARE THE ADVANTAGES OF PASSPORT AUTHENTICATION?

All the websites can be accessed using single login credentials. So no need to remember login credentials for each web site.

Users can maintain his/ her information in a single location.

---

140. WHAT ARE THE ASP.NET SECURITY CONTROLS?

- <asp:Login>: Provides a standard login capability that allows the users to enter their credentials
- <asp:LoginName>: Allows you to display the name of the logged-in user
- <asp:LoginStatus>: Displays whether the user is authenticated or not
- <asp:LoginView>: Provides various login views depending on the selected template
- <asp:PasswordRecovery>: email the users their lost password

---

141. HOW DO YOU REGISTER JAVASCRIPT FOR WEBCONTROLS ?

We can register javascript for controls using <CONTROL - name>Attribtues.Add(scriptname,scripttext) method.

---

142. DIFFERENTIATE STRONG TYPING AND WEAK TYPING

In strong typing, the data types of variable are checked at compile time. On the other hand, in case of weak typing the variable data types are checked at runtime. In case of strong typing, there is no chance of compilation error. Scripts use weak typing and hence issues arises at runtime.

---

143. LIST ALL TEMPLATES OF THE REPEATER CONTROL.

- ItemTemplate
- AlternatingItemTemplate
- SeparatorTemplate
- HeaderTemplate
- FooterTemplate

144.   LIST THE MAJOR BUILT-IN OBJECTS IN ASP.NET?

- Application
- Request
- Response
- Server
- Session
- Context
- Trace

145.   WHAT IS THE APPSETTINGS SECTION IN THE WEB.CONFIG FILE?

The appSettings block in web config file sets the user-defined values for the whole application.

For example, in the following code snippet, the specified ConnectionString section is used throughout the project for database connection:

<configuration>

<appSettings>

<add key="ConnectionString" value="server=local; pwd=password; database=default" />

</appSettings>

146.   WHICH NAMESPACES ARE NECESSARY TO CREATE A LOCALIZED APPLICATION?

System.Globalization

System.Resources

---

147. WHAT ARE THE DIFFERENT TYPES OF COOKIES IN ASP.NET?

Session Cookie - Resides on the client machine for a single session until the user does not log out.

Persistent Cookie - Resides on a user's machine for a period specified for its expiry, such as 10 days, one month, and never.

---

148. WHAT IS THE FILE EXTENSION OF WEB SERVICE?

Web services have file extension .asmx.

---

149. IS IT POSSIBLE TO CREATE WEB APPLICATION WITH BOTH WEBFORMS AND MVC?

Yes. We have to include below mvc assembly references in the web forms application to create hybrid application.

System.Web.Mvc

System.Web.Razor

System.ComponentModel.DataAnnotations

---

150. CAN WE HAVE MULTIPLE WEB CONFIG FILES FOR AN ASP.NET APPLICATION?

Yes, one can create different folders and create a config for the specific folder.

151.   WHAT IS THE DIFFERENCE BETWEEN WEB CONFIG AND MACHINE CONFIG?

Web config file is specific to a web application where as machine config is specific to a machine or server. There can be multiple web config files into an application where as we can have only one machine config file on a server.

Web config fields will override machine configurations.

152.   WHAT IS CROSS PAGE POSTING?

When we click submit button on a web page, the page post the data to the same page. The technique in which we post the data to different pages is called Cross Page posting. This can be achieved by setting POSTBACKURL property of the button that causes the postback. Findcontrol method of PreviousPage can be used to get the posted values on the page to which the page has been posted.

ADO.NET

153. WHAT IS OBJECT POOLING?

Object pooling is nothing but a repository of the objects in memory which can be used later. This object pooling reduces the load of object creation when it is needed. Whenever there is a need of object, object pool manager will take the request and serve accordingly.

---

154. WHAT IS CONNECTION POOLING?

Connection pooling consists of database connection so that the connection can be used or reused whenever there is request to the database. This pooling technique enhances the performance of executing the database commands. This pooling definitely reduces our time and effort.

---

155. WHAT IS THE DIFFERENCE BETWEEN DATAREADER AND DATASET?

a) Datareader is FORWARD and READ only
b) Dataset used to UPDATE records.

a) Datareader is CONNECTED architecture
b) Dataset is DISCONNECTED Recordset

a) Datareader contains single table

b) Datareader can contains multiple tables.

a) Datareader Occupies Less Memory
b) Dataset Occupies More memory

Following are the components of ADO.Net Data provider:

- Connection object – Represents connection to the Database

- Command object – Used to execute stored procedure and command on Database

- ExecuteNonQuery – Executes command but doesn't return any value

- ExecuteScalar – Executes and returns single value

- ExecuteReader – Executes and returns result set

- DataReader – Forward and read only recordset

- DataAdapter – This acts as a bridge between database and a dataset.

157.	IS IT POSSIBLE TO LOAD MULTIPLE TABLES IN A DATASET?

Yes, it is possible to load multiple tables in a single dataset.**29. Which provider is used to connect MS Access, Oracle, etc…?**

OLEDB Provider and ODBC Provider are used to connect to MS Access and Oracle. Oracle Data Provider is also used to connect exclusively for oracle database.

158. WHAT ARE DIFFERENT LAYERS OF ADO.NET?

There are three different layers of ADO.Net:

- Presentation Layer
- Business Logic Layer
- Database Access Layer

159. WHAT ARE ALL THE CLASSES THAT ARE AVAILABLE IN SYSTEM.DATA NAMESPACE?

Following are the classes that are available in System.Data Namespace:

- Dataset.
- DataTable.
- DataColumn.
- DataRow.
- DataRelation.
- Constraint.

160. WHAT ARE THE DATA PROVIDERS IN ADO.NET?

Following are the Data Providers used in ADO.Net:.

- MS SQL Server.
- OLEDB.
- ODBC.

161. WHAT IS THE DIFFERENCE BETWEEN EXECUTESCALAR AND EXECUTENONQUERY?

1. ExecuteScalar returns output value where as ExecuteNonQuery does not return any value but the number of rows affected by the query.
2. ExecuteScalar used for fetching a single value and ExecuteNonQuery used to execute Insert and Update statements.

---

162. WHICH METHOD IS USED BY COMMAND CLASS TO EXECUTE SQL STATEMENTS THAT RETURN SINGLE VALUE?

ExecuteScalar

SQL

163. WHAT IS DENORMALIZATION.

DeNormalization is a technique used to access the data from higher to lower normal forms of database. It is also process of introducing redundancy into a table by incorporating data from the related tables.

---

164. WHAT ARE ALL THE DIFFERENT NORMALIZATIONS?

The normal forms can be divided into 5 forms, and they are explained below -.

First Normal Form (1NF):.

This should remove all the duplicate columns from the table. Creation of tables for the related data and identification of unique columns.

Second Normal Form (2NF):.

Meeting all requirements of the first normal form. Placing the subsets of data in separate tables and Creation of relationships between the tables using primary keys.

Third Normal Form (3NF):.

This should meet all requirements of 2NF. Removing the columns which are not dependent on primary key constraints.

Fourth Normal Form (4NF):.

Meeting all the requirements of third normal form and it should not have multi-valued dependencies.

165.   WHAT IS A RELATIONSHIP AND WHAT ARE THEY?

Database Relationship is defined as the connection between the tables in a database. There are various data basing relationships, and they are as follows:.

- One to One Relationship.
- One to Many Relationship.
- Many to One Relationship.
- Self-Referencing Relationship.

166.   WHAT ARE THE TYPES OF SUBQUERY?

There are two types of subquery – Correlated and Non-Correlated.

A correlated subquery cannot be considered as independent query, but it can refer the column in a table listed in the FROM the list of the main query.

A Non-Correlated sub query can be considered as independent query and the output of subquery are substituted in the main query.

167.   WHAT ARE LOCAL AND GLOBAL VARIABLES AND THEIR DIFFERENCES?

Local variables are the variables which can be used or exist inside the function. They are not known to the other functions and those variables cannot be referred or used. Variables can be created whenever that function is called.

Global variables are the variables which can be used or exist throughout the program. Same variable declared in global cannot be used in functions. Global variables cannot be created whenever that function is called.

### 168.	WHAT IS DATA INTEGRITY?

Data Integrity defines the accuracy and consistency of data stored in a database. It can also define integrity constraints to enforce business rules on the data when it is entered into the application or database.

### 169.	WHAT IS DATAWAREHOUSE?

Datawarehouse is a central repository of data from multiple sources of information. Those data are consolidated, transformed and made available for the mining and online processing. Warehouse data have a subset of data called Data Marts.

### 170.	WHAT IS CROSS-JOIN?

Cross join defines as Cartesian product where number of rows in the first table multiplied by number of rows in the second table. If suppose, WHERE clause is used in cross join then the query will work like an INNER JOIN.

### 171.	WHAT IS COLLATION?

Collation is defined as set of rules that determine how character data can be sorted and compared. This can be used to compare A and, other language characters and also depends on the width of the characters.

ASCII value can be used to compare these character data.

172.   WHAT ARE ALL DIFFERENT TYPES OF COLLATION SENSITIVITY?

Following are different types of collation sensitivity -.

- Case Sensitivity – A and a and B and b.
- Accent Sensitivity.
- Kana Sensitivity – Japanese Kana characters.
- Width Sensitivity – Single byte character and double byte character.

173.   ADVANTAGES AND DISADVANTAGES OF STORED PROCEDURE?

Stored procedure can be used as a modular programming – means create once, store and call for several times whenever required. This supports faster execution instead of executing multiple queries. This reduces network traffic and provides better security to the data.

Disadvantage is that it can be executed only in the Database and utilizes more memory in the database server.

174.   WHAT IS ONLINE TRANSACTION PROCESSING (OLTP)?

Online Transaction Processing (OLTP) manages transaction based applications which can be used for data entry, data retrieval and data processing. OLTP makes data management simple and efficient. Unlike OLAP systems goal of OLTP systems is serving real-time transactions.

Example – Bank Transactions on a daily basis.

175. WHAT IS CLAUSE?

SQL clause is defined to limit the result set by providing condition to the query. This usually filters some rows from the whole set of records.

Example – Query that has WHERE condition

Query that has HAVING condition.

---

176. WHAT IS RECURSIVE STORED PROCEDURE?

A stored procedure which calls by itself until it reaches some boundary condition. This recursive function or procedure helps programmers to use the same set of code any number of times.

---

177. WHAT IS UNION, MINUS AND INTERACT COMMANDS?

UNION operator is used to combine the results of two tables, and it eliminates duplicate rows from the tables.

MINUS operator is used to return rows from the first query but not from the second query. Matching records of first and second query and other rows from the first query will be displayed as a result set.

INTERSECT operator is used to return rows returned by both the queries.

---

178. WHAT IS AN ALIAS COMMAND?

ALIAS name can be given to a table or column. This alias name can be referred in WHERE clause to identify the table or column.

Example-.

Select st.StudentID, Ex.Result from student st, Exam as Ex where st.studentID = Ex. StudentID

Here, st refers to alias name for student table and Ex refers to alias name for exam table.

---

179. HOW CAN YOU CREATE AN EMPTY TABLE FROM AN EXISTING TABLE?

Example will be -.

Select * into studentcopy from student where 1=2

Here, we are copying student table to another table with the same structure with no rows copied.

---

180. HOW TO FETCH COMMON RECORDS FROM TWO TABLES?

Common records result set can be achieved by -.

Select studentID from student INTERSECT Select StudentID from Exam

---

181. HOW TO FETCH ALTERNATE RECORDS FROM A TABLE?

Records can be fetched for both Odd and Even row numbers -.

To display even numbers-.

Select studentId from (Select rowno, studentId from student) where mod(rowno,2)=0

To display odd numbers-.

Select studentId from (Select rowno, studentId from student) where mod(rowno,2)=1

from (Select rowno, studentId from student) where mod(rowno,2)=1.[/sql]

---

182.   HOW TO SELECT UNIQUE RECORDS FROM A TABLE?

Select unique records from a table by using DISTINCT keyword.

Select DISTINCT StudentID, StudentName from Student.

---

183.   WHAT IS THE COMMAND USED TO FETCH FIRST 5 CHARACTERS OF THE STRING?

There are many ways to fetch first 5 characters of the string -.

Select SUBSTRING(StudentName,1,5) as studentname from student

Select LEFT(Studentname,5) as studentname from student

---

184.   WHAT ARE ALL TYPES OF USER DEFINED FUNCTIONS?

Three types of user defined functions are.

- Scalar Functions.
- Inline Table valued functions.
- Multi statement valued functions.

---

185.   WHAT ARE AGGREGATE AND SCALAR FUNCTIONS?

Aggregate functions are used to evaluate mathematical calculation and return single values. This can be calculated from the columns in a table. Scalar functions return a single value based on the input value.

Example -.

Aggregate – max(), count - Calculated with respect to numeric.

Scalar – UCASE(), NOW() – Calculated with respect to strings.

---

186. WHICH OPERATOR IS USED IN QUERY FOR PATTERN MATCHING?

LIKE operator is used for pattern matching, and it can be used as -.

% - Matches zero or more characters.

_(Underscore) – Matching exactly one character.

Example -.

```
Select * from Student where studentname like 'a%'
Select * from Student where studentname like 'ami_'
```

---

187. DEFINE MAGIC TABLES IN SQL SERVER?

A Table which is automatically created and managed by SQL server internally to store the inserted, updated values for any DML (SELECT, DELETE, UPDATE, etc.) operation, is called as Magic tables in SQL server. The triggers preferably use it.

DESIGN PATTERNS

188.   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.

---

189.   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.

---

190.   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

---

191.   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.

---

192.   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

---

193.   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.

194.   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.

195.   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.

---

196.   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.

WEB API / WEB SERVICE / WCF

197.   IS IT RIGHT THAT ASP.NET WEB API HAS REPLACED WCF?

It's a not at all true that ASP.NET Web API has replaced WCF. In fact, it is another way of building non-SOAP based services, i.e., plain XML or JSON string.

---

198.   WEB API SUPPORTS WHICH PROTOCOL?

Web App supports HTTP protocol.

---

199.   WHICH .NET FRAMEWORK SUPPORTS WEB API?

NET 4.0 and above version supports web API.

---

200.   WEB API USES WHICH OF THE FOLLOWING OPEN-SOURCE LIBRARY FOR JSON SERIALIZATION?

Web API uses Json.NET library for JSON serialization.

---

201. BY DEFAULT, WEB API SENDS HTTP RESPONSE WITH WHICH OF THE FOLLOWING STATUS CODE FOR ALL UNCAUGHT EXCEPTION?

500 - Internal Server Error

---

202. WHAT IS SOAP?

SOAP is an XML message format used in web service interactions. It allows to send messages over HTTP or JMS, but other transport protocols can be used. It is also an XML-based messaging protocol for exchanging information among computers.

---

203. WHAT IS THE BENEFIT OF USING REST IN WEB API?

REST is used to make fewer data transfers between client and server which make it an ideal for using it in mobile apps. Web API also supports HTTP protocol. Therefore, it reintroduces the traditional way of the HTTP verbs for communication.

---

204. HOW CAN WE USE WEB API WITH ASP.NET WEB FORM?

Web API can be used with ASP.NET Web Form

It can be performed in three simple steps:

1. Create a Web API Controller,
2. Add a routing table to Application_Start method of Global.sax
3. Then you need to make a jQuery AJAX Call to Web API method and get data.

205. HOW YOU CAN RETURN VIEW FROM ASP.NET WEB API METHOD?

No, we can't return a view from ASP.NET Web API Method. Web API creates HTTP services that render raw data. However, it's also possible in ASP.NET MVC application.

206. HOW TO REGISTER EXCEPTION FILTER GLOBALLY?

It is possible to register exception filter globally using following code-

GlobalConfiguration.Configuration.Filters.Add(new

MyTestCustomerStore.NotImplExceptionFilterAttribute());

207. HOW CAN YOU RESTRICT ACCESS METHODS TO SPECIFIC HTTP VERBS IN WEB API?

With the help of Attributes (like HTTP verbs), It is possible to implement access restrictions in Web API.

It is possible to define HTTP verbs as an attribute to restrict access. Example:

```
[HttpPost]


public void Method1(Class obj)


{


//logic
```

208.   WHO CAN CONSUME WEBAPI?

WebAPI can be consumed by any client which supports HTTP verbs such as GET, PUT, DELETE, POST. As WebAPI services don't need any configuration, they are very easy to consume by any client. Infract, even portable devices like Mobile devices can easily consume WebAPI which is certainly the biggest advantages of this technology.

209.   WHAT ARE WEB SERVICES?

Web services are open standard (XML, SOAP, HTTP etc.) based Web applications that interact with other web applications for the purpose of exchanging data. Web Services can convert your existing applications into Web-applications.

210.   WHAT ARE THE FEATURES OF WEB SERVICES?

Following are the features of Web service –

- It is available over the Internet or private (intranet) networks.

- It uses a standardized XML messaging system.

- It is not tied to any one operating system or programming language.

- It is self-describing via a common XML grammar.

- It is discoverable via a simple find mechanism.

211.   WHAT THE COMPONENTS OF A WEB SERVICE?

The basic web services platform is XML + HTTP. All the standard web services work using the following components –

- SOAP (Simple Object Access Protocol)

- UDDI (Universal Description, Discovery and Integration)

- WSDL (Web Services Description Language)

---

212. HOW DOES A WEB SERVICE WORK?

A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP.

You can also use C# to build new web services on Windows that can be invoked from your web application that is based on JavaServer Pages (JSP) and runs on Linux.

---

213. WHAT IS THE PURPOSE OF XML IN A WEB SERVICE?

A web services takes the help of XML to tag the data, format the data.

---

214. WHAT ARE THE BENEFITS OF WEB SERVICES?

Following are the benefits of using web services –

- Exposing the Existing Function on the network − Web services allows you to expose the functionality of your existing code over the network. Once it is exposed on the network, other application can use the functionality of your program.
- Interoperability − Web services allow various applications to talk to each other and share data and services among themselves.

- Standardized Protocol − Web services use standardized industry standard protocol for the communication. All the four layers (Service Transport, XML Messaging, Service Description, and Service Discovery layers) use well-defined protocols in the web services protocol stack.
- Low Cost of Communication − Web services use SOAP over HTTP protocol, so you can use your existing low-cost internet for implementing web services.

215.   WHAT DO YOU MEAN BY INTEROPERABILITY OF WEB SERVICES?

Web services allow various applications to talk to each other and share data and services among themselves. Other applications can also use the web services. For example, a VB or .NET application can talk to Java web services and vice versa. Web services are used to make the application platform and technology independent.

216.   WHAT DO YOU MEAN BY LOOSELY COUPLED ARCHITECTURE OF WEB SERVICES?

A consumer of a web service is not tied to that web service directly. The web service interface can change over time without compromising the client's ability to interact with the service. A tightly coupled system implies that the client and server logic are closely tied to one another, implying that if one interface changes, the other must be updated. Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems.

217.   WHAT IS HTTP?

HTTP stands for Hyper Text Transfer Protocol.Currently, HTTP is the most popular option for service transport. HTTP is simple, stable, and widely deployed. Furthermore, most firewalls allow HTTP traffic. This allows XML-RPC or SOAP messages to masquerade as HTTP messages.

218.   WHAT IS WSDL?

WSDL is an XML-based language for describing web services and how to access them. WSDL stands for Web Services Description Language.

219.   WHAT IS UDDI?

UDDI is an XML-based standard for describing, publishing, and finding web services. UDDI stands for Universal Description, Discovery, and Integration.

220.   EXPLAIN WHAT IS WCF?

WCF (Windows Communication Framework) is Microsoft framework to make inter-process communication easier. Through various means, it lets you do the communication like MS messaging Queuing, Services, Remoting and so on. It also allows you talk with other.NET apps, or non-Microsoft technologies (like J2EE).

221.   MENTION WHAT ARE THE MAIN COMPONENTS OF WCF?

Main components of WCF are

- Service: The working logic
- Host: The path where the data is saved. E.g., .exe, process, windows service
- Endpoints: The way the service is exposed to the outside world

222.   EXPLAIN HOW DOES WCF WORKS?

WCF follows the "Software as a Service" model, where all units of functionality are defined as services. For communication, each point is a portal or connection either with the client or other services. It is a program that exposes a collection of endpoints.

223.   EXPLAIN WHAT IS THE DIFFERENCE BETWEEN ASMX WEB SERVICES AND WCF?

The difference between WCF and ASMX or ASP.net web service is that ASMX is designed to send and receive messages using SOAP over HTTP only. While the WCF can exchange messages using any format over any transport protocol.

224.   WHAT ARE THE TRANSPORT SCHEMAS DOES WCF SUPPORTS?

It supports

- HTTP
- TCP
- Peer network
- IPC (Inter Process Communication)
- MSMQ

225.   MENTION WHAT ARE THE WAYS OF HOSTING A WCF SERVICE?

The ways of hosting a WCF service are

- IIS

- Self-Hosting
- WAS (Windows Activation Service)

---

226. MENTION THE ADDRESS SYNTAX AND THE DIFFERENT FORMATS OF WCF TRANSPORT SCHEME?

Address syntax of WCF transport scheme is

[transport]:// [machine or domain] [: optional port] format

---

227. IN WCF WHAT ARE DUPLEX CONTRACTS?

Duplex messaging or call-back is used in WCF to communicate with the client. Over different transport system Duplex messaging in WCF is done like TCP, Named pipe and even HTTP. Collectively this is known as duplex contracts in WCF.

---

228. MENTION WHAT ARE THE DIFFERENT INSTANCE MODES IN WCF?

To a particular service instance WCF binds an incoming message request, so the available modes are

- Per Call: This instance is created for each call, efficient in terms of memory but need to maintain session
- Per Session: For a complete session of a user instance are created
- Single: One instance is created which is shared among all the users and shared among all. In terms of memory it is least efficient.

229.   EXPLAIN WHAT IS SOA?

SOA (Service Oriented Architectural) is a collection of services that determines how two computing entities will communicate with each other to achieve certain business functionality and also how one entity can work on behalf of another entity.

---

230.   WHAT ARE THE TYPES OF DATA CONTRACTS IN WCF?

There are two types of Data Contracts

- Data Contract: Attribute used to define the class
- Data Member: Attribute used to define the properties

---

231.   WHAT ARE THE THREE TYPES OF TRANSACTION MANAGER WCF SUPPORTS?

The types of the transaction manager that WCF supports are

- Light Weight
- WS- Atomic Transaction
- OLE Transaction

---

232.   NAME THE NAMESPACE THAT IS USED TO ACCESS WCF SERVICE?

System.ServiceModel is used to access WCF service

---

233.   WHAT IS SOA STANDS FOR?

.NET CORE

234.   WHAT IS HOST IN ASP.NET CORE?

Host encapsulates all the resources for the app. On startup, ASP.NET Core application creates the host. The Resources which are encapsulated by the host include:

- HTTP Server implementation
- Dependency Injection
- Configuration
- Logging
- Middleware components

235.   DESCRIBE THE GENERIC HOST AND WEB HOST?

The host setup the server, request pipeline and responsible for app startup and lifetime management. There are two hosts:

.NET Generic Host

ASP.NET Core Web Host

.NET Generic Host is recommended and ASP.NET Core template builds a .NET Generic Host on app startup.
ASP.NET Core Web host is only used for backwards compatibility.

```
// Host creation

public class Program

{

    public static void Main(string[] args)

    {

        CreateWebHostBuilder(args).Build().Run();

    }


    public static IWebHostBuilder
CreateWebHostBuilder(string[] args) =>

        WebHost.CreateDefaultBuilder(args)

            .UseStartup();

}
```

236.   DESCRIBE THE SERVERS IN ASP.NET CORE?

Server is required to run any application. **ASP.NET Core** provides an in-process HTTP server implementation to run the app. This server implementation listen for HTTP requests and surface them to the application as a set of request features composed into an HttpContext.

ASP.NET Core use the **Kestrel** web server by default. ASP.NET Core comes with:

- Default Kestrel web server that's cross platform HTTP server implementation.
- IIS HTTP Server that's in-process server for IIS.
- HTTP.sys server that's a Windows-only HTTP server and it's based on the HTTP.sys kernel driver and HTTP Server API.

237. HOW CONFIGURATION WORKS IN ASP.NET CORE?

In ASP.NET Core, **Configuration** is implemented using various configuration providers. Configuration data is present in the form of key value pairs that can be read by configuration providers as key value from different configuration sources as below.

- appsettings.json - settings file
- Azure Key Vault
- Environment variables
- In-memory .Net objects
- Command Line Arguments
- Custom Providers

By default apps are configured to read the configuration data from appsettings.json, environment variables, command line arguments etc. While reading the data, values from environment variables override appsettings.json data values. 'CreateDefaultBuilder' method provide default configuration.

---

238. HOW TO READ VALUES FROM APPSETTINGS.JSON FILE?

You can read values from appsettings.json using below code.

```
class Test{
// requires using Microsoft.Extensions.Configuration;
 private readonly IConfiguration Configuration;
    public TestModel(IConfiguration configuration)
    {
        Configuration = configuration;
    }
// public void ReadValues(){
```

```
var val = Configuration["key"]; // reading direct key values

var name = Configuration["Employee:Name"]; // read complex
values

}

}
```

Default configuration provider first load the values from appsettings.json and
then from appsettings.Environment.json file.
Environment specific values override the values from appsettings.json file. In
development environment appsettings.Development.json file values override the
appsettings.json file values, same apply to production environment.

239.   WHAT IS THE OPTIONS PATTERN IN ASP.NET CORE?

Options Pattern allow you to access related configuration settings in Strongly
typed way using some classes. When you are accessing the configuration settings
with the isolated classes, The app should adhere these two principles.

- **Interface Segregation Principle (ISP) or Encapsulation:** The class the
  depend on the configurations, should depend only on the configuration
  settings that they use.
- **Separation of Concerns:** Settings for different classes should not be related
  or dependent on one another.

240.   HOW TO USE MULTIPLE ENVIRONMENTS IN ASP.NET CORE?

ASP.NET Core use environment variables to configure application behavior based
on runtime environment. launchSettings.json file
sets ASPNETCORE_ENVIRONMENT to Development on local Machine.

241. HOW ASP.NET CORE SERVE STATIC FILES?

In ASP.NET Core, **Static files** such as CSS, images, JavaScript files, HTML are the served directly to the clients. ASP.NET Core template provides a root folder called wwwroot which contains all these static files. UseStaticFiles() method inside Startup.Configure enables the static files to be served to client. You can serve files outside of this webroot folder by configuring Static File Middleware as following.

```
app.UseStaticFiles(new StaticFileOptions

    {

        FileProvider = new PhysicalFileProvider(

            Path.Combine(env.ContentRootPath,
"MyStaticFiles")), // MyStaticFiles is new folder

        RequestPath = "/StaticFiles"  // this is requested
path by client

    });
// now you can use your file as below
<img src="/StaticFiles/images/profile.jpg" class="img"
alt="A red rose" />

 // profile.jpg is image inside MyStaticFiles/images folder
```

242. EXPLAIN SESSION AND STATE MANAGEMENT IN ASP.NET CORE?

As we know HTTP is a stateless protocol. HTTP requests are independent and does not retain user values. There are different ways to maintain user state between multiple HTTP requests.

- Cookies
- Session State
- TempData

- Query strings
- Hidden fields
- HttpContext.Items
- Cache

---

243.   CAN ASP.NET APPLICATION BE RUN IN DOCKER CONTAINERS?

Yes, you can run an ASP.NET application or .NET Core application in Docker containers.

---

244.   EXPLAIN THE CACHING OR RESPONSE CACHING IN ASP.NET CORE?

Caching significantly improves the performance of an application by reducing the number of calls to actual data source. It also improves the scalability. Response caching is best suited for data that changes infrequently. Caching makes the copy of data and store it instead of generating data from original source.

Response caching headers control the response caching. `ResponseCache` attribute sets these caching headers with additional properties.

---

245.   WHAT IS IN-MEMORY CACHE?

In-memory cache is the simplest way of caching by ASP.NET Core that stores the data in memory on web server.

Apps running on multiple server should ensure that sessions are sticky if they are using in-memory cache. Sticky Sessions responsible to redirect subsequent client requests to same server. In-memory cache can store any object but distributed

cache only stores byte[].

IMemoryCache interface instance in the constructor enables the In-memory caching service via ASP.NET Core dependency Injection.

---

246. WHAT IS DISTRIBUTED CACHING?

Applications running on multiple servers (Web Farm) should ensure that sessions are sticky. For Non-sticky sessions, cache consistency problems can occur. **Distributed caching** is implemented to avoid cache consistency issues. It offloads the memory to an external process. Distributed caching has certain advantages as below.

- Data is consistent across client requests to multiple server
- Data keeps alive during server restarts and deployments.
- Data does not use local memory

`IDistributedCache` interface instance from any constructor enable distributed caching service via [Dependency Injection](#).

---

247. WHAT IS XSRF OR CSRF? HOW TO PREVENT CROSS-SITE REQUEST FORGERY (XSRF/CSRF) ATTACKS IN ASP.NET CORE?

Cross-Site Request Forgery (XSRF/CSRF) is an attack where attacker that acts as a trusted source send some data to a website and perform some action. An attacker is considered a trusted source because it uses the authenticated cookie information stored in browser.

For example a user visits some site 'www.abc.com' then browser performs authentication successfully and stores the user information in cookie and perform some actions, In between user visits some other malicious site 'www.bad-user.com' and this site contains some code to make a request to vulnerable site (www.abc.com). It's called cross site part of CSRF.

How to prevent CSRF?

- In ASP.NET Core 2.0 or later FormTaghelper automatically inject the antiforgery tokens into HTML form element.

- You can add manually antiforgery token in HTML forms by using `@Html.AntiForgeryToken()` and then you can validate it in controller by `ValidateAntiForgeryToken()` method.
- For more you can visit [Prevent Cross-Site Request Forgery (XSRF/CSRF)](#)

248.   EXPLAIN SESSION AND STATE MANAGEMENT IN ASP.NET CORE.

As we know HTTP is a stateless protocol. HTTP requests are independent and does not retain user values. There are different ways to maintain user state between multiple HTTP requests.

- Cookies
- Session State
- TempData
- Query strings
- Hidden fields

249.   HOW TO ACCESS HTTPCONTEXT IN ASP.NET CORE?

ASP.NET Core apps access HttpContext through the IHttpContextAccessor interface.

250.   EXPLAIN THE CACHING AND RESPONSE CACHING IN ASP.NET CORE.

Caching significantly improves the performance of an application by reducing the number of calls to actual data source. It also improves the scalability.

Three types of caching

- IN-MEMORY CACHING – Good for single server
- DISTRIBUTED CACHE – Good for web farm
- RESPONSECACHE attribute

Response caching is best suited for data that changes infrequently.

Response caching headers control the response caching. ResponseCache attribute sets these caching headers with additional properties.

```
[ResponseCache(Duration = 60)]
 public class HomeController : Controller
{
    public IActionResult Index()
    {
      return View(new HomeOutputModel
      {
        LastUpdated = DateTime.Now
      });
    }
}
```

THE END…
BEST OF LUCK FOR INTERVIEWS