

TOP

500

.NET

INTERVIEW QUESTIONS

2021

C# OOPS ASP.NET MVC WEB API SQL

Q&A ASKED IN TOP TECH COMPANIES...

**MICROSOFT TCS ACCENTURE
INFOSYS WIPRO HCL IBM
TECHM CTS HP**

ANURAG RAWAT

Rs 249



WHY THIS BOOK?

This book is a collection of TOP .NET INTERVIEW questions asked in TOP 10 TECH companies - Microsoft, TCS, Accenture, Infosys, Wipro, HCL, IBM, Tech Mahindra, CTS, HP.
(based on research)

PREFACE

ABOUT THE AUTHOR

Anurag Rawat have more than 13 years of experience in .Net technologies. He helps candidates in clearing technical interview in tech companies.



<https://www.linkedin.com/in/anuragrawat123/>

ACKNOWLEDGEMENTS

There is no point in reinventing the wheel. So some links has been shared to help candidates to understand the concept better. These websites links are:

<https://www.c-sharpcorner.com/>
<https://www.geeksforgeeks.org/>
<https://www.tutorialsteacher.com/>

ABOUT THE BOOK

Part I – Top 100 interview questions

Part II – 400 more questions

Topic	Part I	Part II
	No. of Questions	
OOPS	7	43
C#	19	52
.NET FRAMEWORK	14	35
ASP.NET MVC	19	67
ASP.NET WEBFORMS	9	40
ADO.NET	9	25
JAVASCRIPT	0	34
SQL	16	43
DESIGN PATTERNS	7	11
WEB API / WEB SERVICE / WCF	0	50
Total	100	400

PART I

TOP 100 INTERVIEW QUESTIONS

INDEX

CHAPTER 1 - OOPS.....	13
1. DIFFERENTIATE BETWEEN AN ABSTRACT CLASS AND AN INTERFACE?	13
2. WHAT ARE THE MAIN CONCEPTS OF OOPS?.....	15
3. WHAT ARE THE DIFFERENT TYPES OF INHERITANCE?	17
4. WHAT IS THE DIFFERENCE BETWEEN OVERLOADING AND OVERRIDING?	19
5. DIFFERENTIATE BETWEEN METHOD OVERRIDING AND METHOD HIDING.	20
6. HOW TO PREVENT A CLASS FROM BEING INHERITED? OR WHAT IS SEALED CLASS? ...	21
7. WHAT IS A CONSTRUCTOR AND WHAT ARE ITS TYPES?	22
CHAPTER 2 - C#.....	25
8. HOW TO IMPLEMENT EXCEPTION HANDLING IN C#?	26
9. WHAT IS A FINALLY BLOCK AND GIVE AN EXAMPLE WHEN TO USE IT?	28
10. WHAT IS THE DIFFERENCE BETWEEN “THROW EX” AND “THROW” METHODS?	29

11. WHAT IS ‘THIS’ KEYWORD IN C#?.....	30
12. LIST OUT THE DIFFERENCES BETWEEN ARRAY AND ARRAYLIST?	31
13. WHAT ARE COLLECTIONS IN C# AND WHAT ARE THEIR TYPES?	32
14. WHAT IS THE DIFFERENCE BETWEEN ARRAYLIST AND HASHTABLE?	33
15. WHAT YOU MEAN BY BOXING AND UNBOXING?	34
16. WHAT IS THE DIFFERENCE BETWEEN “STRING” AND “STRINGBUILDER”?.....	35
17. EXPLAIN GENERICS IN C#?	36
18. WHAT YOU MEAN BY DELEGATE?	37
19. WHAT IS THE PURPOSE OF “USING” KEYWORD IN C#?	40
20. WHAT IS THE DIFFERENCE BETWEEN “IS” AND “AS” OPERATORS?	42
21. WHAT IS THE DIFFERENCE BETWEEN “READONLY” AND “CONSTANT” VARIABLES? ...	44
22. WHAT IS THE DIFFERENCE BETWEEN “OUT” AND “REF” PARAMETERS?	45
23. EXPLAIN “STATIC” KEYWORD?	47
24. WHAT ARE THE BASIC STRING OPERATIONS IN C#?	47
25. WHAT IS THE DIFFERENCE BETWEEN IENUMERABLE AND IQUERYABLE IN C#?	49
26. WHAT IS ENUM KEYWORD USED FOR?.....	50

CHAPTER 3 - .NET FRAMEWORK	51
27.WHAT ARE THE IMPORTANT COMPONENTS OF .NET FRAMEWORK?	52
28.WHAT ARE THE DIFFERENT TYPES OF ASSEMBLY IN .NET?	53
29.WHAT IS GARBAGE COLLECTION(GC)?	55
30.WHAT IS THE DIFFERENCE BETWEEN “DISPOSE” AND “FINALIZE”?	56
31.WHAT IS GAC?	58
32.CAN WE FORCE GARBAGE COLLECTOR TO RUN?	58
33.WHAT IS REFLECTION?	58
34.WHAT IS SERIALIZATION? WHAT ARE THE TYPES OF SERIALIZATION?	60
35.WHAT IS MEANT BY GLOBALIZATION AND LOCALIZATION?.....	62
36.EXPLAIN MULTITHREADING?	63
37.WHAT IS THE DIFFERENCE BETWEEN PROCESS AND THREAD?.....	64
38.WHAT IS DEADLOCK?	65
39.WHAT IS THE DIFFERENCE BETWEEN THREADS AND TASKS?	66
40.WHAT IS THE ASYNC AND AWAIT IN TASK?	68

CHAPTER 4 - ASP.NET MVC	72
41. WHAT IS MVC (MODEL VIEW CONTROLLER)?	72
42. WHAT ARE THE ADVANTAGES OF MVC?	74
43. EXPLAIN MVC LIFE CYCLE?	75
44. LIST OUT DIFFERENT RETURN TYPES OF A CONTROLLER ACTION METHOD?	77
45. WHAT ARE ACTION FILTERS IN MVC?	79
46. WHAT IS ROUTING IN MVC?	80
47. WHAT IS THE DIFFERENCE BETWEEN VIEW DATA, VIEW BAG & TEMP DATA?	81
48. WHAT IS PARTIAL VIEW?	83
49. WHAT ARE HTML HELPERS IN MVC?	84
50. EXPLAIN ATTRIBUTE BASED ROUTING IN MVC?	85
51. EXPLAIN AREAS IN MVC?	86
52. EXPLAIN THE CONCEPT OF MVC SCAFFOLDING?	87
53. WHAT IS OUTPUT CACHING IN MVC?	88
54. WHAT IS BUNDLING AND MINIFICATION IN MVC?	91

55.WHAT DIFFERENT APPROACHES TO MODEL ENTITIES IN ENTITY FRAMEWORK ?	92
56.WHAT IS DIFFERENCE BETWEEN MVC AND WEB FORMS?	93
57.HOW CAN WE PASS THE DATA FROM CONTROLLER TO VIEW IN MVC?.....	95
58.WHAT CAN BE DONE TO IMPROVE PERFORMANCE IN ASP.NET MVC?	95
59.WHAT IS THE USE OF ACTION FILTERS IN AN MVC APPLICATION?	96
CHAPTER 5 - ASP.NET WEBFORMS.....	98
60.WHAT IS THE DIFFERENCE BETWEEN SERVER.TRANSFER AND RESPONSE.REDIRECT?. .	98
61.WHERE THE VIEWSTATE IS STORED AFTER THE PAGE POSTBACK?.....	99
62. WHAT IS CACHING?	100
63.WHAT ARE THE DIFFERENT TYPES OF CACHING?	100
64.LIST THE EVENTS IN PAGE LIFE CYCLE.....	102
65.WHAT IS AUTHENTICATION AND AUTHORIZATION IN ASP.NET?	103
66.WHAT ARE THE TYPES OF AUTHENTICATION IN ASP.NET?	103
67.IN WHICH EVENT ARE THE CONTROLS FULLY LOADED?	104
68. HOW WE CAN FORCE ALL THE VALIDATION CONTROLS TO RUN?	105

CHAPTER 6 - ADO.NET	107
69. EXPLAIN ADO.NET ARCHITECTURE?	107
70. WHAT IS CONNECTED ARCHITECTURE AND DISCONNECTED ARCHITECTURE?.....	108
71. WHAT IS THE DIFFERENCE BETWEEN DATAREADER AND DATASET?.....	109
72. WHAT ARE ALL COMPONENTS OF ADO.NET DATA PROVIDER?.....	109
73. WHAT ARE THE DIFFERENT EXECUTE METHODS OF ADO.NET?	110
74. WHAT IS LINQ?	112
75. WHAT IS THE DIFFERENCE BETWEEN ENTITY FRAMEWORK AND ADO.NET?	113
76. WHAT IS THE DIFFERENCE BETWEEN DATASET AND DATATABLE?	114
77. WHAT ARE THE AUTHENTICATION TECHNIQUES USED TO CONNECT TO SQL SERVER?	115
CHAPTER 7 - SQL FOR DEVELOPER	117
78. WHAT IS THE DIFFERENCE BETWEEN STORED PROCEDURE AND FUNCTIONS?	117
79. WHAT ARE INDEXES AND WHAT IS THE DIFFERENCE BETWEEN CLUSTERED AND NON-CLUSTERED INDEX?	118
80. WHAT ARE THE TYPES OF JOIN AND EXPLAIN EACH?	120

81. WHAT IS NORMALIZATION?	122
82. WHAT IS A VIEW?	123
83. HOW TO CREATE CLUSTERED AND NON-CLUSTERED INDEX IN A TABLE?	125
84. WHAT IS A CURSOR? WHY TO AVOID THEM?	126
85. WHAT IS THE DIFFERENCE BETWEEN DELETE AND TRUNCATE COMMANDS?	127
86. WHAT IS THE DIFFERENCE BETWEEN SCOPE_IDENTITY AND @@IDENTITY?	128
87. WHAT IS A CONSTRAINT?	128
88. WHAT IS SELF-JOIN?	130
89. EXPLAIN TRIGGERS AND TYPES OF TRIGGERS?	131
90. WHAT IS THE DIFFERENCE BETWEEN HAVING CLAUSE AND WHERE CLAUSE?	132
91. IS THERE ANY DIFFERENCE BETWEEN PRIMARY KEY AND UNIQUE KEY?	133
92. WHAT ARE ACID PROPERTIES.....	134
93. WHAT IS CTE IN SQL?.....	135
CHAPTER 8 - DESIGN PATTERNS	137
94. WHAT ARE DESIGN PATTERNS AND WHAT PROBLEM THEY SOLVE?	137

95. WHAT ALL ARE THE TYPES OF DESIGN PATTERNS?	137
96. WHAT IS SINGLETON DESIGN PATTERN?	139
97. WHAT IS FACTORY DESIGN PATTERN?	141
98. WHAT IS ABSTRACT FACTORY DESIGN PATTERN?	142
99. WHAT ARE SOLID PRINCIPLES?.....	143
100..... EXPLAIN DEPENDENCY INJECTION?	144

Chapter 1 - OOPS

1

DIFFERENTIATE BETWEEN AN ABSTRACT CLASS AND AN INTERFACE?

- a) Abstract class contains both DECLARATION & DEFINITION of methods.
- b) Interface contains ONLY DECLARATION of methods.

- a) Abstract class keyword: ABSTRACT
- b) Interface keyword: INTERFACE

- a) Abstract class does not Support MULTIPLE INHERITANCE
- b) Interface supports multiple inheritance

- a) WHEN TO USE - Abstract class is a good choice when you are sure some methods are concrete/defined and has to be implemented in the same way in some derived classes.
- b) An interface is a good choice when you know a method has to be there in but it can be implemented differently by independent derived classes.

- a) Abstract class can methods, fields, constants, constructor, static members, methods.

b) Interface can contain undefined methods only nothing else.

a) Abstract class example:

```
// abstract class 'G'  
public abstract class G {  
  
    // abstract method 'gfg1()'  
    public abstract void gfg1();  
}
```

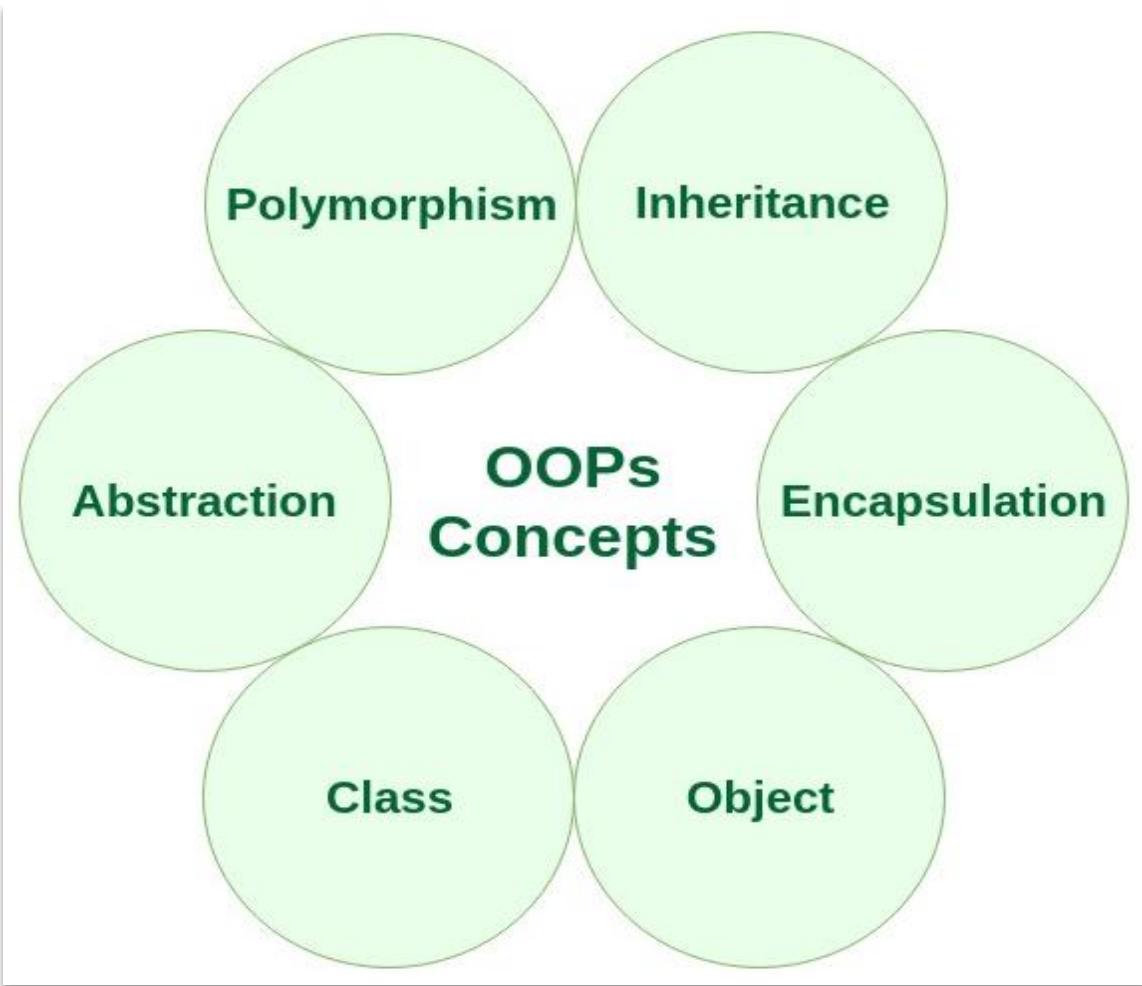
b) Interface example:

```
// A simple interface  
interface interface1 {  
  
    // method having only declaration  
    // not definition  
    void show();  
}
```

[More in Detail](#)

2

WHAT ARE THE MAIN CONCEPTS OF OOPS?



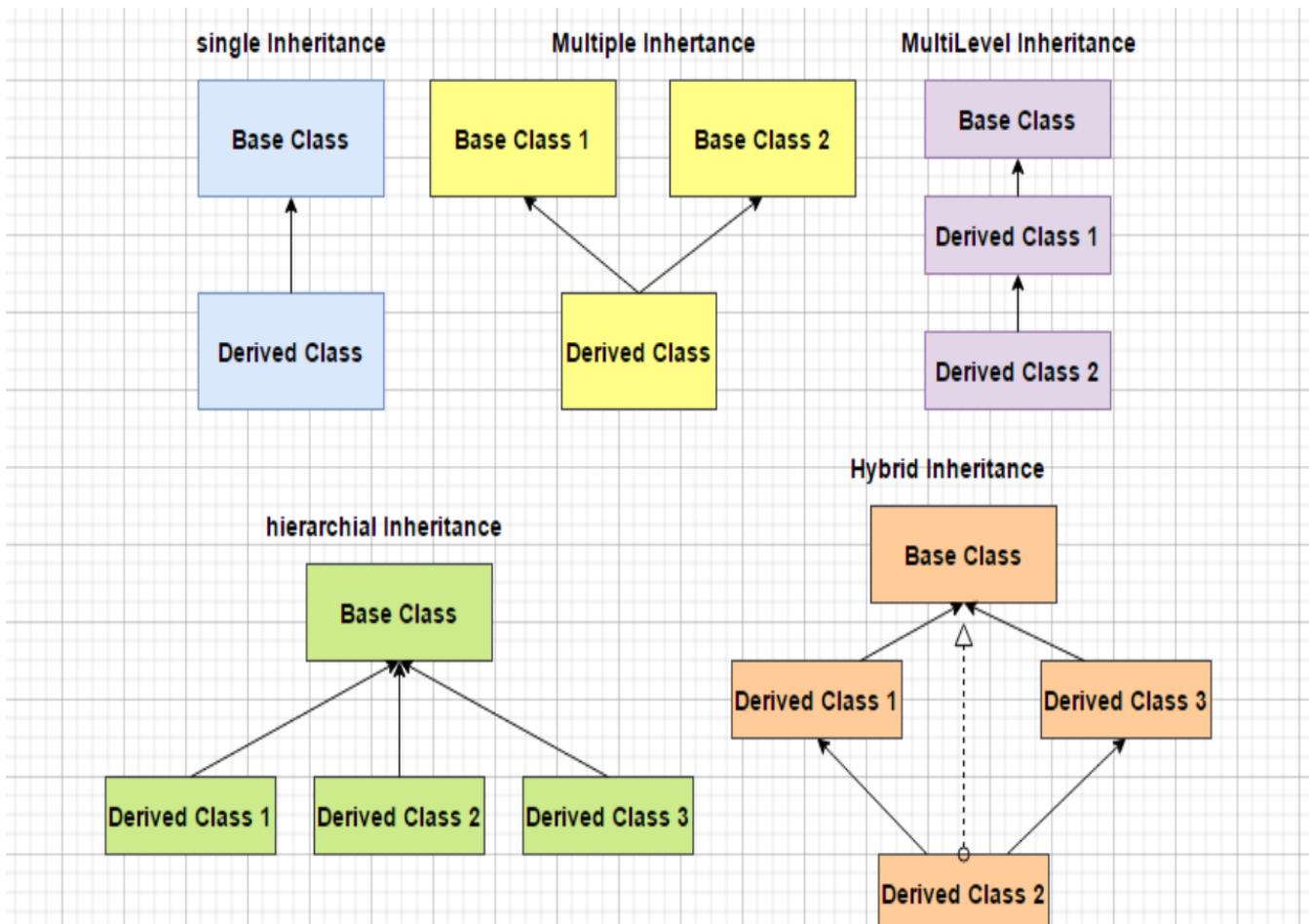
- CLASS - A class is a BLUEPRINT that defines the variables and the methods common to all objects of a certain kind. For example “Expensive Cars” is a class.
- OBJECT - An object can be defined as an INSTANCE of a class which contains both the data and the function. It’s a real world entity for example - Mercedes, BMW, Toyota are part of “Expensive Cars” class.

- INHERITANCE - Inheritance is creating a PARENT-CHILD relationship between two classes where child acquires the properties and behaviors of the parent.
- ENCAPSULATION – encapsulation refers to the WRAPPING of data and methods into a single unit.
- POLYMORPHISM - Polymorphism is the ability of a variable, object or function to take on MULTIPLE FORMS. For example, in English, the verb “run” has a different meaning if you use it with a laptop, a foot race, and business.
- ABSTRACTION - Abstraction is an act of representing essential features without including BACKGROUND details.

[More in Detail](#)

3

WHAT ARE THE DIFFERENT TYPES OF INHERITANCE?



SINGLE INHERITANCE - When a Derived Class inherit properties and behavior from a single Base Class , it is called as single inheritance.

- **MULTI LEVEL INHERITANCE** - A derived class is created from another derived class is called Multi Level Inheritance .
- **HIERARCHICAL INHERITANCE** - More than one derived classes are created from a single base class, is called Hierarchical Inheritance.
- **HYBRID INHERITANCE** - Any combination of above three inheritance (single, hierarchical and multi level) is called as hybrid inheritance .
- **MULTIPLE INHERITANCE** - Multiple inheritances allows programmers to create classes that combine aspects of multiple classes and their corresponding hierarchies.

[More in Detail](#)

4

WHAT IS THE DIFFERENCE BETWEEN OVERLOADING AND OVERRIDING?



- a) **OVERLOADING** - Creating more than one method or function having same name but different signatures or the parameters in the same class is called method overloading.

b) OVERRIDING - Creating a method in the derived class with the same name and signature as a method in the base class is called as method overriding. It uses VIRTUAL keyword for base class method and OVERRIDE keyword for derived class method.

a) Overloading is a type of COMPILE TIME polymorphism.

b) Overriding is a type of RUN TIME polymorphism.

a) Method overloading doesn't need inheritance.

b) Method overriding NEEDS INHERITANCE.

a) Method overloading is also called EARLY BINDING.

b) Method overriding is also called LATE BINDING.

[More in Detail](#)

5

DIFFERENTIATE BETWEEN METHOD OVERRIDING AND METHOD HIDING.

a) In method overriding, you need to define the method of a parent class as a virtual method using virtual keyword and

- the method of child class as an overridden method using override keyword.
- b) In method hiding, you just simply create a method in a parent class and in child class you need to define that method using NEW keyword.
- a) Method overriding only redefines the implementation of the method.
- b) In method hiding, you can completely redefine the method.

[More in Detail](#)

6

HOW TO PREVENT A CLASS FROM BEING INHERITED?

-  With SEALED keyword you can prevent a class from inherited.

Sealed classes are used to restrict the users from inheriting the class. A class can be sealed by using the *sealed* keyword. The keyword tells the compiler that the class is sealed, and therefore, cannot be extended. No class can be derived from a sealed class.

7

WHAT IS A CONSTRUCTOR AND WHAT ARE ITS TYPES?

- + A special method of the class that is automatically invoked when an instance of the class is created is called a constructor.
- + The main use of constructors is to initialize the private fields of the class while creating an instance for the class.
- + When you have not created a constructor in the class, the compiler will automatically create a default constructor of the class.



- 1. Default constructor
- 2. Parameterized constructor
- 3. Copy constructor
- 4. Static constructor
- 5. Private constructor

1. **DEFAULT CONSTRUCTOR** - A constructor without any parameters is called a default constructor.
2. **PARAMETERIZED CONSTRUCTOR** - A constructor with at least one parameter is called a parameterized constructor.
3. **COPY CONSTRUCTOR** - The constructor which creates an object by copying variables from another object is called a copy constructor.
4. **STATIC CONSTRUCTOR** - When a constructor is created using a static keyword, it will be invoked only once for all of the instances of the class and it is invoked during the creation of the first instance of the class.
5. **PRIVATE CONSTRUCTOR** - When a constructor is created with a private specifier, it is not possible for other classes to derive from this class, neither is it possible to create an

instance of this class. They are usually used in classes that contain static members only.

[More in Detail](#)



#RESUME TIP 1

INCLUDE EACH AND EVERY TECHNICAL SKILL

Your resume will be shortlisted based on the skills(keywords) recruiters are looking for. So include all technical skills you know.

For example:

Skills - ~~.NET, SQL Server~~ (This is not enough)

Skills - .NET, C#, SQL Server, Javascript, Jquery, HTML5(as much as you know)

Chapter 2 - C#

8

HOW TO IMPLEMENT EXCEPTION HANDLING IN C#?

- ✚ Exception handling in Object-Oriented Programming is a very important concept that is used to manage errors.

An exception handler allows errors to be thrown and caught and implements a centralized mechanism to resolve them.

C# exception handling is built upon four keywords: try, catch, finally, and throw.

- ✚ TRY – A try block identifies a block of code for which particular exceptions is activated. It is followed by one or more catch blocks.
- ✚ CATCH – A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.
- ✚ FINALLY – The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown.

For example, if you open a file, it must be closed whether an exception is raised or not.

- THROW – A program throws an exception when a problem shows up. This is done using a throw keyword.

Syntax:

```
try
{
    // statements causing exception
}
catch(ExceptionName e1 ) {
    // error handling code
}
catch (ExceptionName e2)
{
    // error handling code
}
finally
{
    // statements to be executed
}
```

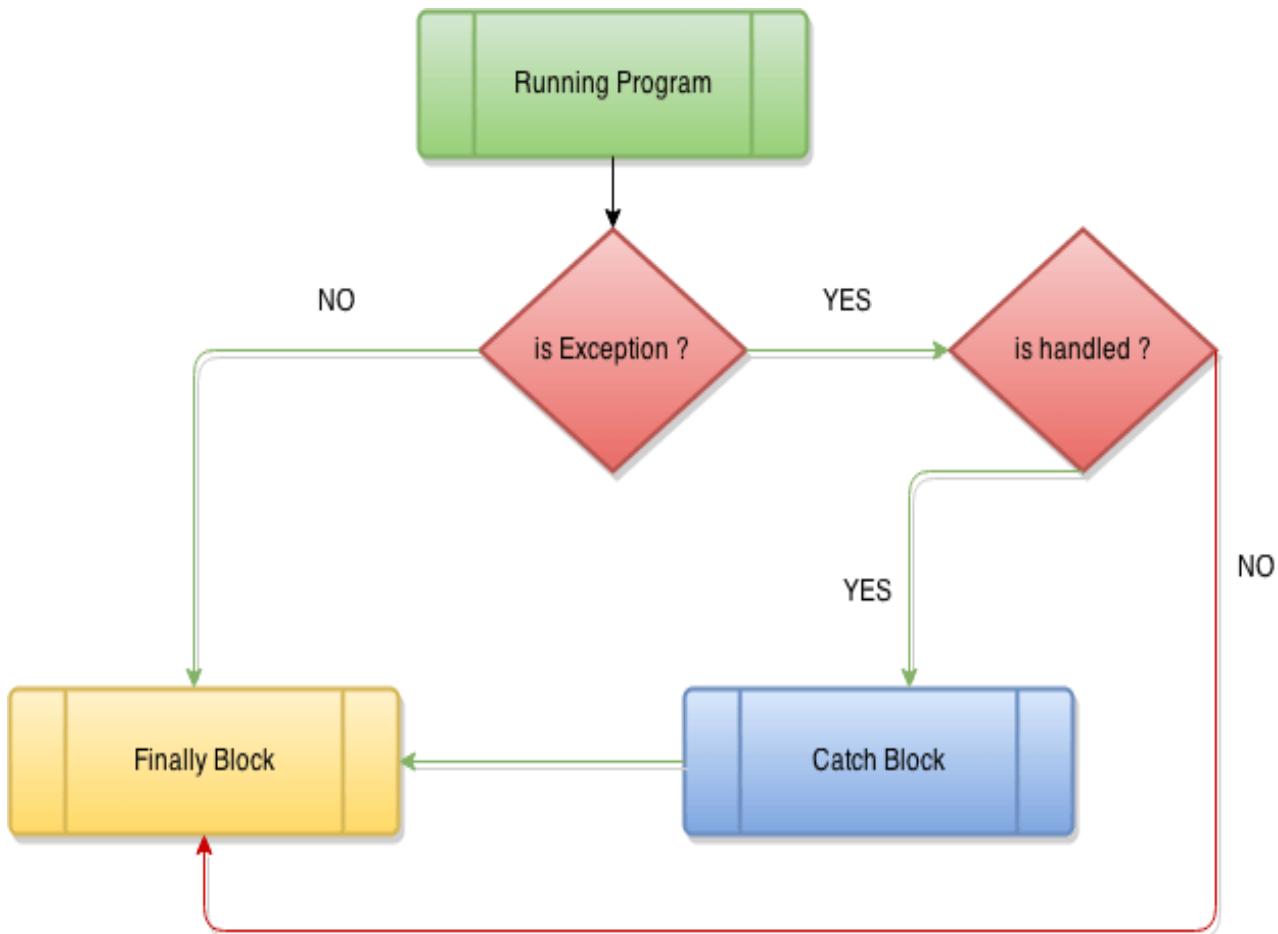
9

WHAT IS A FINALLY BLOCK AND GIVE AN EXAMPLE WHEN TO USE IT?

FINALLY BLOCK WILL BE EXECUTED IRRESPECTIVE OF EXCEPTION.

So while executing the code in try block when exception is occurred, control is returned to catch block and at last finally block will be executed.

For example, closing connection to database can be kept in



“finally” block as it has to happen irrespective of the result.

10

WHAT IS THE DIFFERENCE BETWEEN “THROW EX” AND “THROW” METHODS?

- a) “throw ex” will replace the stack trace of the exception with stack trace info of re throw point.
- b) “throw” will preserve the original stack trace info. Using throw one will able to view the actual stack trace info.

 THROW is always better than throw ex as it preserves original stack trace.

WHAT IS ‘THIS’ KEYWORD IN C#?

 *this* keyword is used to refer to the CURRENT INSTANCE of the class. It is used to access members from the constructors, instance methods, and instance accessors.

Here, for an example, we are showing a record of Students i.e: id, Name, Age, and Subject. To refer to the fields of the current class, we have used the “this” keyword in C# –

```
class Student {  
    public int id;  
    public String name;  
  
    public Student(int id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
}
```

[More in Detail](#)

12

LIST OUT THE DIFFERENCES BETWEEN ARRAY AND ARRAYLIST?

- a) Array is strongly typed. This means that an array can store only specific type of items\elements.
 - b) ArrayList can store any type of items\elements.
-
- a) Array can contain fixed number of items.
 - b) ArrayList can store any number of items.

a) Example Array:

```
int[] intArray = new int[] { 2 };  
intArray[0] = 1;  
intArray[2] = 2;
```

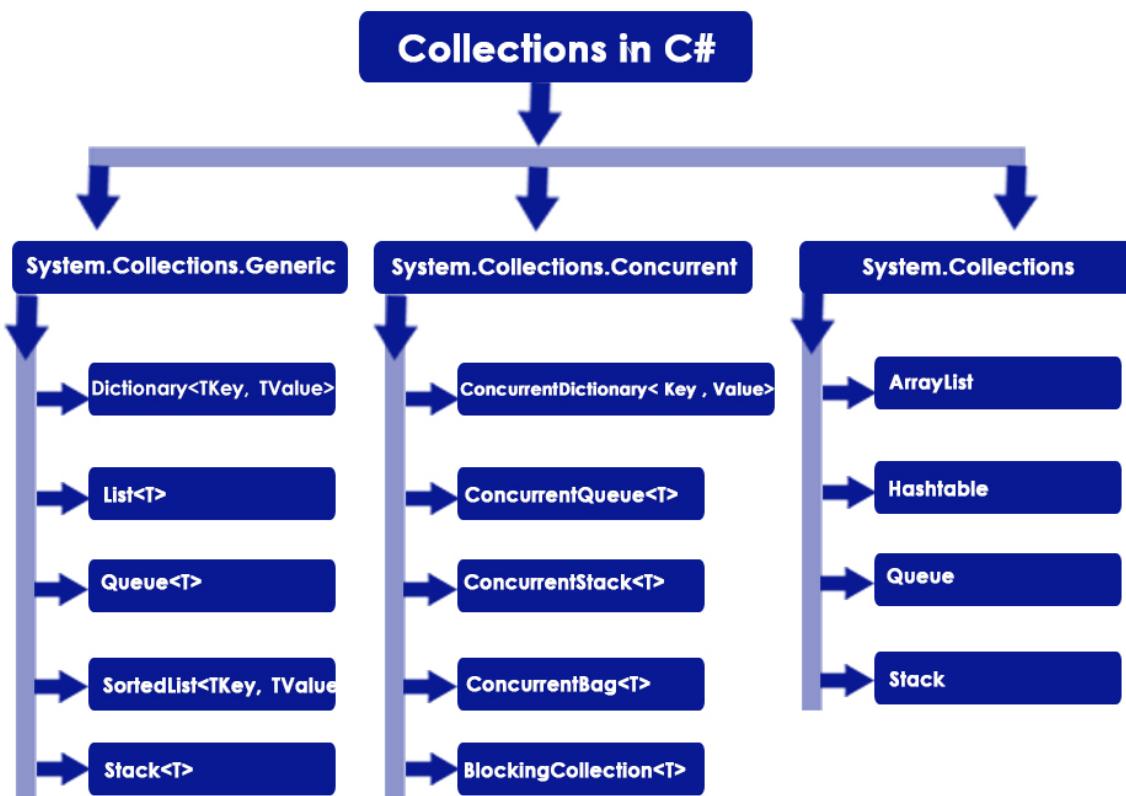
b) Example ArrayList:

```
ArrayList Arrlst = new ArrayList();  
Arrlst.Add("Sagar");  
Arrlst.Add(1);  
Arrlst.Add(null);
```

13

WHAT ARE COLLECTIONS IN C# AND WHAT ARE THEIR TYPES?

C# collection types are designed to store, manage and manipulate similar data more efficiently. Data manipulation includes adding, removing, finding, and inserting data in the collection.



In non-generic collections, each element can represent a value of a different type. The collection size is not fixed.

Items from the collection can be added or removed at runtime.

- ▀ Generic Collections work on the specific type that is specified in the program whereas non-generic collections work on the object type.

[More in Detail](#)

14

WHAT IS THE DIFFERENCE BETWEEN ARRAYLIST AND HASHTABLE?

- a) Array List is a List.
 - b) Hash Table is a map.
 - c) In ArrayList we can only add items to the list.
 - d) In hashtable we can add items with the key.
- a) Example ArrayList:
- ```
ArrayList arrList = new ArrayList();
arrList.Add(7896);
arrList.Add("Seven");
```

b) Exampel Hashtable:

```
Hashtable hashtable = new Hashtable();
hashtable.Add("Number",1);
hashtable.Add("Car", "Ferrari");
```

---

15

## WHAT YOU MEAN BY BOXING AND UNBOXING?

Boxing –

- + This is the process of converting from value type to reference type.
- + Boxing is implicit conversion process in which object type (super type) is used.
- + The Value type is always stored in Stack. The Referenced Type is stored in Heap.

UnBoxing – It's completely opposite to boxing.

- + The process of converting reference type into the value type is known as Unboxing.
- + It is explicit conversion process.

## **Example :**

```
int num = 23; // value type is int
and assigned value 23
```

```
Object Obj = num; // Boxing
```

```
int i = (int)Obj; // Unboxing -
Explicit
```

---

16

## **WHAT IS THE DIFFERENCE BETWEEN “STRING” AND “STRINGBUILDER”?**

- a) String is IMMUTABLE in C# that mean you couldn't modify it after it is created. It creates a new object of string type in memory if you will perform any operation.
- b) StringBuilder is mutable in C#. This means that if an operation is performed on the string, it will not create a new instance every time. With that, it will not create new space in memory, unlike Strings.

---

17

## EXPLAIN GENERICS IN C#?

 C# allows you to define generic classes, interfaces, abstract classes, fields, methods, static methods, properties, events, delegates, and operators using the type parameter and without the specific data type.

Generic classes are defined using a type parameter in angle brackets after the class name. The following defines a generic class.

Example: Define Generic Class

```
class DataStore<T>
{
 public T Data { get; set; }
}
```

Instantiating Generic Class

You can create an instance of generic classes by specifying an actual type in angle brackets. The

following creates an instance of the generic class DataStore.

```
DataStore<string> store = new
DataStore<string>();
```

[More in Detail](#)

---

18

## WHAT YOU MEAN BY DELEGATE?

- + A Delegate is a variable that holds the REFERENCE TO A METHOD.

All Delegates are derived from System.Delegate namespace. Both Delegate and the method that it refers to can have the same signature.

```

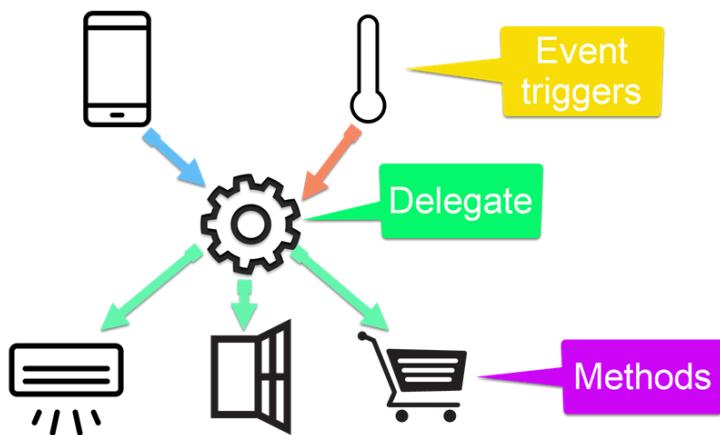
 public delegate void MyDelegate(string msg);
 Access modifier Delegate type
 Delegate function signature

 MyDelegate del = MethodA; Function signature must match with
 delegate signature

 public static void MethodA(string msg){

 }

```



[More in Detail](#)



## #RESUME TIP 2

### SHOW—DON'T TELL—YOUR SOFT SKILLS

In a technical resume, recruiters rarely care about soft skills like “Communication” and all. So if possible replace them with some technical skills.

19

## WHAT IS THE PURPOSE OF “USING” KEYWORD IN C#?

**There are two purpose of using keyword in C#:**

### 1. USING DIRECTIVE

 Generally, we use the using keyword to add namespaces in code-behind and class files.

Example:

**using System.IO;**

**using System.Text;**

### 2. USING STATEMENT

 This is another way to use the using keyword in C#. It plays a vital role in improving performance in GARBAGE COLLECTION.

The using statement ensures that DISPOSE() is called even if an exception occurs when you are creating objects and calling methods, properties and so on.

Dispose() is a method that is present in the IDisposable interface that helps to implement custom Garbage Collection.

Example:

```
using(SqlHelper sqlHelper = new SqlHelper())
```

```
{
```

```
 // use sqlHelper object
```

```
} //automatically calls Dispose method
```

---

20

## WHAT IS THE DIFFERENCE BETWEEN “IS” AND “AS” OPERATORS?

- a) The **IS** operator is USED TO CHECK if the run-time type of an object is compatible with the given type or not.
  - b) **AS** operator is used to PERFORM CONVERSION between compatible reference types or Nullable types.
- 
- a) The **IS** operator is of boolean type.
  - b) whereas **AS** operator is not of boolean type.
- 
- a) **IS** operator returns true if the given object is of the same type.
  - b) Whereas **AS** operator returns the object when they are compatible with the given type.
- 
- a) Is operator example

```
P o1 = new P();
```

```
P1 o2 = new P1();
```

```
Console.WriteLine(o1 is P); //Output: true;
```

- b) As operator example

```
object[] o = new object[1];
o[0] = "Hello";
string str1 = o[0] as string;
Console.WriteLine(str1); //output: Hello
```

[More in Detail](#)

---

## WHAT IS THE DIFFERENCE BETWEEN “READONLY” AND “CONSTANT” VARIABLES?

- a) In readonly fields, we can assign values in declaration and in the CONSTRUCTOR PART.
  - b) In const fields, we can only assign values in DECLARATION PART.
- 
- a) The value of readonly field can be changed.
  - b) The value of the const field can not be changed.
- 
- a) Readonly fields can be created using readonly keyword
  - b) Constant fields are created using const keyword.
- 
- a) ReadOnly is a RUNTIME constant.
  - b) Const is a COMPILE time constant.
- 
- a) Example Constant fields:

```
public const int myvar = 10;
public const string str = "C-Sharp";
```

- b) Example Readonly fields:

```
public readonly int myvar1;
```

```
// Values of the readonly variables are assigned
using constructor
public GFG(int b)
{
 myvar1 = b;
 Console.WriteLine("Display value of myvar1
{0}, ");
}
```

---

22

## WHAT IS THE DIFFERENCE BETWEEN “OUT” AND “REF” PARAMETERS?

- a) REF parameters must be initialized before it pass to another method.
- b) It is not necessary to initialize OUT parameters before it pass to another method.

-  In ref parameter, it is not necessary to initialize the value of a parameter before returning to the calling method.
-  In out parameter, it is necessary to initialize the value of a parameter before returning to the calling method.

- a) When ref keyword is used the data may pass in bi-directional.
- b) When out keyword is used the data only passed in unidirectional.

Example:

```
public class Program
{
 public static void update(out int a, ref int
d)
 {
 a = 10;
 d = 11;
 }
 public static void Main()
 {
 int b; //out parameter not initialized
 int c = 9; //ref parameter initialized
 Program p1 = new Program();
 update(out b, ref c);
 Console.WriteLine("Updated out value is:
{0}", b);
 Console.WriteLine("Updated ref value is:
{0}", c);
 }
}
```

}

---

23

## EXPLAIN “STATIC” KEYWORD?

 A static class is declared with the help of *static* keyword. A static class can only contain static data members, static methods, and a static constructor. It is not allowed to create objects of the static class. Static classes are sealed, means one cannot inherit a static class from another class.

[More in Detail](#)

---

24

## WHAT ARE THE BASIC STRING OPERATIONS IN C#?

Some of the basic string operations are:

- **Concatenate:** Two strings can be concatenated either by using a `System.String.Concat` or by using `+` operator.

```
string str1 = "This is one";
 string str2 = "This is two";
string str2 = str1 + str2;
```

- Modify: Replace(a,b) is used to replace a string with another string. Trim() is used to trim the string at the end or at the beginning.

```
string str1 = "This is one";
string str2 = str1.Replace("one", "two");
```

- Compare: System.StringComparison() is used to compare two strings, either a case-sensitive comparison or not case sensitive. Mainly takes two parameters, original string, and string to be compared with.

```
string str1 = "This is test";
string str2 = "This is text";
```

```
if (String.Compare(str1, str2) == 0) {
 Console.WriteLine(str1 + " and " + str2 + " are equal.");
} else {
 Console.WriteLine(str1 + " and " + str2 + " are not equal.");
```

}

- Contains: Check if a string contains a pattern of substring or not.

```
string str = "This is test";
if (str.Contains("test")) {
 Console.WriteLine("The sequence 'test' was found.");
}
```

---

25

## WHAT IS THE DIFFERENCE BETWEEN IENUMERABLE AND IQUERYABLE?

- IEnumarable exists in System.Collections Namespace.
- IQueryable exists in System. Linq Namespace.
- Both IEnumarable and IQueryable are forward collection.
- IEnumarable doesn't support lazy loading
- IQueryable support lazy loading

- Querying data from a database, `IEnumerable` execute a select query on the server side, load data in-memory on a client-side and then filter data.
- Querying data from a database, `IQueryable` execute the select query on the server side with all filters.

[More in Detail](#)

---

26

## WHAT IS ENUM KEYWORD USED FOR?

 **Enum** keyword is used for declaring an enumeration, which consists of named constants and it is called as enumerator lists. Enums are value types and these can't be inherited. Below is the sample code of using Enums.

Example:

```
enum Fruits { Apple, Orange, Banana, WaterMelon
};
```

---



## #RESUME TIP 3

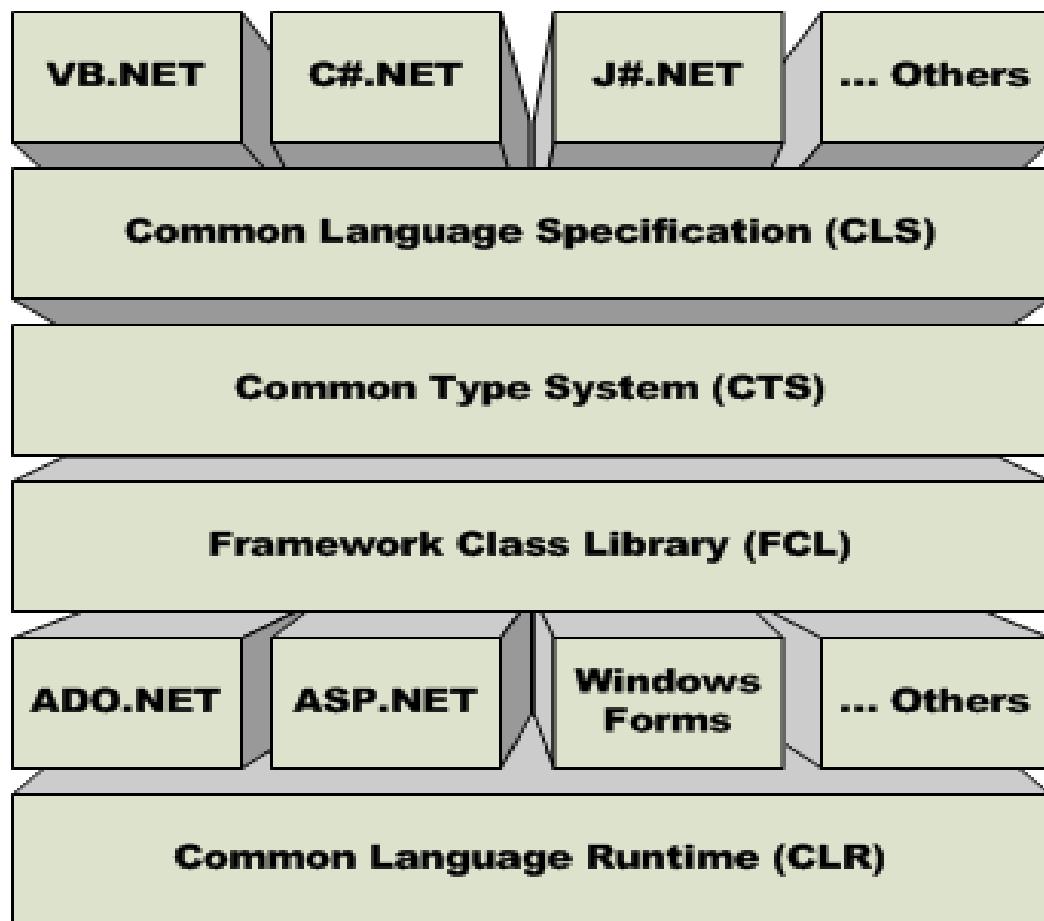
### ONE PAGE RESUME.

Do not create resume of more than one page if you have less than 8 years of experience. Remove short term projects if required. Avoid long project descriptions.

# Chapter 3 -.NET FRAMEWORK

27

## WHAT ARE THE IMPORTANT COMPONENTS OF .NET FRAMEWORK?



- CTS – CTS stands for Common Type System. It has a set of rules which state how a data type should be declared, defined and used in the program.
- CLS – CLS stands for Common Language Specification and it is a subset of CTS. It defines a set of rules and restrictions that every language must follow which runs under .NET framework.
- CLR – Common Language Runtime (CLR) is the programming (Virtual Machine component) that manages the execution of programs written in any language that uses the .NET Framework, for example C#, VB.Net, F# and so on. CLR does various operations like memory management, security checks etc
- FCL or BCL – Framework Class Library is the collection of classes, namespaces, interfaces and value types that are used for .NET applications.

---

28

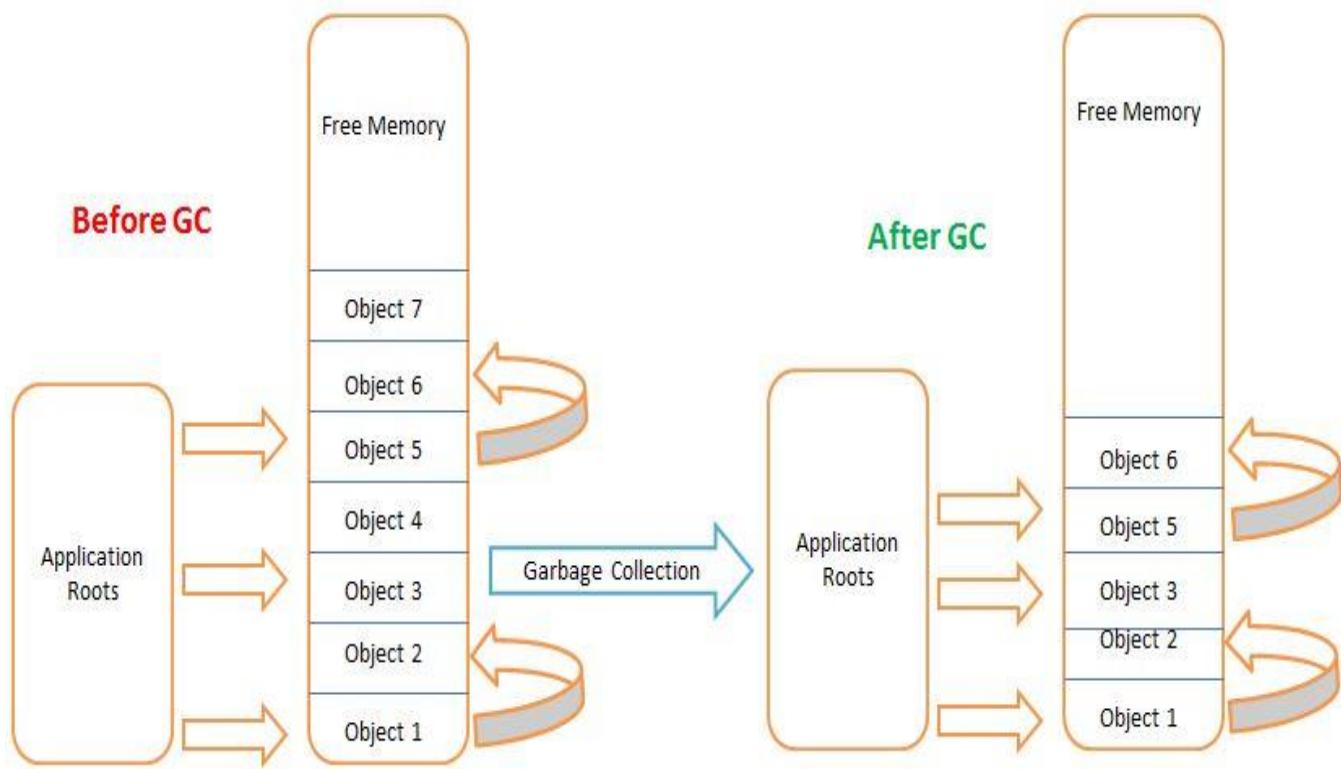
## WHAT ARE THE DIFFERENT TYPES OF ASSEMBLY IN .NET?

There are 3 types of assembly Private and Public assembly.

- **PRIVATE ASSEMBLY** - A private assembly is normally used by a single application, and is stored in the application's directory, or a sub-directory beneath.
  - **SHARED ASSEMBLY** - A shared assembly is normally stored in the global assembly cache, which is a repository of assemblies maintained by the .NET runtime. Shared assemblies are usually libraries of code, which many applications will find useful, e.g. Crystal report classes that will be used by all application for Reports.
  - **SATELLITE ASSEMBLY** - A satellite Assembly is defined as an assembly with resources only, no executable code.
-

## WHAT IS GARBAGE COLLECTION(GC)?

The garbage collector (GC) manages the allocation and release of memory. The garbage collector serves as an AUTOMATIC MEMORY MANAGER.



You do not need to know how to allocate and release memory or manage the lifetime of the objects that use that memory.

[More in Detail](#)

30

## WHAT IS THE DIFFERENCE BETWEEN “DISPOSE” AND “FINALIZE”?

- a) Dispose is used to free unmanaged resources like files, database connections etc. at any time.
  - b) Finalize is used to free unmanaged resources (when you implement it) like files, database connections etc. held by an object before that object is destroyed.
- 
- a) Disposed is CALLED EXPLICITLY, it is called by user code and the class which is implementing dispose method, must has to implement IDisposable interface.
  - b) Finalize is CALLED BY GARBAGE COLLECTOR and cannot be called by user code.
- 
- a) There is no performance costs associated with Dispose method.
  - b) There is performance costs associated with Finalize method since it doesn't clean the memory immediately and called by GC automatically.
- 
- a) Dispose Example:

```
public class Demo : IDisposable {
```

```
private bool disposed = false;
public void Dispose() {
 Dispose(true);
 GC.SuppressFinalize(this);
}

protected virtual void Dispose(bool disposing) {
 if (!disposed) {
 if (disposing) {
 //clean up managed objects
 }
 //clean up unmanaged objects
 disposed = true;
 }
}
```

b) Finalize is automatically called by Garbage collector.

---

31

## WHAT IS GAC?

GAC (Global Assembly Cache) is where all shared .NET assembly reside.

---

32

## CAN WE FORCE GARBAGE COLLECTOR TO RUN?

`System.GC.Collect ()` forces garbage collector to run.

This is not recommended but can be used if situations arise.

---

33

## WHAT IS REFLECTION?

 Reflection is the ability of a code to access the metadata of the assembly during runtime.

A program reflects upon itself and uses the metadata to inform the user or modify its behavior. Metadata refers to information

about objects, methods. The namespace System.Reflection contains methods and classes that manage the information of all the loaded types and methods. It is mainly used for windows applications,

**For Example**, to view the properties of a button in a windows form.

The MemberInfo object of the class reflection is used to discover the attributes associated with a class. Reflection is implemented in two steps, first, we get the type of the object, and then we use the type to identify members such as methods and properties.

**To get type of a class, we can simply use,**

```
Type mytype = myClass.GetType();
```

Once we have a type of class, the other information about the class can be easily accessed.

```
System.Reflection.MemberInfo Info =
mytype.GetMethod("AddNumbers");
```

Above statement tries to find a method with name *AddNumbers* in the class *myClass*.

[More in Detail](#)

---



## #RESUME TIP 4

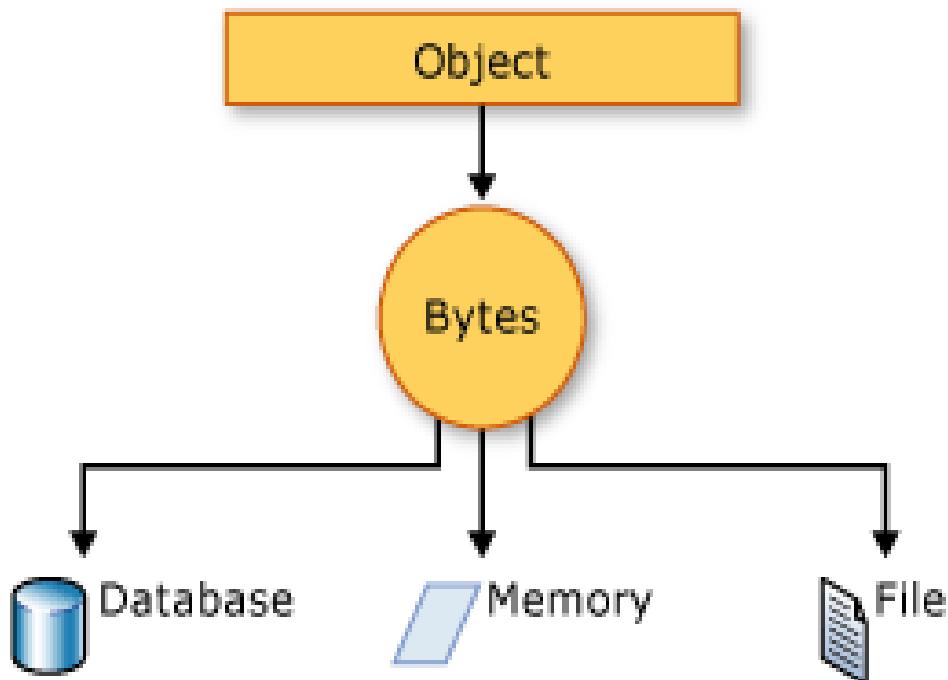
### DEAL WITH THE GAPS.

Justify any gap in education or work. For example fill it with any volunteer work or as a freelancer or with any course.

34

## WHAT IS SERIALIZATION? WHAT ARE THE TYPES OF SERIALIZATION?

Serialization is a process of converting code to its binary format.



Once it is converted to bytes, it can be easily stored and written to a disk or any such storage devices.

Serializations are mainly useful when we do not want to lose the original form of the code and it can be retrieved anytime in the future.

Any class which is marked with the attribute [Serializable] will be converted to its binary form.

The different types of Serialization are:

- + **XML SERIALIZATION** – It serializes all the public properties to the XML document. Since the data is in XML format, it can be easily read and manipulated in various formats. The classes reside in System.xml.Serialization.
- + **SOAP** – Classes reside in System.Runtime.Serialization. Similar to XML but produces a complete SOAP compliant envelope that can be used by any system that understands SOAP.
- + **BINARY SERIALIZATION** – Allows any code to be converted to its binary form. Can serialize and restore public and non-public properties. It is faster and occupies less space.

[More in Detail](#)

---

35

## WHAT IS MEANT BY GLOBALIZATION AND LOCALIZATION?

- + **GLOBALIZATION** is the process of designing and developing web applications that are culture neutral and language neutral. The executable code is separate from the data

specific to the locale. The global resource can be used with any page in an application.

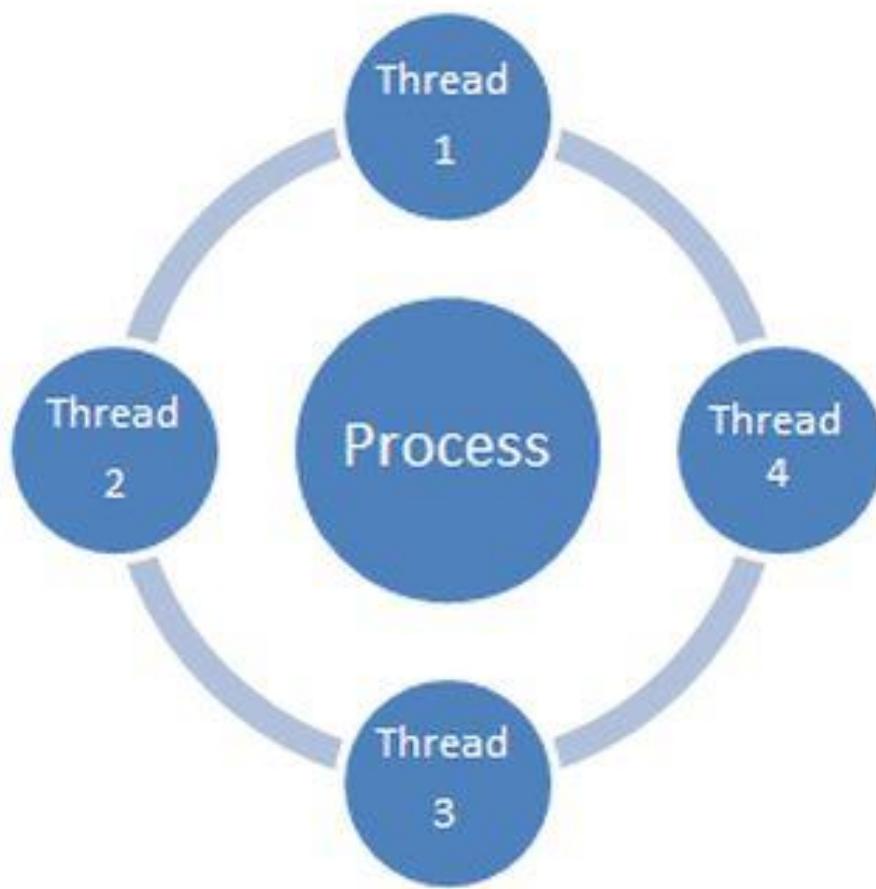
 LOCALIZATION is the process of customizing the globalized web application to a specific locale and culture. Various resources such as images and text for the specific locale are created. The resource file in localization is scoped to a particular page in an application.

---

36

## EXPLAIN MULTITHREADING?

Multithreading in C# is a process in which multiple threads work simultaneously. It is a process to achieve multitasking. It saves time because multiple tasks are being executed at a time. To create multithreaded application in C#, we need to use **System.Threading** namespace.



[More in Detail](#)

---

37

## WHAT IS THE DIFFERENCE BETWEEN PROCESS AND THREAD?

- a) Process means a program is in execution
- b) whereas thread means a segment of a process.

- a) A Process is not Lightweight, whereas Threads are Lightweight.
  - b) A Process takes more time to terminate, and the thread takes less time to terminate.
- 
- a) Process takes more time for creation.
  - b) Whereas Thread takes less time for creation.
- 
- a) A Process is mostly isolated.
  - b) Whereas Threads share memory.
- 
- a) Process does not share data.
  - b) Threads share data with each other.

[More in Detail](#)

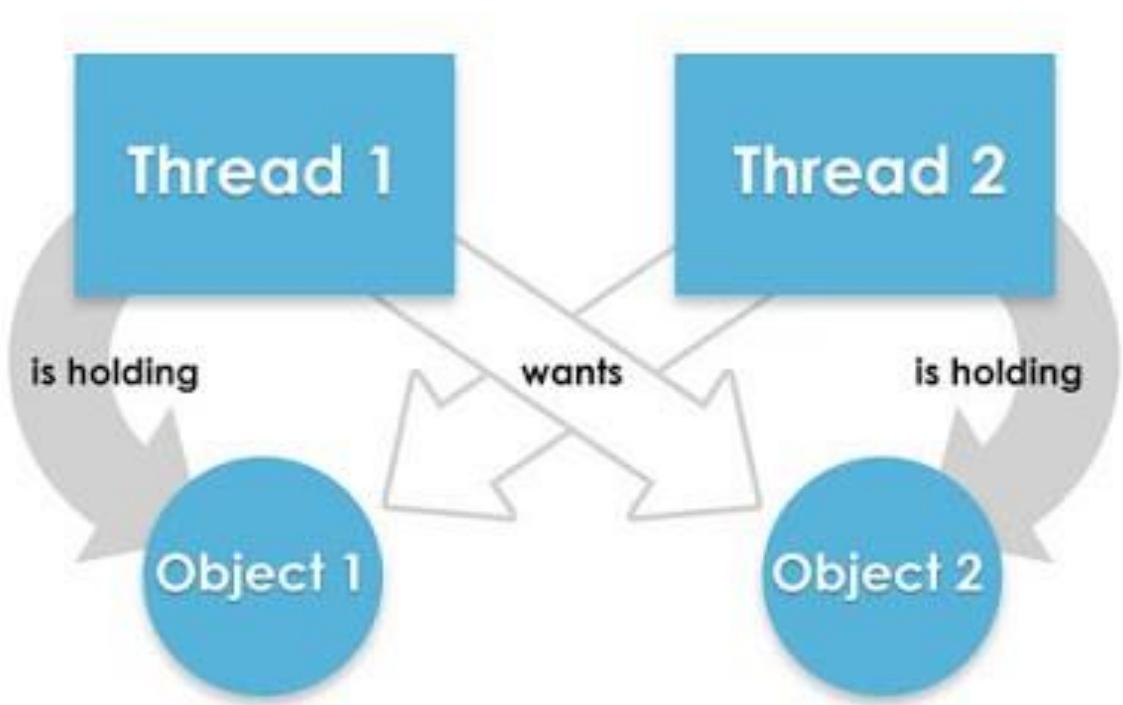
---

38

## WHAT IS DEADLOCK?

 A deadlock is a situation in which two or more processes cannot complete their work. They are blocked and each thread waiting for other threads to release resource they want.

For example Object 1 is hold by Thread 1 and Object 2 is hold by Thread 2. Now Thread 1 wantd Object 2 but as it is hold by Thread 2 and it is not released therefore a deadlock will occur.



39

## WHAT IS THE DIFFERENCE BETWEEN THREADS AND TASKS?

Tasks are wrapper around Thread and ThreadPool classes. Below are some major differences between Threads and Tasks:

1. A Task can return a result but there is no proper way to return a result from Thread.

2. We can apply chaining on multiple tasks but we can not in threads.
3. We can wait on Tasks without using Signalling. But in Threads we have to use event signals like AutoResetEvent and ManualResetEvent.
4. We can apply Parent/Child relationship in Tasks. A Task at one time becomes parent of multiple tasks. Parent Task does not complete until it's child tasks are completed. We do not have any such mechanism in Thread class.
5. Child Tasks can propagate their exceptions to parent Task and All child exceptions are available in AggregateException class.
6. Tasks has in-build cancellation mechanism using CancellationToken class.

## Task

```
1. static void Main(string[] args) {
2. Task < string > obTask = Task.Run(() => (
3. return "Hello");
4. Console.WriteLine(obTask.Result);
5. }
```

## Thread

```
1. static void Main(string[] args) {
2. Thread thread = new Thread(new ThreadStart(getMyName));
3. thread.Start();
```

4. }

The bottom line is that Task is almost always the best option; it provides a much more powerful API and avoids wasting OS threads.

[More in Detail](#)

---

40

## WHAT IS THE ASYNC AND AWAIT IN TASK?

Suppose, we are using two methods as Method1 and Method2 respectively and both the methods are not dependent on each other and Method1 is taking a long time to complete its task. In Synchronous programming, it will execute the first Method1 and it will wait for completion of this method and then it will execute Method2. Thus, it will be a time intensive process even though both the methods are not depending on each other.

We can run all the methods parallelly by using the simple thread programming but it will block UI and wait to complete all the tasks. To come out of this problem, we have to write too many codes in traditional programming.

So the simple solution for this it to use the `async` and `await` keywords, then we will get the solutions in much less code.

```
class Program
{
 static void Main(string[] args)
 {
 Method1();
 Method2();
 Console.ReadKey();
 }

 public static async Task Method1()
 {
 await Task.Run(() =>
 {
 for (int i = 0; i < 100; i++)
 {
 Console.WriteLine(" Method 1");
 }
 });
 }
}
```

```
}

public static void Method2()
{
 for (int i = 0; i < 25; i++)
 {
 Console.WriteLine(" Method 2");
 }
}

```



## #RESUME TIP 5

### **FOCUS MORE INDIVIDUAL ROLES AND RESPONSIBILITIES (IN BULLETS)**

Do not describe your projects in detail, rather focus more on roles and responsibilities (atleast 3-5 per project). Always list down your roles and responsibilities in bullets. Interviewer are more interested in individual contributions.

# Chapter 4 - ASP.NET MVC

41

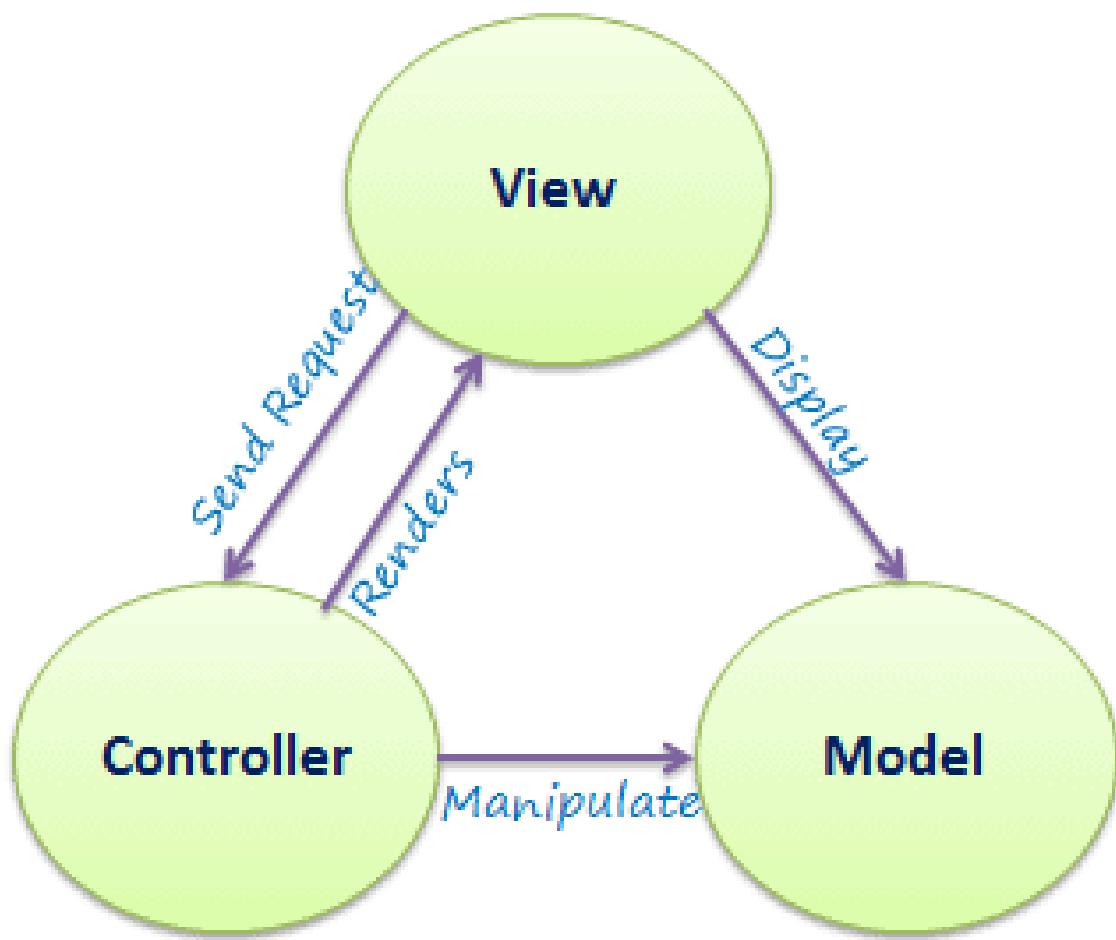
## WHAT IS MVC (MODEL VIEW CONTROLLER)?

Model–View–controller (MVC) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representation of information from the way that information is presented to or accepted from the user.

MVC is a framework for building web applications using an MVC (Model View Controller) design:

- The Model represents the application core (for instance a list of database records).
- The View displays the data.
- The Controller handles the input (to the database records).

The MVC model also provides full control over HTML, CSS, and JavaScript.



[More in Detail](#)

---

## WHAT ARE THE ADVANTAGES OF MVC?

- 1. SOC (SEPARATION OF CONCERNS)** - Separation of Concerns is one of the core advantages of ASP.NET MVC. The MVC framework provides a clean separation of the UI, Business Logic, Model or Data.
- 2. MULTIPLE VIEW SUPPORT** - Due to the separation of the model from the view, the user interface can display multiple views of the same data at the same time.
- 3. CHANGE ACCOMMODATION** - User interfaces tend to change more frequently than business rules (different colors, fonts, screen layouts, and levels of support for new devices such as cell phones or PDAs) because the model does not depend on the views, adding new types of views to the system generally does not affect the model. As a result, the scope of change is confined to the view.
- 4. MORE CONTROL** - The ASP.NET MVC framework provides more control over HTML, JavaScript, and CSS than the traditional Web Forms.

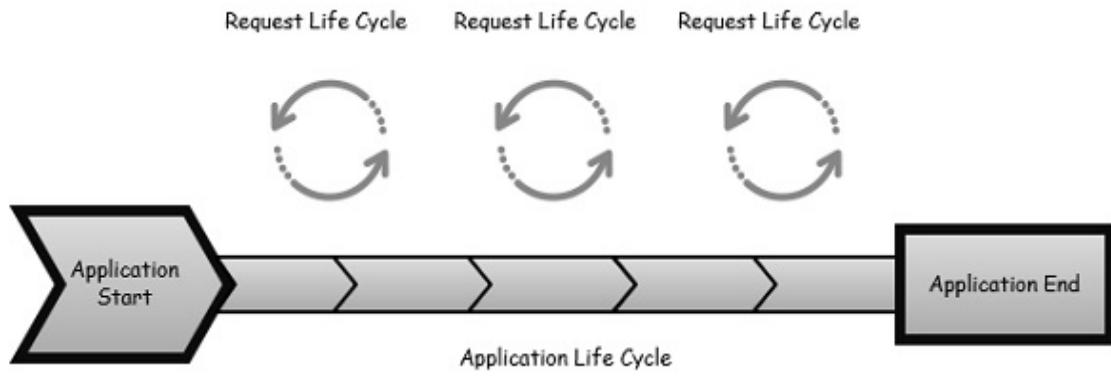
- 5. TESTABILITY** - ASP.NET MVC framework provides better testability of the Web Application and good support for test driven development too.
  - 6. LIGHTWEIGHT** - ASP.NET MVC framework doesn't use View State and thus reduces the bandwidth of the requests to an extent.
  - 7. FULL FEATURES OF ASP.NET** - One of the key advantages of using ASP.NET MVC is that it is built on top of the ASP.NET framework and hence most of the features of the ASP.NET like membership providers, roles, etc can still be used.
- 

43

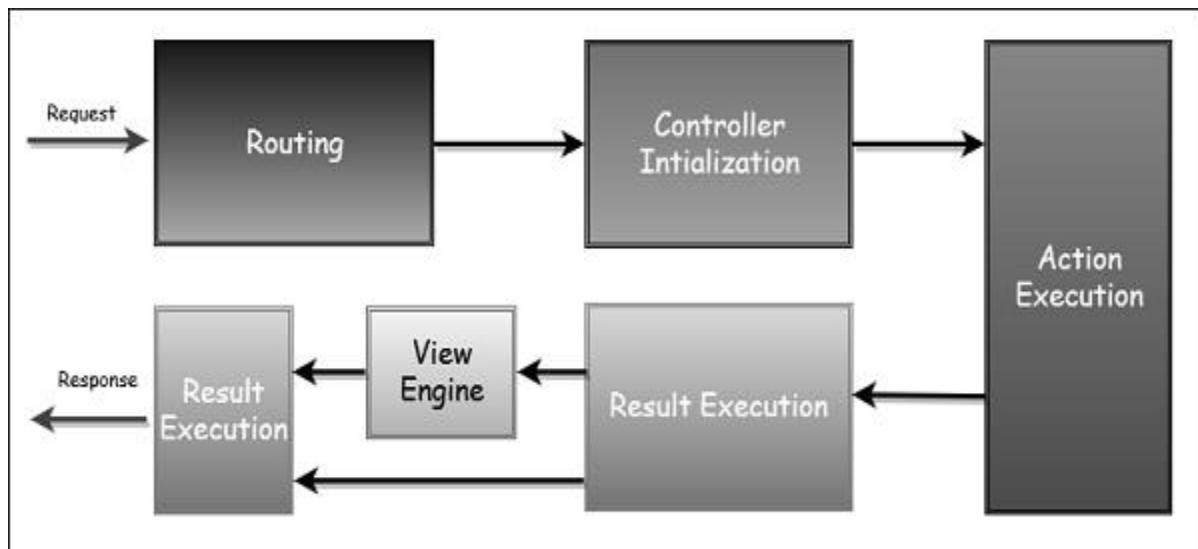
## EXPLAIN MVC LIFE CYCLE?

MVC has two life cycles –

- ⊕ The application life cycle - The application life cycle refers to the time at which the application process actually begins running IIS until the time it stops.



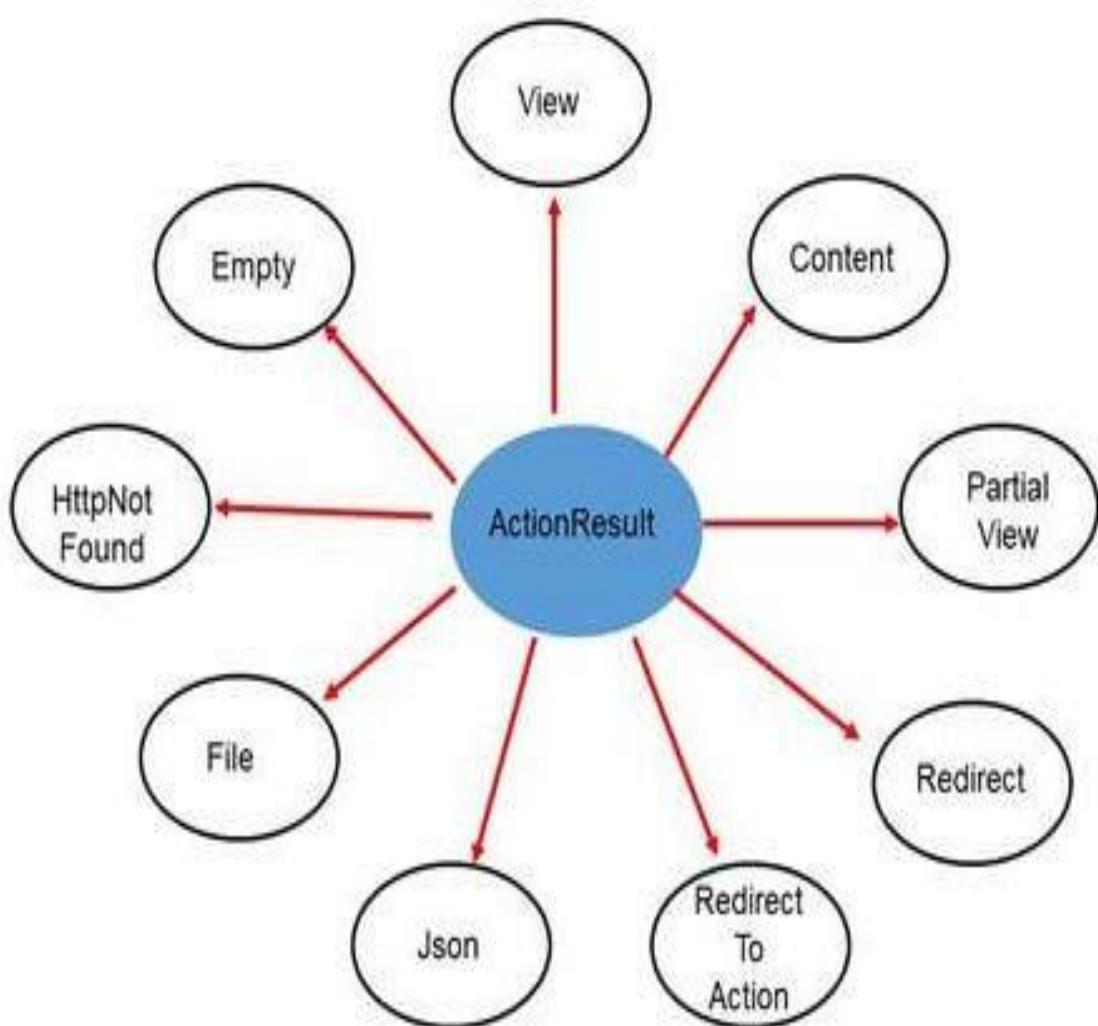
- + The request life cycle - It is the sequence of events that happen every time an HTTP request is handled by our application.



44

## LIST OUT DIFFERENT RETURN TYPES OF A CONTROLLER ACTION METHOD?

There are total of nine return types we can use to return results from the controller to view. The base type of all these result types is ActionResult.

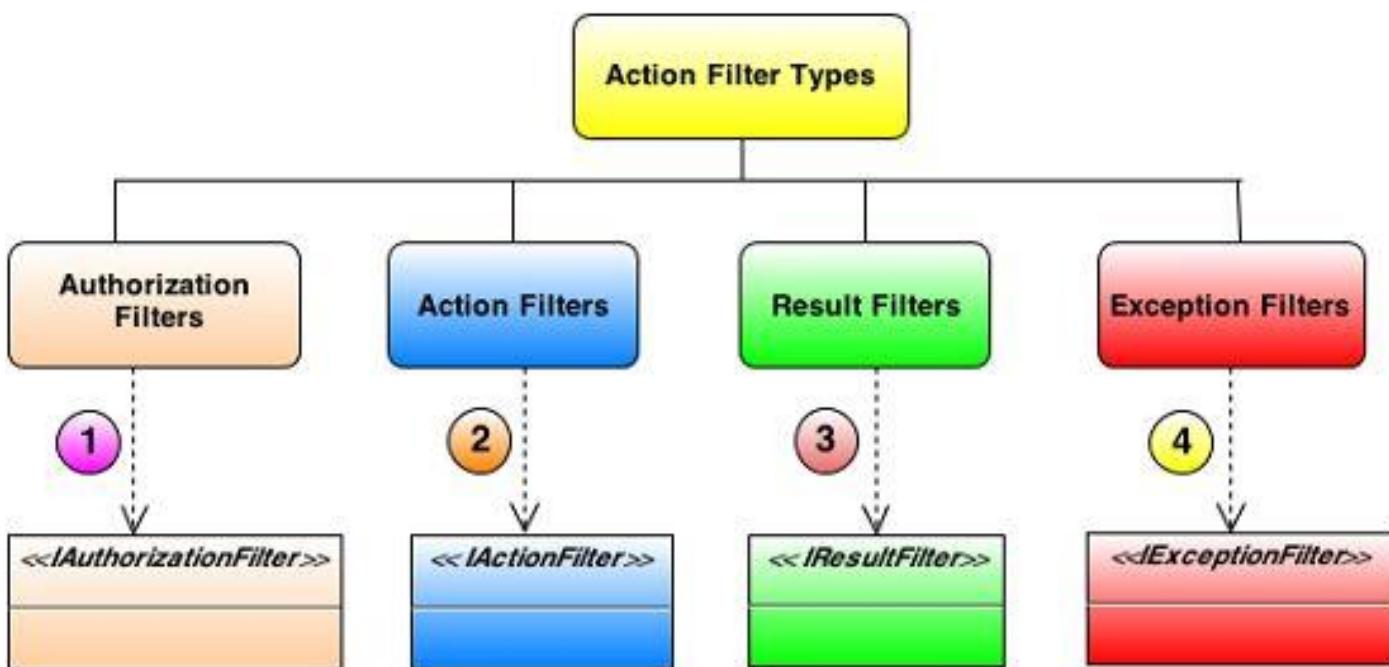


- 1) **VIEWRESULT** - This return type is used to return a webpage from an action method.
- 2) **PARTIALVIEWRESULT** - This return type is used to send a part of a view that will be rendered in another view.
- 3) **REDIRECTRESULT** - This return type is used to redirect to any other controller and action method depending on the URL.
- 4) **REDIRECTTROUTERRESULT** - This return type is used when we want to redirect to any other action method.
- 5) **CONTENTRESULT** - This return type is used to return HTTP content type like text/plain as the result of the action.
- 6) **JSONRESULT** - This return type is used when we want to return a JSON message.
- 7) **JAVASCRIPTRESULT** - This return type is used to return JavaScript code that will run in the browser.
- 8) **FILERESULT** - This return type is used to send binary output in response.
- 9) **EMPTYRESULT** - This return type is used to return nothing (void) in the result.

45

## WHAT ARE ACTION FILTERS IN MVC?

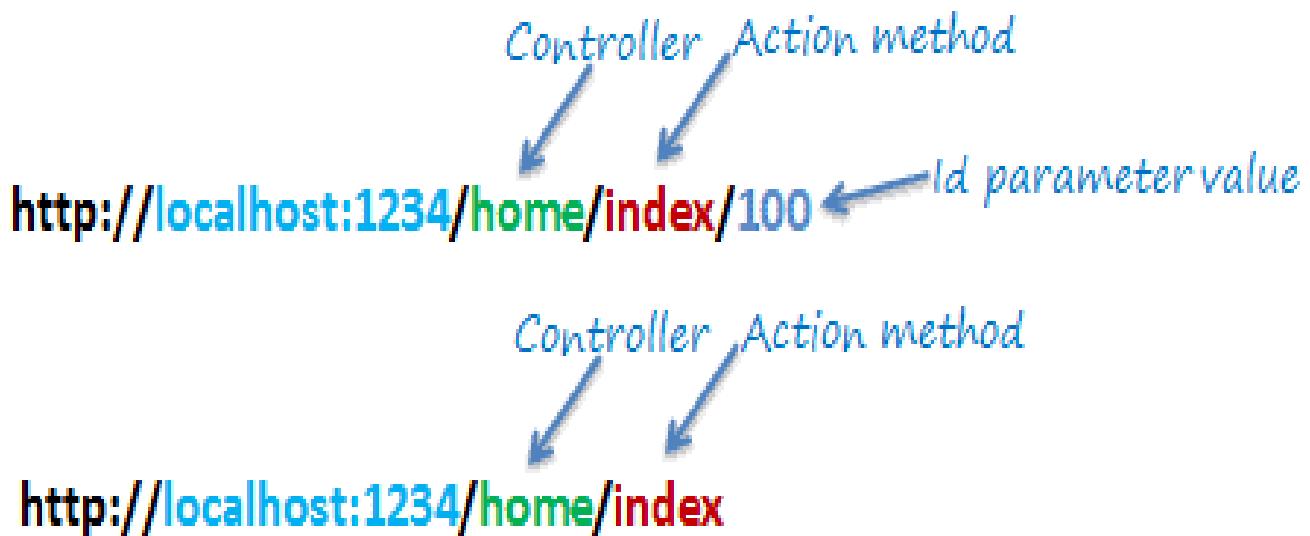
- + Action Filters are additional attributes that can be applied to either a controller section or the entire controller to modify the way in which action is executed.
- + This **filter** will be called before and after the **action** starts executing and after the **action** has executed.



46

## WHAT IS ROUTING IN MVC?

Routing is a mechanism to process the incoming URL that is more descriptive and gives the desired response. In this case, URL is not mapped to specific files or folder as was the case of earlier days web sites.



---

47

## WHAT IS THE DIFFERENCE BETWEEN VIEW DATA, VIEW BAG & TEMP DATA?

- + In ASP.NET MVC there are three ways to pass/store data between the controllers and views.

### VIEW DATA

1. ViewData is used to pass data from CONTROLLER TO VIEW.
2. REQUIRES TYPECASTING for complex data types.

### VIEW BAG

1. ViewBag is also used to pass data from the controller to the respective view.
2. It DOESN'T REQUIRE TYPECASTING for the complex data type.

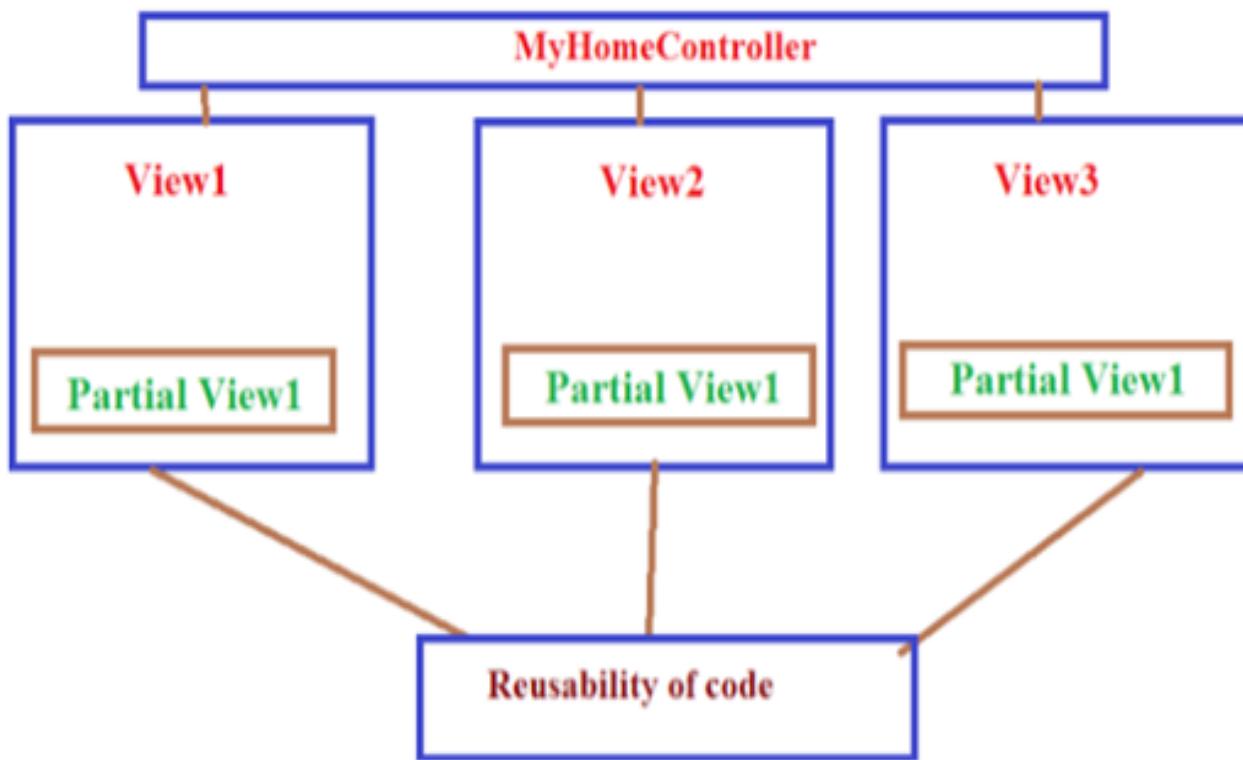
### TEMPDATA

1. TempData is used to pass data from CONTROLLER TO CONTROLLER.

2. It keeps the information for the time of an HTTP Request.  
This means only from one page to another.
  3. It requires typecasting for complex data types and checks for null values to avoid an error.
-

## WHAT IS PARTIAL VIEW?

- Partial view in ASP.NET MVC is special view which renders a



portion of view content.

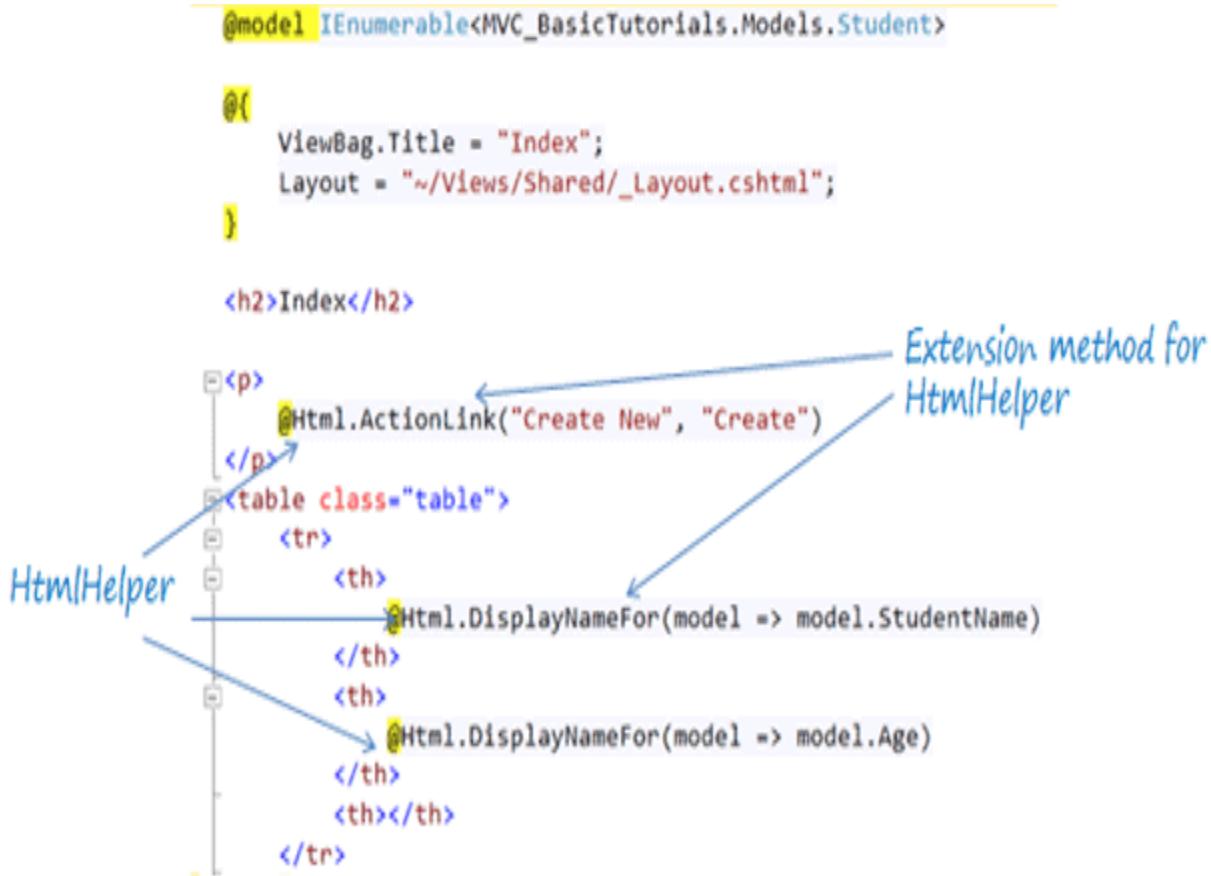
It is just like a user control of a web form application. Partial can be reusable in multiple views. It helps us to reduce code duplication.

49

## WHAT ARE HTML HELPERS IN MVC?

- + The `HtmlHelper` class renders HTML controls in the razor view.

It binds the model object to HTML controls to display the value of model properties into those controls and also assigns the value of the controls to the model properties while submitting a web form. So always use the `HtmlHelper` class in razor view instead of writing HTML tags manually.



[More in Detail](#)

50

## EXPLAIN ATTRIBUTE BASED ROUTING IN MVC?

In ASP.NET MVC 5.0 we have a new attribute route, By using the "Route" attribute we can define the URL structure.

For example in the below code we have decorated the "GotoAbout" action with the route attribute. The route attribute says that the "GotoAbout" can be invoked using the URL structure "Users/about".

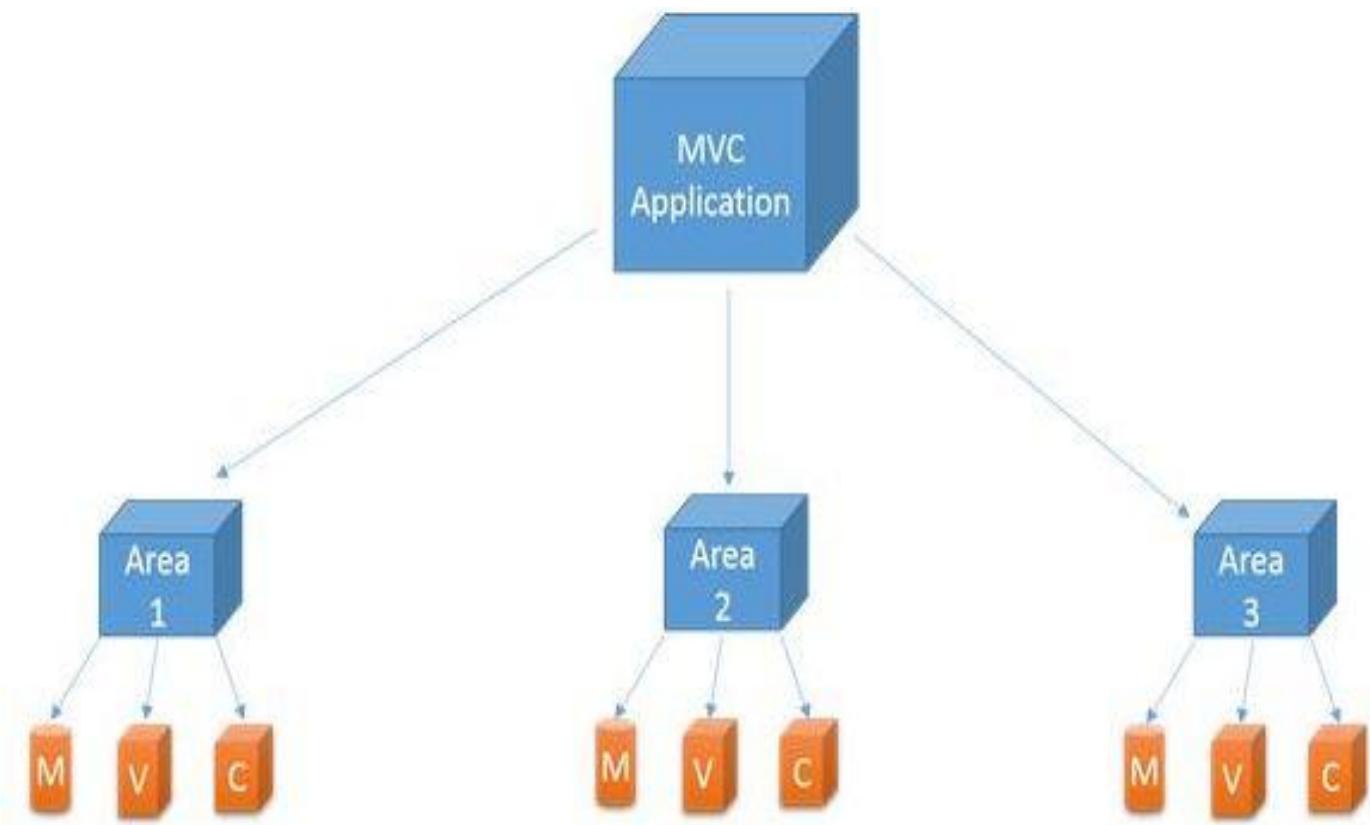
```
public class HomeController: Controller
{
 [Route("Users/about")]
 Public ActionResult GotoAbout()
 {
 return View();
 }
}
```

---

51

## EXPLAIN AREAS IN MVC?

Areas are just a way to divide or “isolate” the modules of large applications in multiple or separated MVC.



52

## EXPLAIN THE CONCEPT OF MVC SCAFFOLDING?

- Scaffolding is a code generation framework for ASP.NET Web applications. Using scaffolding can reduce the amount of time to develop standard data operations in your project.

### Scaffolding Templates:

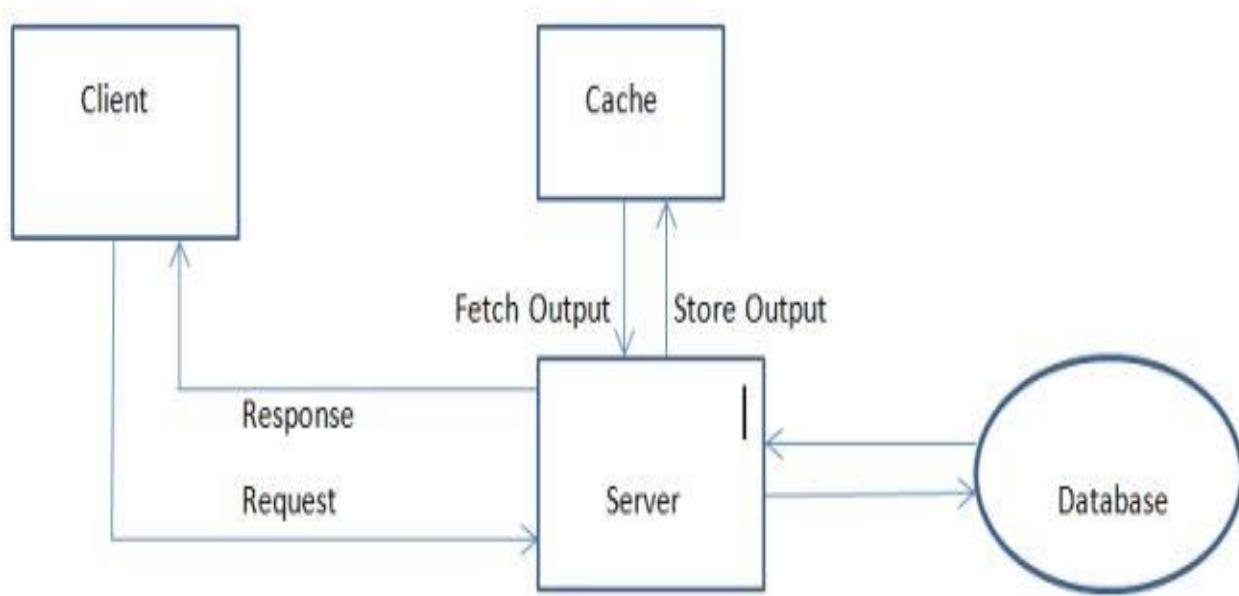


53

## WHAT IS OUTPUT CACHING IN MVC?

- + The main purpose of using Output Caching is to dramatically improve the performance of an ASP.NET MVC Application.

- It enables us to cache the content returned by any controller method so that the same content does not need to be generated each time the same controller method is invoked.
- Output Caching has huge advantages, such as it reduces server round trips, reduces database server round trips, reduces network traffic etc.
- The output cache is located on the Web server where the request was processed.





## #RESUME TIP 6

### **FOCUS MORE ON INDIVIDUAL ROLES AND RESPONSIBILITIES (IN BULLETS)**

Do not describe your projects in detail, rather focus more on roles and responsibilities (atleast 3-5 per project). Always list down your roles and responsibilities in bullets. Interviewer are more interested in individual contributions.

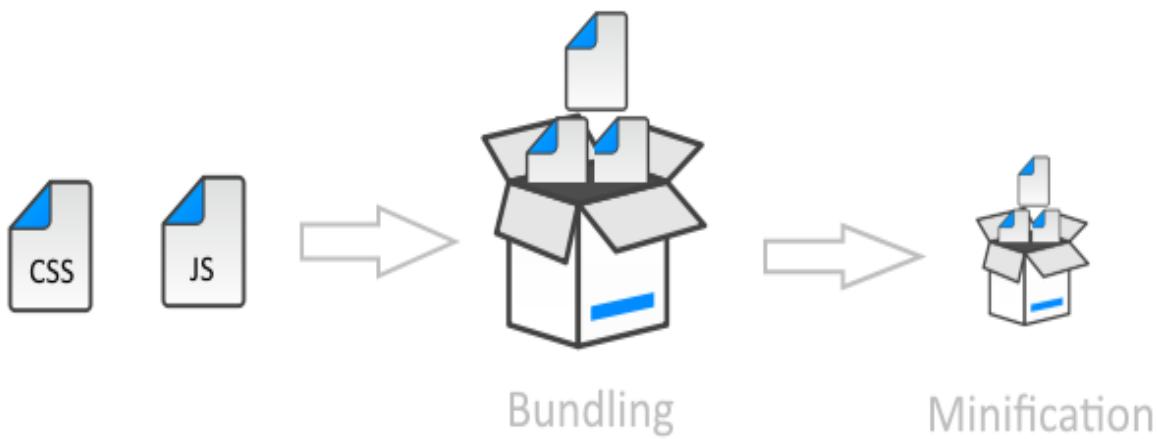
54

## WHAT IS BUNDLING AND MINIFICATION IN MVC?

Bundling and minification are two new techniques introduced to improve request load time.

**BUNDLING** - It lets us combine multiple JavaScript (.js) files or multiple cascading style sheet (.css) files so that they can be downloaded as a unit, rather than making individual HTTP requests.

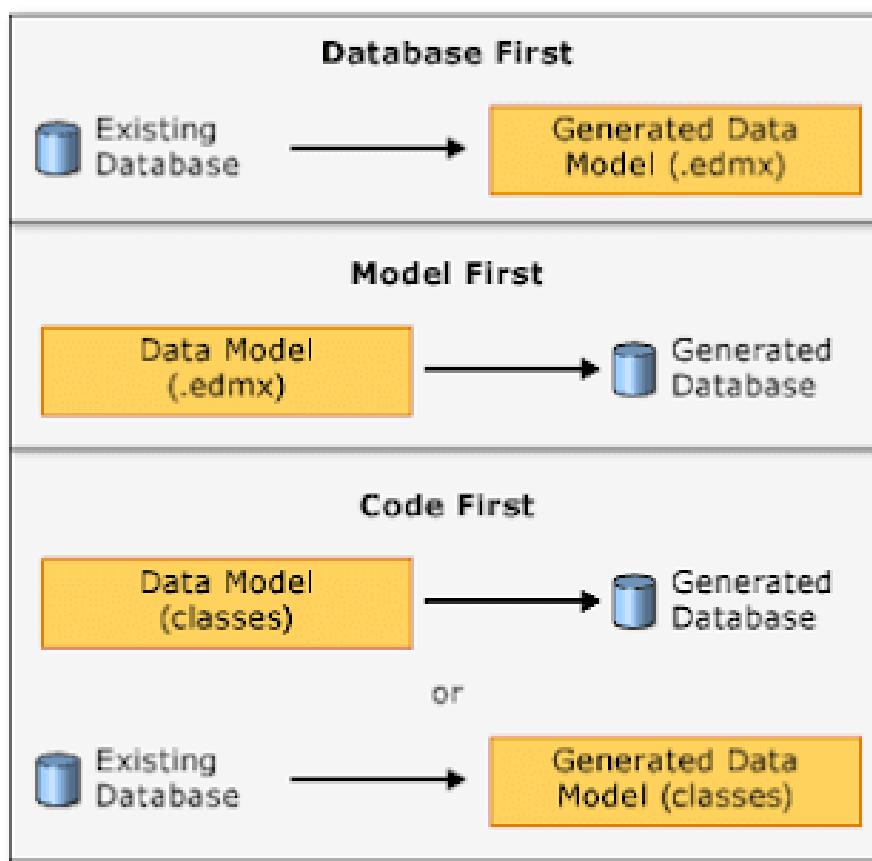
**MINIFICATION** - It squeezes out whitespace and performs other types of compression to make the downloaded files as small as possible.



55

## WHAT DIFFERENT APPROACHES TO MODEL ENTITIES IN ENTITY FRAMEWORK ?

1. Database First
2. Model First
3. Code First



56

## WHAT IS DIFFERENCE BETWEEN MVC AND WEB FORMS?

- a) In MVC View and logic are separate, it has separation of concerns theory. MVC 3 onwards has .aspx page as .cshtml.
  - b) In Web forms no separation of concerns; Views are tightly coupled with logic (.aspx.cs / .vb file).
- 
- a) MVC introduced concept of routing for route based URL.  
Routing is declared in Global.asax for example.
  - b) Web forms has file-based routing .Redirection is based on pages.
- 
- a) MVC Support Razor syntax as well as .aspx
  - b) Web forms support web forms syntax only.
- 
- a) In MVC State management handled via TempData, ViewBag, and ViewData. Since the controller and view are not dependent and also since there is no view state concept in ASP.NET, MVC keeps the pages lightweight.

b) In Web forms State management handled via View State.  
Large viewstate, in other words increase in page size.

a) MVC has Partial Views  
b) Web forms has User Controls

a) MVC has HTML Helpers  
b) Web forms has Server Controls

a) In MVC Multiple pages can have the same controller to satisfy their requirements. A controller may have multiple Actions (method name inside the controller class).

b) In Web forms Each page has its own code, in other words direct dependency on code. For example Sachin.aspx is dependent on Sachin.aspx.cs (code behind) file.

a) In MVC Unit Testing is quite easier than ASP.Net Web forms Since a web form and code are separate files.

b) Web forms has Direct dependency, tight coupling raises issues in testing.

a) MVC has Layouts  
b) Web forms has Master pages

57

## HOW CAN WE PASS THE DATA FROM CONTROLLER TO VIEW IN MVC?

By using ViewData, ViewBag and TempData

---

58

## WHAT CAN BE DONE TO IMPROVE PERFORMANCE IN ASP.NET MVC?

Here are some of the techniques to improve performance.

- Make use of a profiler to discover memory leaks and performance problems in your application.
- Run your site in Release mode, not Debug mode.
- Use cookies for frequently accessed non sensitive information.
- Use Forms Authentication, Keep your frequently accessed sensitive data in the authentication ticket.
- Use AJAX to update components of your UI, avoid a whole page update when possible.
- Use CDN's for scripts and media content to improve loading on the client side.

---



59

## WHAT IS THE USE OF ACTION FILTERS IN AN MVC APPLICATION?

Action Filters allow us to add pre-action and post-action behavior to controller action methods.

[More in Detail](#)

---



## #RESUME TIP 7

### YOUR RECENT PROJECT MUST BE ON THE TOP (REVERSE-CHRONOLOGICAL ORDER)

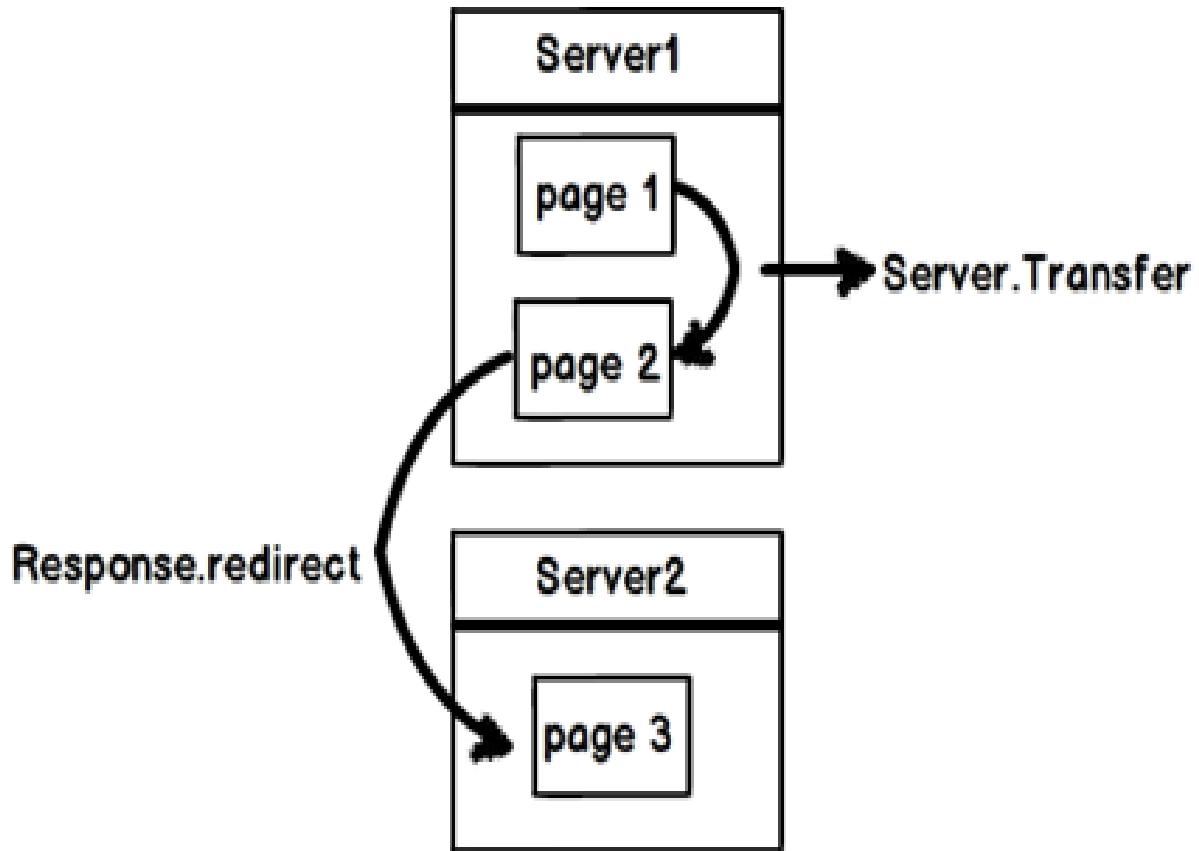
Your recent project must be on the top of project list.

# Chapter 5 - ASP.NET WEBFORMS

60

## WHAT IS THE DIFFERENCE BETWEEN SERVER.TRANSFER AND RESPONSE.REDIRECT?

- + In Server.Transfer page processing transfers from one page to the other page WITHOUT MAKING A ROUND-TRIP BACK to the client's browser. This provides a faster response with a little less overhead on the server. The clients url history list or current url Server does not update in case of Server.Transfer.
- + Response.Redirect is used to redirect the user's browser to another page or site. It performs trip back to the client where the client's browser is redirected to the new page. The user's browser history list is updated to reflect the new address.



61

## WHERE THE VIEWSTATE IS STORED AFTER THE PAGE POSTBACK?

ViewState is stored in a HIDDEN FIELD on the page at client side. ViewState is transported to the client and back to the server, and is not stored on the server or any other external source.

62

## WHAT IS CACHING?

+ Caching is a technique used to increase performance by keeping frequently accessed data or files in memory. The request for a cached file/data will be accessed from cache instead of actual location of that file.

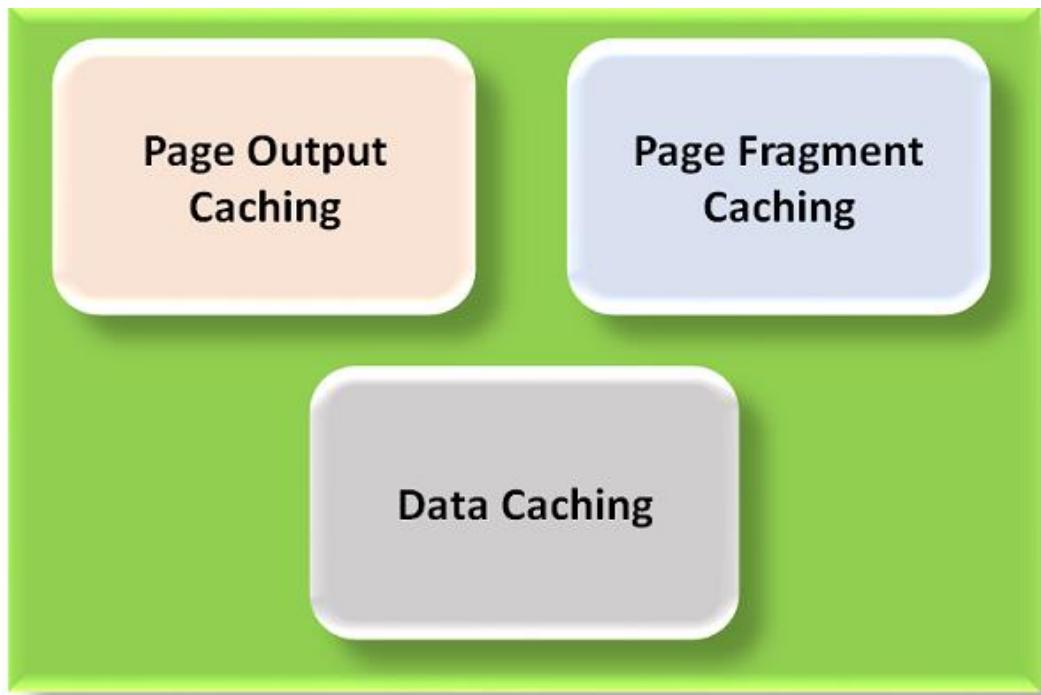
[More in Detail](#)

---

63

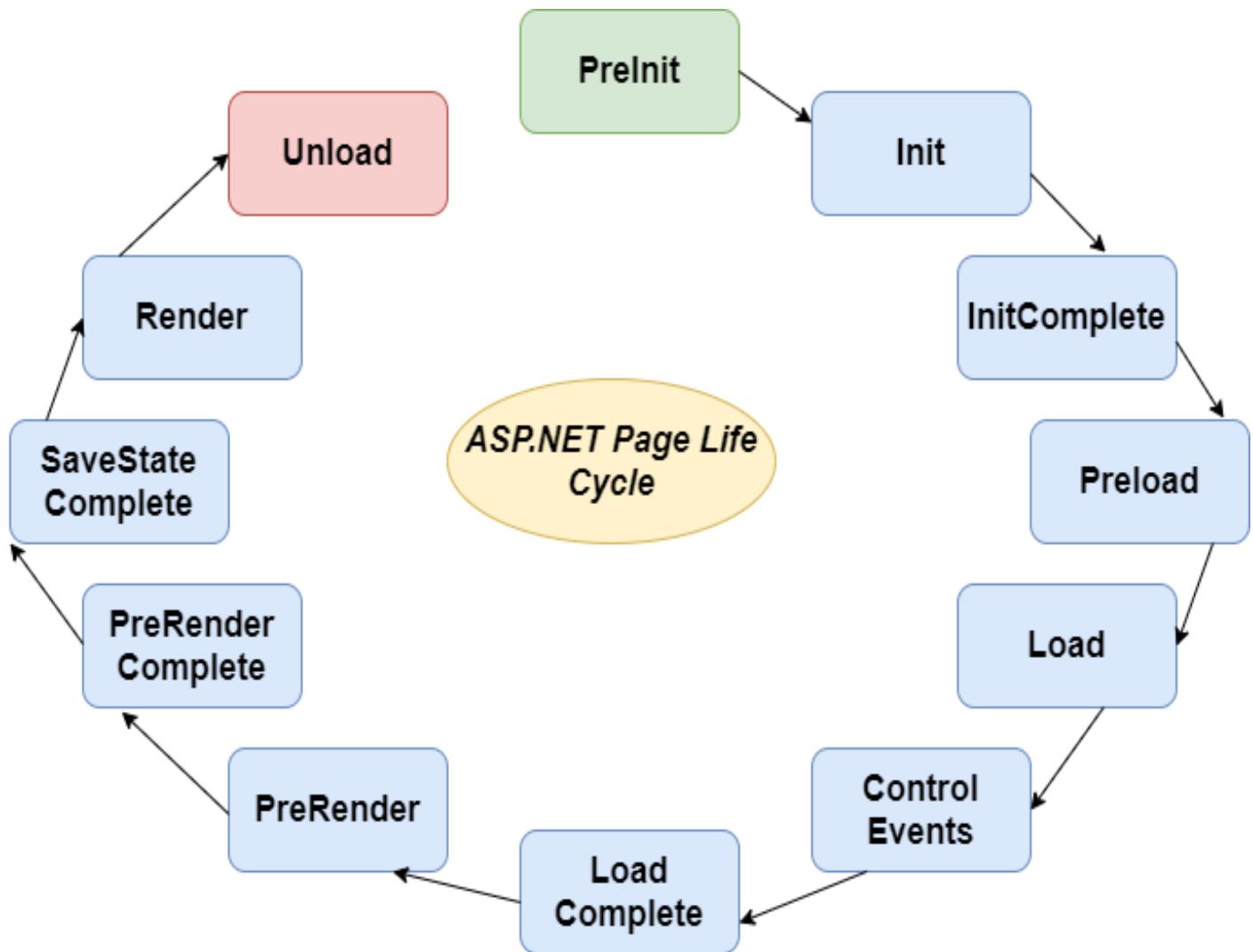
## WHAT ARE THE DIFFERENT TYPES OF CACHING?

1. Page Output Caching
2. Page Fragment Caching
3. Data Caching



64

## LIST THE EVENTS IN PAGE LIFE CYCLE.



[More in Detail](#)

65

## WHAT IS AUTHENTICATION AND AUTHORIZATION IN ASP.NET?

- + Authentication is the process of obtaining some sort of credentials from the users and using those credentials to verify the user's identity.
  - + Authorization is the process of allowing an authenticated user access to resources. Authentication is always precedes to Authorization; even if your application lets anonymous users connect and use the application, it still authenticates them as being anonymous.
- 

66

## WHAT ARE THE TYPES OF AUTHENTICATION IN ASP.NET?

### 1. FORMS AUTHENTICATION

form authentication is based on cookies, the authentication and permission settings are stored in cookies. However, we can also

use form authentication without cookies, and in cookie-less form authentication we can use query string for passing user details.

## 2. PASSPORT AUTHENTICATION

Passport authentication is a centralized authentication service provided by Microsoft. The .NET Passport single sign-in service.

## 3. WINDOWS AUTHENTICATION

We use windows authentication when we are creating a web application for a limited number of users who already have Windows account and this type of authentication is quite useful in an intranet environment.

[More in Detail](#)

---

67

## IN WHICH EVENT ARE THE CONTROLS FULLY LOADED?

Page load event.

---



68

## HOW WE CAN FORCE ALL THE VALIDATION CONTROLS TO RUN?

The `Page.Validate()` method is used to force all the validation controls to run and to perform validation.

---



## #RESUME TIP 8

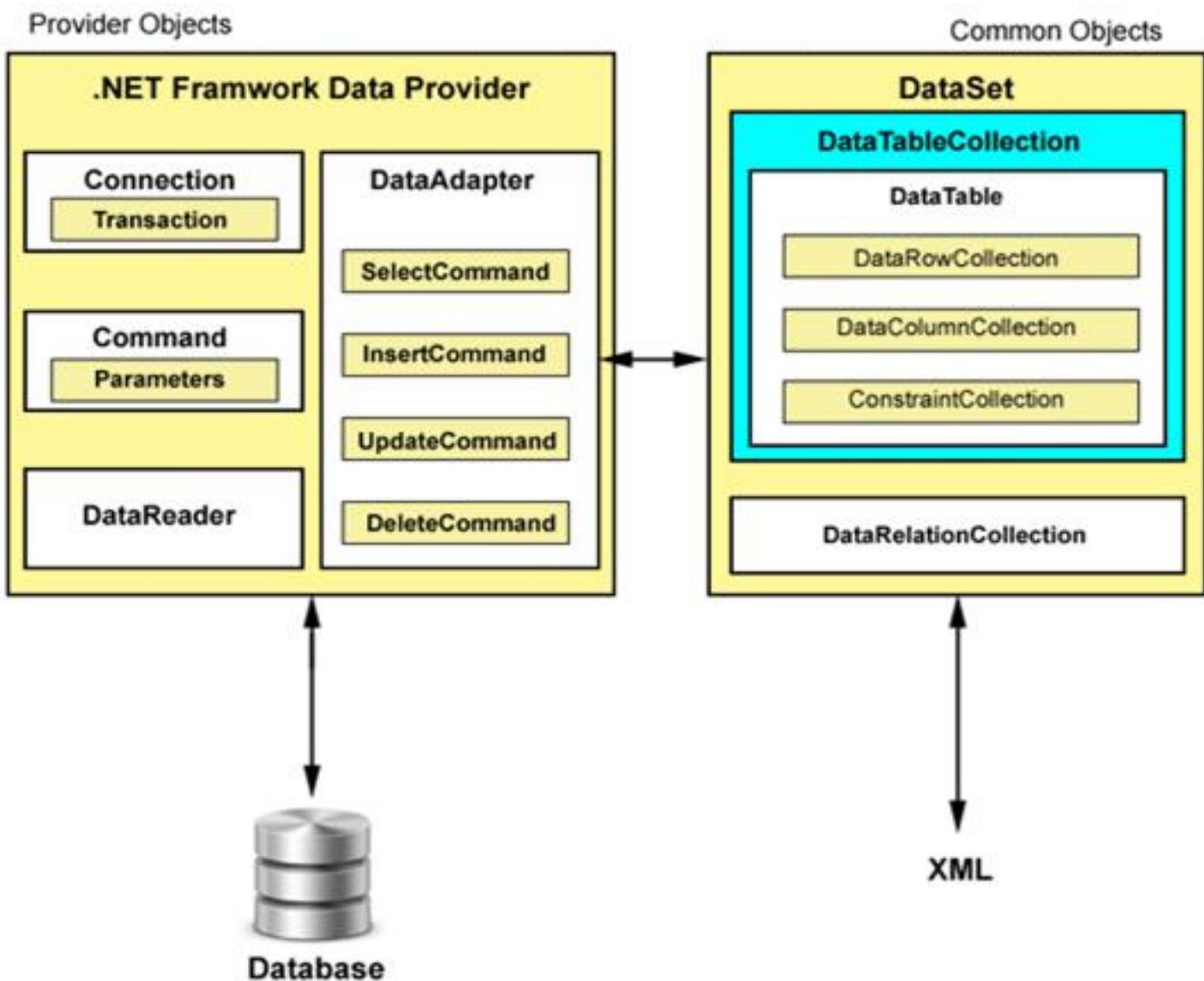
### **AVOID LONG OBJECTIVE STATEMENT.**

In technical interviews your objective statement rarely matters, especially if you have less than 5 years of experience. So why give it more space in a single page resume. Keep it short and simple.

# Chapter 6 - ADO.NET

69

## EXPLAIN ADO.NET ARCHITECTURE?



[More in Detail](#)

## WHAT IS CONNECTED ARCHITECTURE AND DISCONNECTED ARCHITECTURE?

### DATASET

It is a type of disconnected architecture. Disconnected architecture means, you don't need to connect always when you want to get data from the database. You can get data from dataset; basically DataSet is a collection of datatables. We can store the database table, view data in the DataSet and can also store the xml value in dataset and get it if required.

### DATAREADER

It is a connected architecture, which means when you require data from the database you need to connect with database and fetch the data from there. You can use if you need updated data from the database in a faster manner.

71

## 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
- 

72

## WHAT ARE ALL COMPONENTS OF ADO.NET DATA PROVIDER?

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.
- 

73

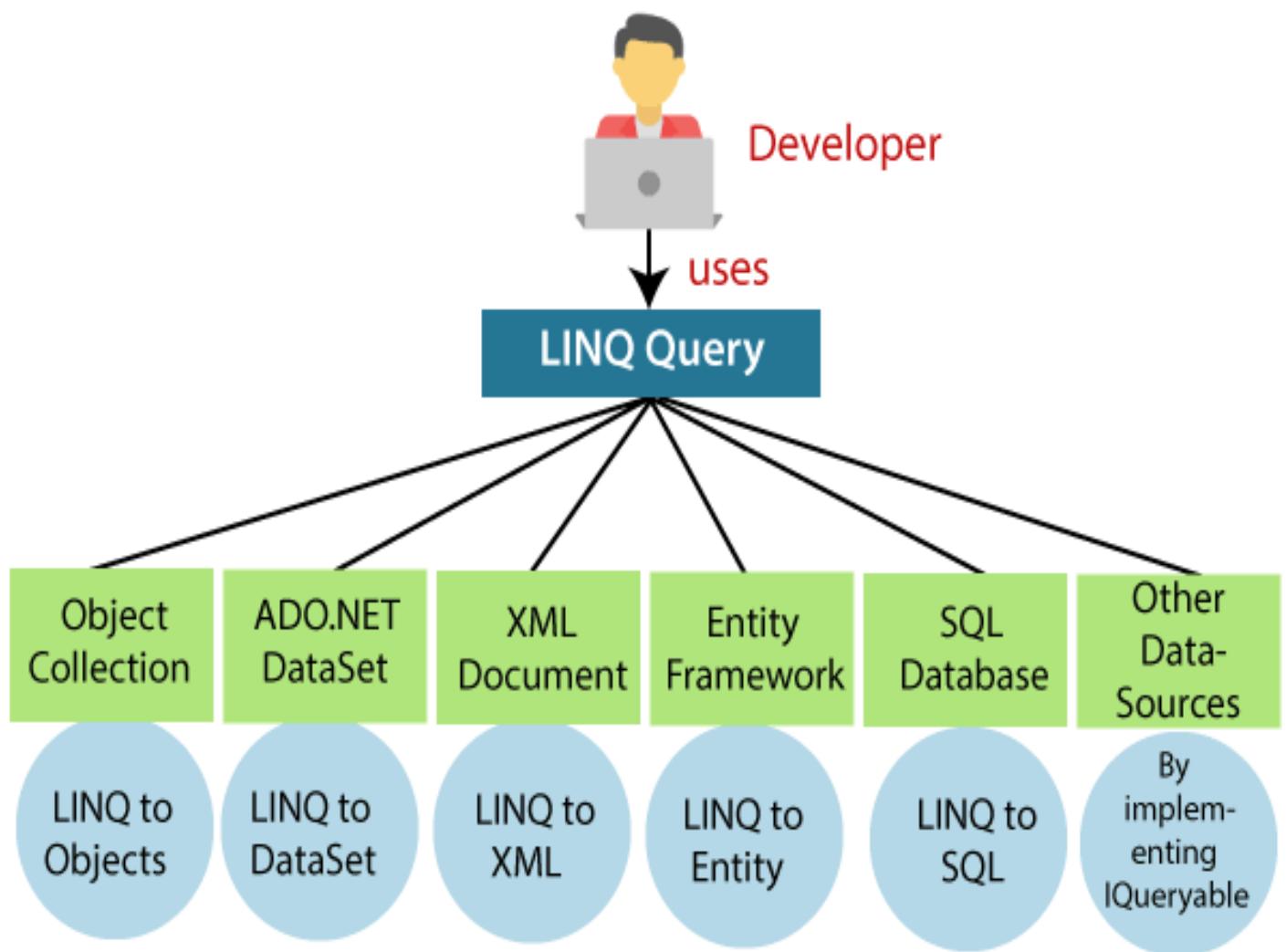
## WHAT ARE THE DIFFERENT EXECUTE METHODS OF ADO.NET?

- ExecuteScalar – Returns single value from the dataset

- ExecuteNonQuery – Returns resultset from dataset and it has multiple values
  - ExecuteReader – Forwardonly resultset
  - ExecuteXMLReader – Build XMLReader object from a SQL Query
-

## WHAT IS LINQ?

LINQ is native query language for .NET framework and it is specially designed to support queries with the .net applications.



[More in Detail](#)

75

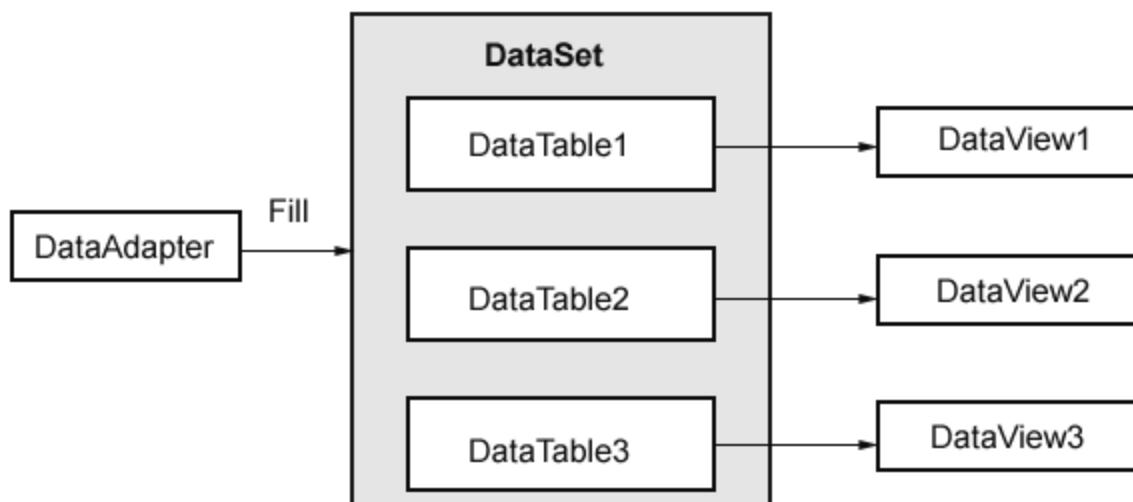
## WHAT IS THE DIFFERENCE BETWEEN ENTITY FRAMEWORK AND ADO.NET?

**ENTITY FRAMEWORK:** Entity Framework happens to be the ORM model and utilizes LINQ to read databases and codes are generated. Entity Framework lowers development cost and time. It also helps the designers to visually develop models and delineating of the database. Moreover, Entity Framework offers the power to program a conceptual prototype.

**ADO.NET:** ADO.Net happens to be ORM but it creates an elevated abstract object prototype over ADO.Net components. ADO.Net functions on disjunction architecture. Data happens to hold as XML. ADO.Net is extremely swift and data may be collected in data sets. ADO.Net also abstracts esoteric programming which is necessary for accessing a database.

## WHAT IS THE DIFFERENCE BETWEEN DATASET AND DATATABLE?

DataTable represents a single table in the database. It has rows and columns. There is no much difference between dataset and datatable, dataset is simply the collection of datatables.



## WHAT ARE THE AUTHENTICATION TECHNIQUES USED TO CONNECT TO SQL SERVER?

SQL Server should authenticate before performing any activity in the database. There are two types of authentication:

- Windows Authentication – Use authentication using Windows domain accounts only.
- SQL Server and Windows Authentication Mode – Authentication provided with the combination of both Windows and SQL Server Authentication.



## #RESUME TIP 9

### EXPERIENCE OVER EDUCATION

Even if you have 2 months of project experience then try to elaborate it more as comparison to your education. Everyone know the value of experience.

# Chapter 7 - SQL FOR DEVELOPER

78

## WHAT IS THE DIFFERENCE BETWEEN STORED PROCEDURE AND FUNCTIONS?

- a) A function has a return type and returns a value.
  - b) A procedure does not have a return type. But it returns values using the OUT parameters.
- 
- a) You cannot use a function with Data Manipulation queries. Only Select queries are allowed in functions.
  - b) You can use DML queries such as insert, update, select etc... with procedures.
- 
- a) A function does not allow output parameters
  - b) A procedure allows both input and output parameters.
- 
- a) You cannot manage transactions inside a function.
  - b) You can manage transactions inside a function.
- 
- a) You cannot call stored procedures from a function.

- b) You can call a function from a stored procedure.
  - a) You can call a function using a select statement.
  - b) You cannot call a procedure using select statements.
- 

79

## WHAT ARE INDEXES AND WHAT IS THE DIFFERENCE BETWEEN CLUSTERED AND NON-CLUSTERED INDEX?

 SQL Indexes are used in relational databases to quickly retrieve data. They are similar to indexes at the end of the books whose purpose is to find a topic quickly.

- a) A Clustered index is a type of index in which table records are physically reordered to match the index.
- b) A Non-Clustered index is a special type of index in which logical order of index does not match physical stored order of the rows on disk.

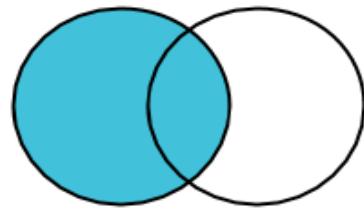
- a) A table can have only one clustered index.
  - b) A table can have multiple non-clustered index.
- 
- a) Clustered index is faster.
  - b) Non-clustered index is slower.
- 
- a) In clustered index, index is the main data.
  - b) Non-Clustered index requires more memory for operations.
- 
- a) Clustered index has inherent ability of storing data on the disk.
  - b) Non-Clustered index does not have inherent ability of storing data on the disk.
- 
- a) Clustered index store pointers to block not data.
  - b) Non-Clustered index store both value and a pointer to actual row that holds data.
- 
- a) In Clustered index leaf nodes are actual data itself.
  - b) In Non-Clustered index, index key defines order of data within index.

- a) In Clustered index, Clustered key defines order of data within table.
  - b) In Non-Clustered index, index key defines order of data within index.
- 

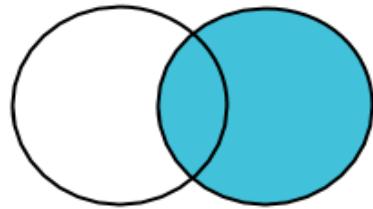
80

## WHAT ARE THE TYPES OF JOIN AND EXPLAIN EACH?

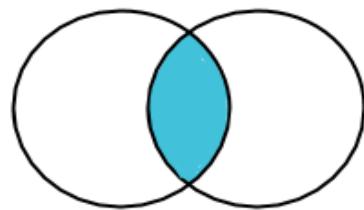
There are various types of join which can be used to retrieve data and it depends on the relationship between tables.



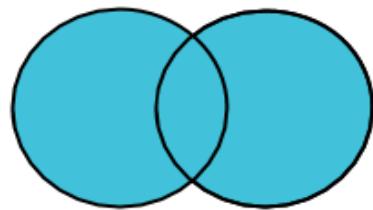
**Left Join**



**Right Join**



**Inner Join**



**Full Outer Join**

■ **INNER JOIN** - Inner join return rows when there is at least one match of rows between the tables.

■ **RIGHT JOIN** - Right join return rows which are common between the tables and all rows of Right hand side table. Simply, it returns all the rows from the right hand side table even though there are no matches in the left hand side table.

■ **LEFT JOIN** - Left join return rows which are common between the tables and all rows of Left hand side table. Simply, it returns all the rows from Left hand side table even though there are no matches in the Right hand side table.

 **FULL JOIN** - Full join return rows when there are matching rows in any one of the tables. This means, it returns all the rows from the left hand side table and all the rows from the right hand side table.

[More in Detail](#)

---

81

## WHAT IS NORMALIZATION?

 Normalization is the process of minimizing redundancy and dependency by organizing fields and table of a database.

Normalization rules are divided into the following normal forms:

|     |                                                                                                                         |
|-----|-------------------------------------------------------------------------------------------------------------------------|
| 1NF | A relation is in 1NF if it contains an atomic value.                                                                    |
| 2NF | A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key. |
| 3NF | A relation will be in 3NF if it is in 2NF and no transition dependency exists.                                          |

|     |                                                                                                           |
|-----|-----------------------------------------------------------------------------------------------------------|
| 4NF | A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.          |
| 5NF | A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless. |

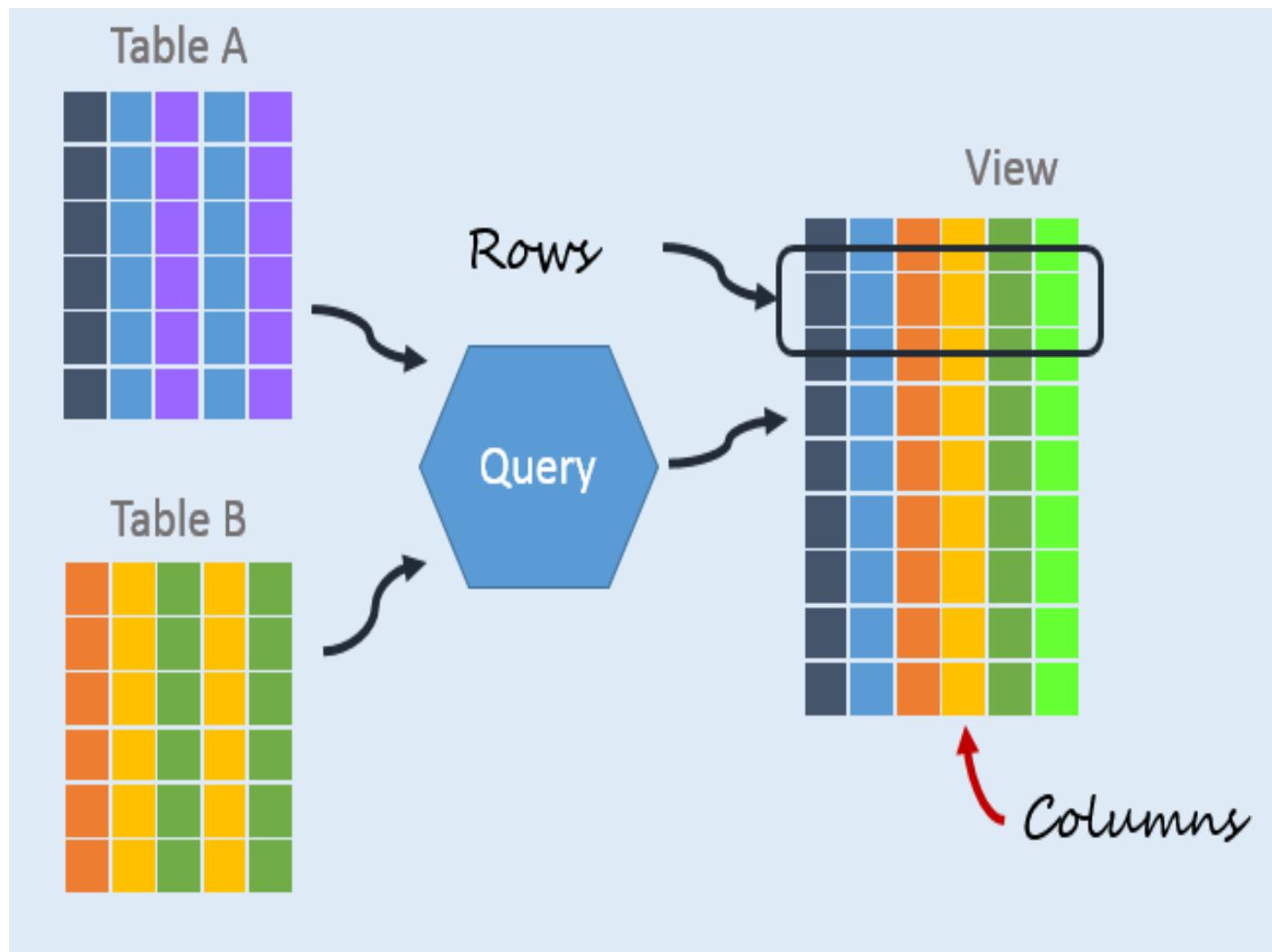
[More in Detail](#)

---

82

## WHAT IS A VIEW?

- + A view is a virtual table which consists of a subset of data contained in a table.
- + Views are not virtually present, and it takes less space to store.
- + View can have data of one or more tables combined, and it is depending on the relationship.



83

## HOW TO CREATE CLUSTERED AND NON-CLUSTERED INDEX IN A TABLE?

When you create a PRIMARY KEY constraint, a unique clustered index on the column or columns is automatically created if a clustered index on the table does not already exist and you do not specify a unique nonclustered index.

`CREATE CLUSTERED INDEX <index_name>`

`ON <table_name>(<column_name> ASC/DESC)`

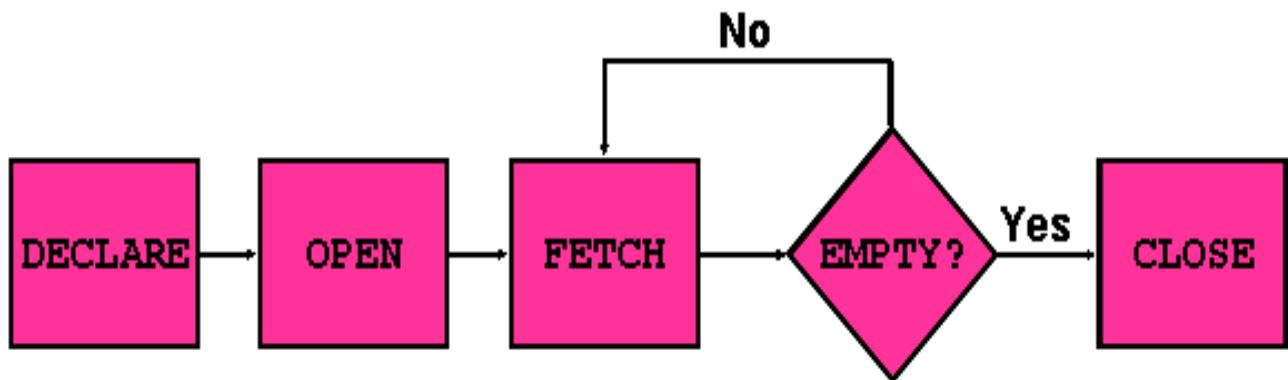
`CREATE NONCLUSTERED INDEX <index_name>`

`ON <table_name>(<column_name> ASC/DESC)`

## WHAT IS A CURSOR? WHY TO AVOID THEM?

A database Cursor is a control which enables traversal over the rows or records in the table.

This can be viewed as a pointer to one row in a set of rows.



- It is creating a memory Area.
- Identify the active set from a Sql Query
- Load the current Information into variables.
- It is check for existing rows.
- Release the active set.
- Return to FETCH if rows are found.

### LIMITATIONS:

A cursor is a memory resident set of pointers -- meaning it occupies memory from your system that may be available for other processes.

---

85

## WHAT IS THE DIFFERENCE BETWEEN DELETE AND TRUNCATE COMMANDS?

DELETE command is used to remove rows from the table, and WHERE clause can be used for conditional set of parameters. Commit and Rollback can be performed after delete statement.

TRUNCATE removes all rows from the table. Truncate operation cannot be rolled back.

---

86

## WHAT IS THE DIFFERENCE BETWEEN SCOPE\_IDENTITY AND @@IDENTITY?

SCOPE\_IDENTITY and @@IDENTITY will return the last identity values generated in any table in the current session. However, SCOPE\_IDENTITY returns values inserted only within the current scope; @@IDENTITY is not limited to a specific scope.

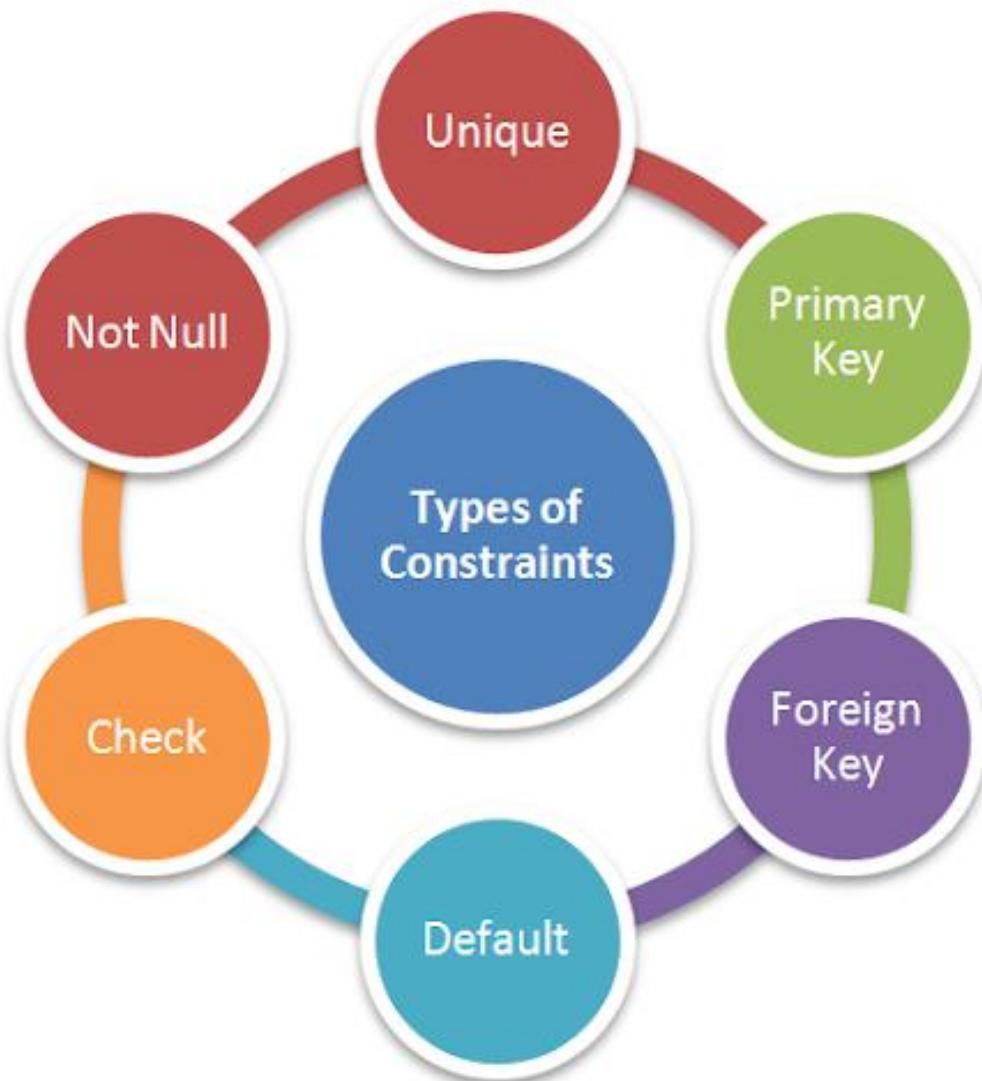
A scope is a module; a Stored Procedure, trigger, function, or batch.

---

87

## WHAT IS A CONSTRAINT?

 Constraint can be used to specify the limit on the data type of table. Constraint can be specified while creating or altering the table statement.



1. NOT NULL - Ensures that a column cannot have a NULL value
2. UNIQUE - Ensures that all values in a column are different
3. PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
4. FOREIGN KEY - Uniquely identifies a row/record in another table

5. CHECK - Ensures that all values in a column satisfies a specific condition
  6. DEFAULT - Sets a default value for a column when no value is specified
- 

88

## WHAT IS SELF-JOIN?

Self-join is set to be query used to compare to itself. This is used to compare values in a column with other values in the same column in the same table. ALIAS ES can be used for the same table comparison.

```
SELECT A.CustomerName AS CustomerName1, B.CustomerName
AS CustomerName2, A.City
FROM Customers A, Customers B
WHERE A.CustomerID <> B.CustomerID
AND A.City = B.City
ORDER BY A.City;
```

[More in Detail](#)

## EXPLAIN TRIGGERS AND TYPES OF TRIGGERS?

Triggers are stored programs, which are automatically executed or fired when some events occur.

The following are the various types of triggers.

**DML** - In SQL Server we can create triggers on DML statements (like INSERT, UPDATE and DELETE) and Stored Procedures that do DML-like operations. DML Triggers are of two types.

- Instead of Trigger: An Instead of trigger is fired instead of the triggering action such as an insert, update, or delete
- After Trigger: An After trigger executes following the triggering action, such as an insert, update or delete

### DDL Trigger

This type of trigger is fired against DDL statements like Drop Table, Create Table or Alter Table. DDL Triggers are always After Triggers.

### Logon trigger

This type of trigger is fired against a LOGON event before a user session is established to the SQL Server.

## More in Detail

---

90

### WHAT IS THE DIFFERENCE BETWEEN HAVING CLAUSE AND WHERE CLAUSE?

- a) WHERE Clause is used to filter the records from the table based on the specified condition.
  - b) HAVING Clause is used to filter record from the groups based on the specified condition.
- 
- a) WHERE Clause can be used without GROUP BY Clause.
  - b) HAVING Clause implements in column operation.
- 
- a) WHERE Clause cannot contain aggregate function.
  - b) HAVING Clause can contain aggregate function.
- 
- a) WHERE Clause is used before GROUP BY Clause.
  - b) HAVING Clause is used after GROUP BY Clause.

---

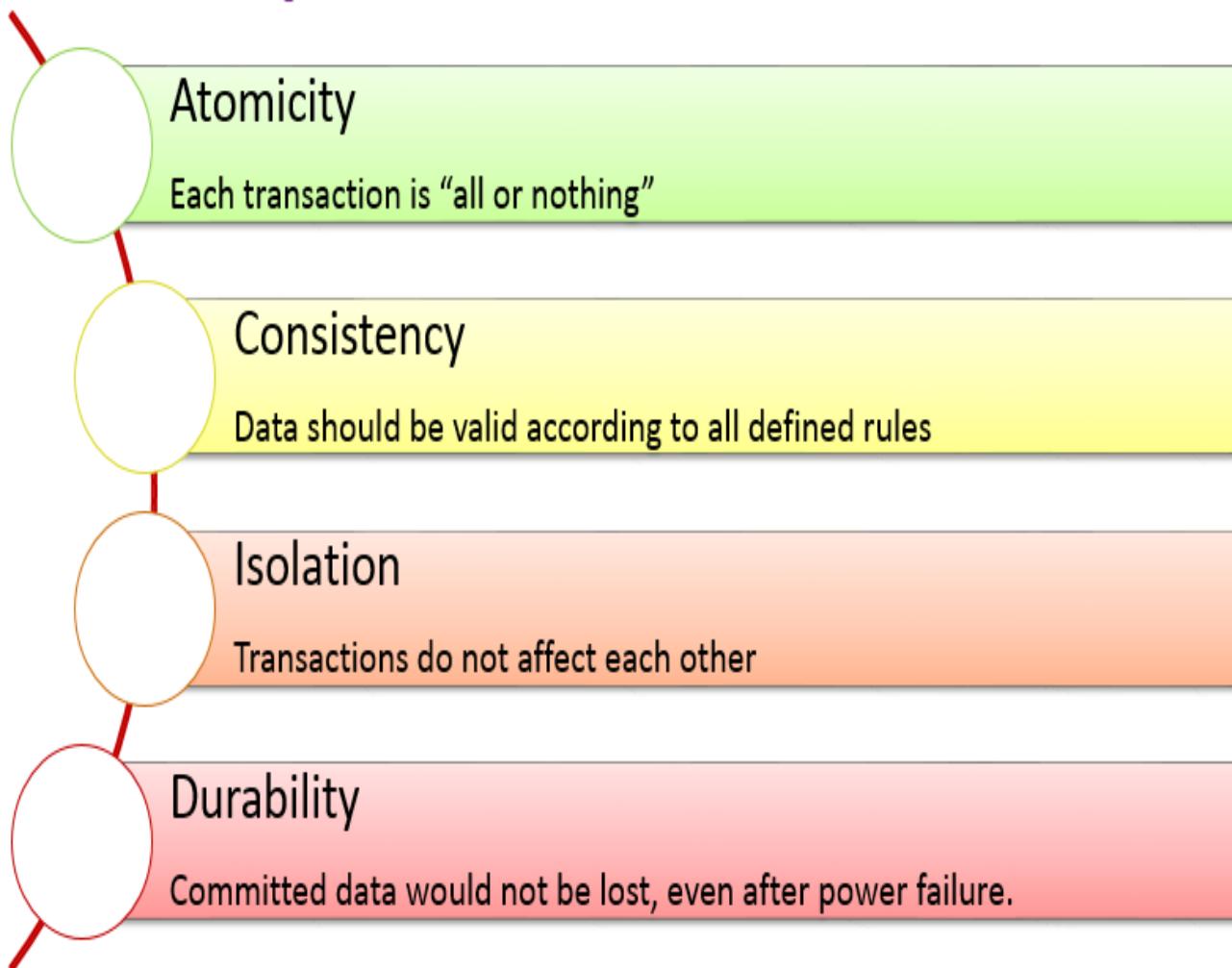
91

## IS THERE ANY DIFFERENCE BETWEEN PRIMARY KEY AND UNIQUE KEY?

There is no difference between primary key and unique key but, a unique key will allow single NULL, but in the primary key, no NULL is accepted.

---

## WHAT ARE ACID PROPERTIES



[More in Detail](#)

## WHAT IS CTE IN SQL?

A Common Table Expression, also called as CTE in short form, is a temporary named result set that you can reference within a SELECT, INSERT, UPDATE, or DELETE statement. The CTE can also be used in a View.

```
WITH
with engineers as (
 select *
 from employees
 where dept='Engineering'
)
select *
from engineers ← CTE Usage
where ...
```

CTE name  
CTE Body  
CTE Usage

[More in Detail](#)



## #RESUME TIP 10

### USE PDF FORMAT

Now a day's companies use software's to scan and shortlist resumes so it is better to use pdf's rather than any other format.

# Chapter 8 - DESIGN PATTERNS

94

## WHAT ARE DESIGN PATTERNS AND WHAT PROBLEM THEY SOLVE?

- + A design pattern is a general reusable solution to a commonly occurring problem within a given context in software design.
- 

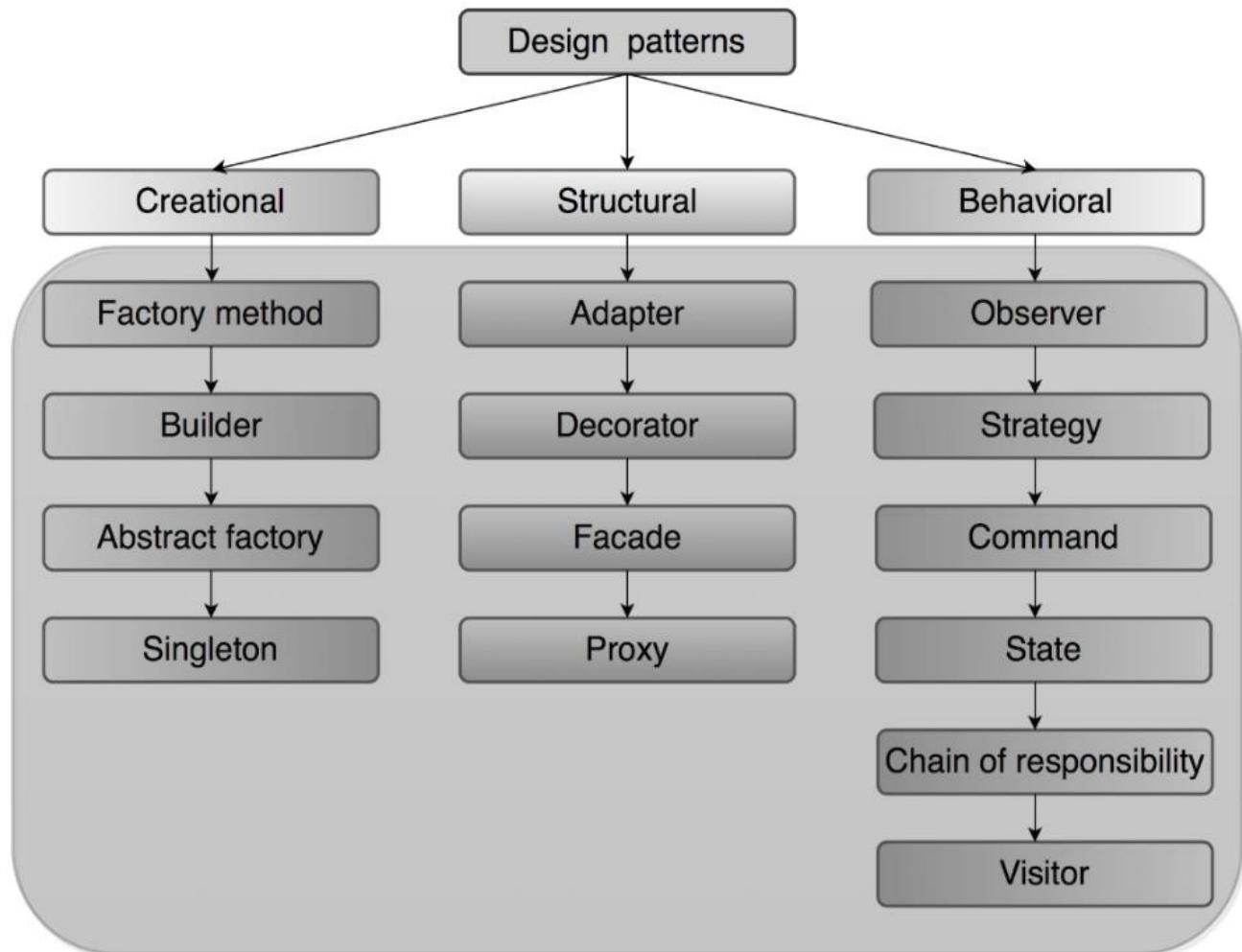
95

## WHAT ALL ARE THE TYPES OF DESIGN PATTERNS?

There are 3 types of Design Pattern.

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

 Behavioral Patterns - This type of pattern address problems of assigning responsibilities to classes.



[\*\*More in Detail\*\*](#)

## WHAT IS SINGLETON DESIGN PATTERN?

A singleton is a class that only allows a single instance of itself to be created and usually gives simple access to that instance.

There are many ways to implement a Singleton Pattern in C#.

1. No Thread Safe Singleton.
2. Thread-Safety Singleton.
3. Thread-Safety Singleton using Double-Check Locking.
4. Thread-Safe Singleton without using locks and no lazy instantiation.
5. Fully lazy instantiation.

Example of thread safety Singleton:

```
public sealed class Singleton {
 private static Singleton instance = null;
 private static readonly object padlock = new object();
```

## Singleton()

```
{
}
public static Singleton Instance
{
 get
 {
 lock (padlock)
 {
 if (instance == null)
 {
 instance = new Singleton();
 }
 return instance;
 }
 }
}
```

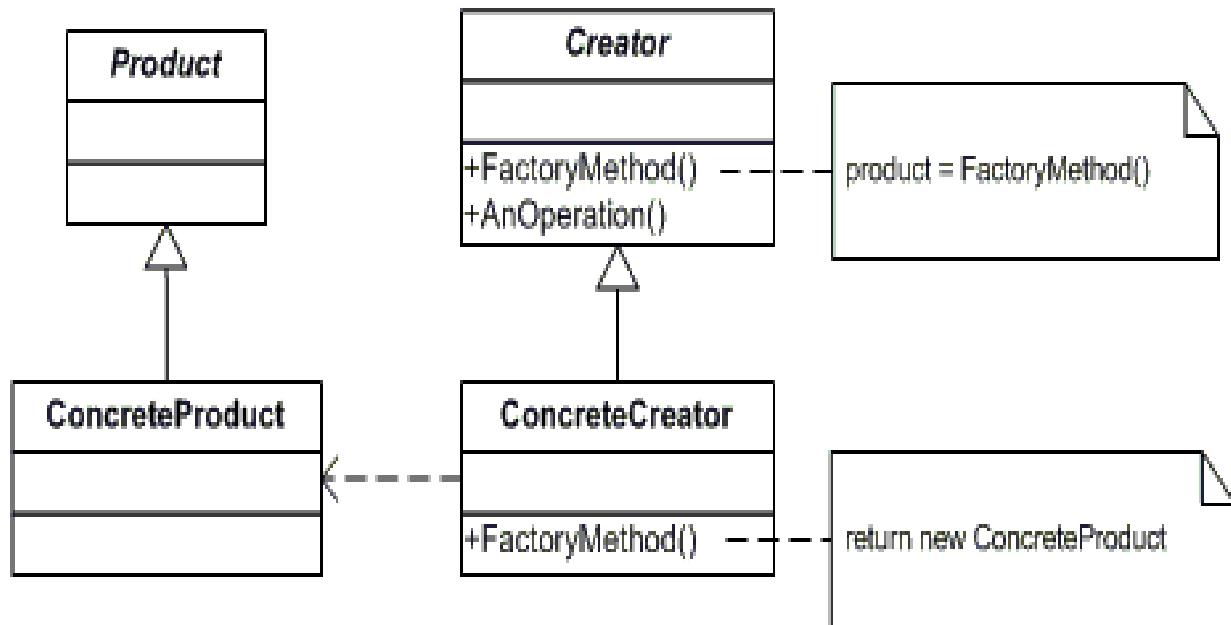
}

[\*\*More in Detail\*\*](#)

---

## WHAT IS FACTORY DESIGN PATTERN?

Factory Design patterns define an Interface for creating an object but let subclasses decide which class to instantiate.



[More in Detail](#)

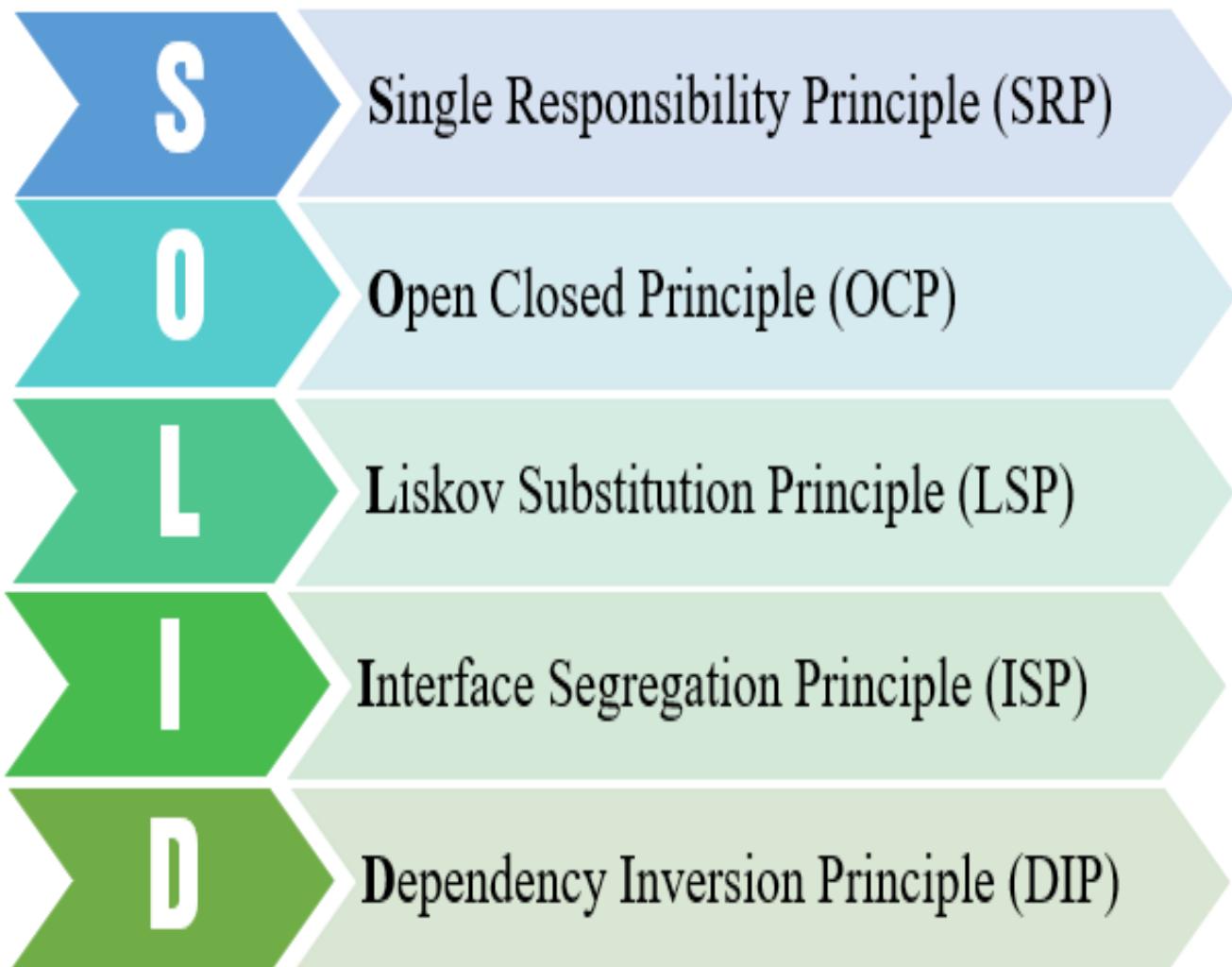
## WHAT IS ABSTRACT FACTORY DESIGN PATTERN?

Abstract Factory Design patterns provides an interface for creating families of related or dependent objects without specifying their concrete classes.

[More in Detail](#)

---

## WHAT ARE SOLID PRINCIPLES?



[More in Detail](#)

100

## EXPLAIN DEPENDENCY INJECTION?

 Dependency injection is a programming technique that makes a class independent of its dependencies. It achieves that by decoupling the usage of an object from its creation. This helps you to follow SOLID's dependency inversion and single responsibility principles.

[More in Detail](#)

---

## PART II

MORE  
INTERVIEW  
QUESTIONS...

# OOPS

## 1. WHAT IS OOPS?

- OOPs (Object-Oriented Programming) is a type of programming which is based on objects rather than just functions and procedures.

Individual objects are grouped into classes. OOPs implements real-world entities like inheritance, polymorphism, hiding, etc into programming. It also allows binding data and code together.

---

## 2. WHAT IS STRUCTURAL PROGRAMMING (SOP)?

- Structural programming refers to the traditional method of programming, which is based on functions.

The overall program logic is divided into functions to provide a logical structure. It is based on a top-down approach. Structural programming is suitable for easy to moderately complex problems.

---

## 3. WHAT IS THE DIFFERENCE BETWEEN OOP AND SOP?

- a) In procedural programming(SOP), program is divided into small parts called *functions*.
  - b) In object oriented programming(OOP), program is divided into small parts called *objects*.
- 
- a) Procedural programming follows *top down approach*.
  - b) Object oriented programming follows *bottom up approach*.
- 

## 4.WHY USE OOPS?

- + OOPs allows clarity in programming thereby allowing simplicity in solving complex problems
  - + Code can be reused through inheritance thereby reducing redundancy
  - + Data and code are bound together by encapsulation
  - + OOPs allows data hiding, therefore, private data is kept confidential
  - + Problems can be divided into different parts making it simple to solve
  - + The concept of polymorphism gives flexibility to the program by allowing the entities to have multiple forms
-

## 5.WHAT IS AN OBJECT?

- An object is a real-world entity which is the basic unit of OOPs for example chair, cat, dog, etc. Different objects have different states or attributes, and behaviors.
- 

## 6.WHAT IS A CLASS?

- A class is a prototype that consists of objects in different states and with different behaviors. It has a number of methods that are common the objects present within that class.
- 

## 7.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.

---

## 8.WHAT IS THE DIFFERENCE BETWEEN A CLASS AND AN OBJECT?

- a) Class is a blueprint or template from which objects are created.
  - b) Object is an instance of a class.
- 
- a) Class is a group of similar objects.
  - b) Object is a real world entity such as pen, laptop, mobile, bed, keyboard, mouse, chair etc.
- 
- a) Class is a logical entity.
  - b) Object is a physical entity.

---

## 9.WHAT IS INHERITANCE?

 Inheritance is a feature of OOPs which allows classes inherit common properties from other classes.  
For example, if there is a class such as 'vehicle', other classes like 'car', 'bike', etc can inherit common properties from the vehicle class. This property helps you get rid of redundant code thereby reducing the overall size of the code.

---

## **10. WHAT IS THE DIFFERENCE BETWEEN MULTIPLE AND MULTILEVEL INHERITANCE?**

- a) Multiple Inheritance is an Inheritance type where a class inherits from more than one base class.
  - b) Multilevel Inheritance is an Inheritance type that inherits from a derived class, making that derived class a base class for a new class.
- 

## **11. WHAT IS THE MEANING OF HIERARCHICAL INHERITANCE?**

When multiple subclasses inherit a base class, it is called hierarchical inheritance.

---

## **12. WHAT IS HYBRID INHERITANCE?**

Hybrid inheritance is a combination of multiple and multi-level inheritance.

---

## 13. WHAT ARE THE LIMITATIONS OF INHERITANCE?

- Increases the time and effort required to execute a program as it requires jumping back and forth between different classes
- The parent class and the child class get tightly coupled
- Any modifications to the program would require changes both in the parent as well as the child class
- Needs careful implementation else would lead to incorrect results

---

## 14. WHAT IS A SUPERCLASS?

A superclass or base class is a class that acts as a parent to some other class or classes. For example, the Vehicle class is a superclass of class Car.

---

## 15. WHAT IS A SUBCLASS?

A class that inherits from another class is called the subclass. For example, the class Car is a subclass or a derived of Vehicle class.

---

## 16. WHAT IS POLYMORPHISM?

Polymorphism refers to the ability to exist in multiple forms. Multiple definitions can be given to a single interface. For example, if you have a class named Vehicle, it can have a method named speed but you cannot define it because different vehicles have different speed. This method will be defined in the subclasses with different definitions for different vehicles.

---

## 17. WHAT IS STATIC POLYMORPHISM?

Static polymorphism (static binding) is a kind of polymorphism that occurs at compile time. An example of compile-time polymorphism is method overloading.

---

## 18. WHAT IS DYNAMIC POLYMORPHISM?

Runtime polymorphism or dynamic polymorphism (dynamic binding) is a type of polymorphism which is resolved during runtime. An example of runtime polymorphism is method overriding.

---

## 19. WHAT IS METHOD OVERLOADING?

Method overloading is a feature of OOPs which makes it possible to give the same name to more than one methods within a class if the arguments passed differ.

---

## 20. WHAT IS METHOD OVERRIDING?

Method overriding is a feature of OOPs by which the child class or the subclass can redefine methods present in the base class or parent class. Here, the method that is overridden has the same name as well as the signature meaning the arguments passed and the return type.

---

## 21. WHAT IS OPERATOR OVERLOADING?

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

---

## 22. WHAT IS ENCAPSULATION?

Encapsulation refers to binding the data and the code that works on that together in a single unit. For example, a class. Encapsulation also allows data-hiding as the data specified in one class is hidden from other classes.

---

## 23. WHAT ARE 'ACCESS SPECIFIERS'?

Access specifiers or access modifiers are keywords that determine the accessibility of methods, classes, etc in OOPs. These access specifiers allow the implementation of encapsulation. The most common access specifiers are public, private and protected. However, there are a few more which are specific to the programming languages.

---

## 24. WHAT IS DATA ABSTRACTION?

Data abstraction is a very important feature of OOPs that allows displaying only the important information and hiding the implementation details.

---

## 25. WHAT IS AN ABSTRACT CLASS?

An abstract class is a class that consists of abstract methods. These methods are basically declared but not defined. If these methods are to be used in some subclass, they need to be exclusively defined in the subclass.

## 26. HOW TO ACHIEVE DATA ABSTRACTION?

Data abstraction can be achieved through:

- Abstract class.
  - Abstract method.
- 

## 27. 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.

---

## 28. CAN YOU CREATE AN INSTANCE OF AN ABSTRACT CLASS?

No.

---

## 29. WHAT IS AN INTERFACE?

It is a concept of OOPs that allows you to declare methods without defining them. Interfaces, unlike classes, are not blueprints because they do not contain detailed instructions or actions to be performed. Any class that implements an interface defines the methods of the interface.

---

## 30. WHAT ARE 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.

---

## 31. WHAT ARE PURE VIRTUAL FUNCTIONS?

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.

---

## 32. WHAT IS A CONSTRUCTOR?

A constructor is a special type of method that has the same name as the class and is used to initialize objects of that class.

---

## 33. WHAT IS A DESTRUCTOR?

A destructor is a method that is automatically invoked when an object is destroyed. The destructor also recovers the heap space that was allocated to the destroyed object, closes the files and database connections of the object, etc.

---

## 34. WHAT IS A COPY CONSTRUCTOR?

A copy constructor creates objects by copying variables from another object of the same class. The main aim of a copy constructor is to create a new object from an existing one.

---

### 35. DO WE REQUIRE A PARAMETER FOR CONSTRUCTORS?

No, we do not need a parameter for constructors.

---

### 36. WHEN TO USE PRIVATE CONSTRUCTOR?

Private constructor provides an implementation of a singleton class pattern or when a class have only static methods.

---

### 37. WHAT IS THE DEFAULT ACCESS MODIFIER IN A CLASS?

The default access modifier of a class is Private by default.

---

## **38. WHICH OOPS CONCEPT EXPOSES ONLY THE NECESSARY INFORMATION TO THE CALLING FUNCTIONS?**

Encapsulation

---

## **39. 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.

---

## **40. 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.

---

## **41. EXPLAIN SEALED CLASS?**

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

---

## 42. WHAT ARE THE LIMITATIONS OF OOPS?

- Usually not suitable for small problems
  - Requires intensive testing
  - Takes more time to solve the problem
  - Requires proper planning
  - The programmer should think of solving a problem in terms of objects
- 

## 43. WHICH OOPS CONCEPT IS USED AS A REUSE MECHANISM?

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

---

# C#

## 1.WHAT IS C-SHARP (C#)?

C# is a type-safe, managed and object oriented language, which is compiled by .Net framework for generating intermediate language (IL).

---

## 2.EXPLAIN THE FEATURES OF C#?

Below are some of the features supported in C# -

- Constructors and Destructors
  - Properties
  - Passing Parameters
  - Arrays
  - Main
  - XML Documentation and
  - Indexers
- 

## 3.WHAT ARE THE LOOP TYPES IN C#?

Below are the loop types -

For

While

Do.. While

---

## 4. WHAT IS THE DIFFERENCE BETWEEN CONTINUE AND BREAK STATEMENT?

Break statement breaks the loop. It makes the control of the program to exit the loop. Continue statement makes the control of the program to exit only the current iteration. It does not break the loop.

---

## 5. CAN WE EXECUTE MULTIPLE CATCH BLOCKS?

No. Once any exception is occurred it executes specific exception catch block and the control comes out.

---

## 6. WHAT IS THE DIFFERENCE BETWEEN “FINALIZE” AND “FINALLY” METHODS?

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

---

## 7. 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.

---

## 8. 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.

---

## 9. CAN WE HAVE ONLY “TRY” BLOCK WITHOUT “CATCH” BLOCK?

Yes, we can have only try block without catch block but we have to have finally block.

```
try {
```

```
 // statements
```

```
}
```

```
finally
```

```
{
```

```
// statements to be executed
```

```
}
```

---

## 10. LIST OUT SOME OF THE EXCEPTIONS?

Below are some of the exceptions -

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

---

## 11. EXPLAIN HASHTABLE?

It is used to store the key/value pairs based on hash code of the key. Key will be used to access the element in the collection. For example,

```
Hashtable myHashtbl = new Hashtable();
myHashtbl.Add("1", "TestValue1");
myHashtbl.Add("2", "TestValue2");
```

---

## 12. HOW TO CHECK WHETHER HASH TABLE CONTAINS SPECIFIC KEY?

Method – “ContainsKey” can be used to check the key in hash table. Below is the sample code for the same –

Eg: myHashtbl.ContainsKey("1");

---

## 13. WHAT IS THE DIFFERENCE BETWEEN LIST AND DICTIONARY?

List collection is a generic class and can store any data types to create a list. It only contains value not key.

Dictionary is a collection of keys and values in C#. Dictionary< TKey, TValue> is included in the System.Collection.Generics namespace.

---

## 14. 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.

---

## 15. 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.

---

## 16. 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.

---

## 17. 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;
 }
 }
}
```

```
}
```

---

## 18. 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.
- 

## 19. EXPLAIN NAMESPACES?

Namespaces are containers for the classes. We will use namespaces for grouping the related classes. “Using” keyword can be used for using the namespace in other namespace.

---

## 20. WHAT ARE VALUE TYPE?

Value types stored in a stack. Here are the value types:

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

## 21. WHAT ARE REFERENCE TYPES?

Below are the list of reference types -

- Class, string, interface, object
- 

## 22. CAN WE OVERRIDE PRIVATE VIRTUAL METHOD?

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

---

## 23. EXPLAIN ACCESS MODIFIER – “PROTECTED INTERNAL”?

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

---

## 24. 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.

---

## 25. 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.

---

## 26. WHAT ARE THE TYPES OF DELEGATES?

Below are the uses of delegates -

- Single Delegate
  - Multicast Delegate
  - Generic Delegate
- 

## 27. WHAT ARE THE THREE TYPES OF GENERIC DELEGATES?

Below are the three types of generic delegates -

- Func

- Action
  - Predicate
- 

## 28. CAN WE USE DELEGATES FOR ASYNCHRONOUS METHOD CALLS?

Yes. We can use delegates for asynchronous method calls.

---

## 29. WHAT ARE THE DIFFERENCES BETWEEN EVENTS AND DELEGATES?

- a) Delegate is a function pointer. It holds the reference of one or more methods at runtime.
  - b) The event is a notification mechanism that depends on delegates.
- 
- a) A delegate is declared using the delegate keyword.
  - b) An event is declared using the event keyword.
- 
- a) Delegate is independent and not dependent on events.

b) An event is dependent on a delegate and cannot be created without delegates.

a) A delegate can be passed as a method parameter.

b) An event is raised but cannot be passed as a method parameter.

---

## 30. DEFINE MULTICAST DELEGATE?

A **Multicast Delegate** is a delegate that holds the references of more than one function. When we invoke the multicast delegate, then all the functions which are referenced by the delegate are going to be invoked. If you want to call multiple methods using a delegate then all the method signature should be the same.

The example to demonstrate the same is given below

```
public delegate void CalculateNumbers(int x, int y);
int x = 6;
int y = 7;
CalculateNumbers addNumbers = new
CalculateNumbers(FuncForAddingNumbers);
CalculateNumbers multiplyNumbers = new
CalculateNumbers(FuncForMultiplyingNumbers);
CalculateNumbers multiCast =
```

```
(CalculateNumbers)Delegate.Combine (addNumbers,
multiplyNumbers);
multiCast.Invoke(a,b)
```

---

## 31. WHAT ARE THE USES OF DELEGATES?

Below are the list of uses of delegates -

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

## 32. WHAT IS NULLABLE TYPES?

Variable types does not hold null values so to hold the null values we have to use nullable types. So nullable types can have values either null or other values as well.

Eg: Int? mynullablevar = null;

---

### **33. IS C# CODE IS UNMANAGED OR MANAGED CODE?**

C# code is managed code because the compiler – CLR will compile the code to Intermediate Language.

---

### **34. 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.

---

### **35. 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.

---

### **36. 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.

---

### 37. WHY TO USE KEYWORD “CONST”?

“Const” keyword is used for making an entity constant. We can’t reassign the value to constant.

---

### 38. 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();
```

---

### 39. 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
```

---

## 40. WHAT IS TERNARY OPERATOR?

The conditional operator ?:, also known as the ternary conditional operator, evaluates a Boolean expression and returns the result of one of the two expressions, depending on whether the Boolean expression evaluates to true or false.

```
condition ? consequent : alternative
```

---

## 41. CAN WE USE “THIS” INSIDE A STATIC METHOD?

No. We can't use “this” in static method.

---

## 42. 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.
- 

### 43. 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)
```

---

### 44. 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);
```

---

## 45. 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.

---

## 46. 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"};
```

---

## 47. 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.

---

## 48. WHAT ARE EXTENSION METHODS IN C#?

Extension methods allow existing classes to be extended without relying on inheritance or having to change the class's source code.

---

## 49. 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";
```

---

## 50. WHAT IS THE NAMESPACE USED FOR FILE HANDLING IN C#?

System.IO

---

## 51. WHAT IS COVARIANCE IN C#?

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

```
Small sml = new Bigger();
```

---

## 52. WHAT IS THE PURPOSE OF PARAMS KEYWORD?

Params is used as a parameter which can take the variable number of arguments. It is useful when programmer don't have any prior knowledge about the number of parameters to be used.

Example:

```
// function containing params parameters
public static int Add(params int[]
ListNumbers)
{

}
```

---

## .NET FRAMEWORK

### 1. WHAT IS THE .NET FRAMEWORK?

It is a platform for building various applications on windows. It has a list of inbuilt functionalities in the form of class, library, and APIs

which are used to build, deploy and run web services and different applications. It supports different languages such as C#, VB .Net, Cobol, Perl, etc.

This framework supports the object-oriented programming model.

---

## 2.WHAT IS CTS?

CTS stands for **Common Type System**. It has a set of rules which state how a data type should be declared, defined and used in the program. It describes the data types that are to be used in the application.

We can design our own classes and values by following the rules that are present in the CTS. The rules are made so that the data type declared using a programming language can be called by an application that is developed using a different language.

---

## 3.WHAT IS CLS?

CLS stands for **Common Language Specification**. With the rules mentioned under CLS, the developers are made to use the

components that are inter-language compatible. They are reusable across all the .Net Compliant languages.

---

## 4.WHAT IS CLR?

CLR stands for **Common Language Runtime**. It is one of the most important components of the .Net framework. It provides building blocks for many applications.

An application built using C# gets compiled by its own compiler and is converted into an Intermediate language. This is then targeted to CLR. CLR does various operations like memory management, security checks, assemblies to be loaded and thread management. It provides a secure execution environment for applications.

---

## 5.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.

---

## 6.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.

---

## 7.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.

---

## 8. WHAT IS AN ASSEMBLY?

- Assembly is unit of deployment like EXE or a DLL.
- 

## 9. 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.
- 

## 10. 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.
- 

## 11. WHERE IS VERSION INFORMATION STORED OF AN ASSEMBLY?

Version information is stored in assembly inside the manifest.

---

## 12. 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.

---

## 13. 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.

---

## 14. 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.

---

## 15. WHAT ARE GENERATIONS AND HOW ARE THEY USED BY THE GARBAGE COLLECTOR?

Basically the generation of Garbage Collection (GC) shows the life of objects, it means it defines how long an object will stay in the memory. It's categorized into the following three generations:

- Generation 0
- Generation 1
- Generation 2

---

## 16. WHAT IS THE USE OF 'FINALIZE'?

Finalize as an object method used to free up unmanaged resources and cleanup before Garbage Collection(GC). It performs memory management tasks.

---

(THREADING)

## 17. 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.

---

## 18. 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.

---

## 19. 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.

---

## 20. 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.
- 

## 21. 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.

---

## 22. WHICH NAMESPACE HAS THREADING?

‘Systems.Threading’ has all the classes related to implement threading.

---

## 23. 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);}

## 24. HOW CAN YOU REFERENCE CURRENT THREAD OF THE METHOD?

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

---

## 25. 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).

---

## 26. WHAT IS SUSPEND AND RESUME IN THREADING?

It is Similar to Sleep and Interrupt Suspend allows you to block a thread until another

Thread calls Thread.Resume. The difference between Sleep and Suspend is that the latter does not immediately place a thread in the wait state. The thread does not suspend until the .NET runtime determines that it is in a safe place to suspend it. Sleep will immediately place a thread in a wait state.

---

## 27. WHAT THE WAY TO STOP A LONG RUNNING THREAD?

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

---

## 28. 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.
- 

## 29. 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 that we have to call the start method of Thread class. Below is the example.

---

## 30. WHAT IS RACE CONDITION?

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

---

## 31. 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.

---

## 32. 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.

---

## 33. 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.

---

## 34. 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.

---

## 35. 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

## 1. IN WHICH ASSEMBLY IS THE MVC FRAMEWORK DEFINED?

System.Web.Mvc

---

## 2. DEFINE CONTROLLER IN MVC?

The controller provides model data to the view, and interprets user actions such as button clicks. The controller depends on the view and the model. In some cases, the controller and the view are the same object.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace DemoMVC.Controllers
{
 public class EmployeeController : Controller
 {
 public string EmployeeNames()
 {
 return @"
 Sourabh Soman
 Shaili Dashora
 Saloni Choudhry
 Mahesh Chand
 DJ
 Dinesh Beniwal
 ";
 }
 }
}
```

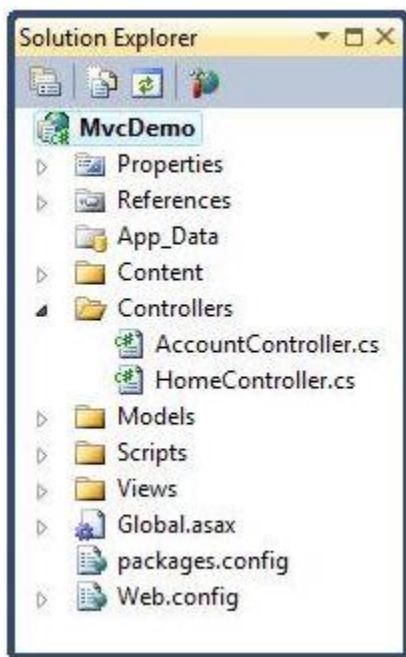
**Controller**

**Action**

## The Controllers Folder

The Controllers Folder contains the controller classes responsible for handling user input and responses. MVC requires the name of all controllers to end with "Controller".

In our example, Visual Web Developer has created the following files: HomeController.cs (for the Home and About pages) and AccountController.cs (For the Log On pages):



### 3. EXPLAIN MODEL IN MVC?

The model represents the data, and does nothing else. The model does NOT depend on the controller or the view. The MVC Model contains all application logic (business logic, validation logic, and data access logic), except pure view and controller logic. With MVC, models both hold and manipulate application data.

### The Models Folder

The Models Folder contains the classes that represent the application model.

Visual Web Developer automatically creates an AccountModels.cs file that contains the models for application security.

---

## 4. EXPLAIN VIEW IN MVC?

A view is responsible for displaying all of, or a portion of, data for users. In simple terms, whatever we see on the output screen is a view.

### The Views Folder

The Views folder stores the files (HTML files) related to the display of the application (the user interfaces). These files may have the extensions html, asp, aspx, cshtml, and vbhtml, depending on the language content.

The Views folder contains one folder for each controller. Visual Web Developer has created an Account folder, a Home folder, and a Shared folder (inside the Views folder). The Account folder contains pages for registering and logging in to user accounts. The Home folder is used for storing application pages like the home page and the about page. The Shared folder is used to store views shared between controllers (master pages and layout pages).

---

## 5.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.

---

## 6.WHAT ARE THE FILTERS IN MVC?

Many times we would like to perform some action before or after a particular operation. For achieving this functionality, ASP.NET MVC provides a feature to add pre and post-action behaviors on the controller's action methods.

---

## 7.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}");

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

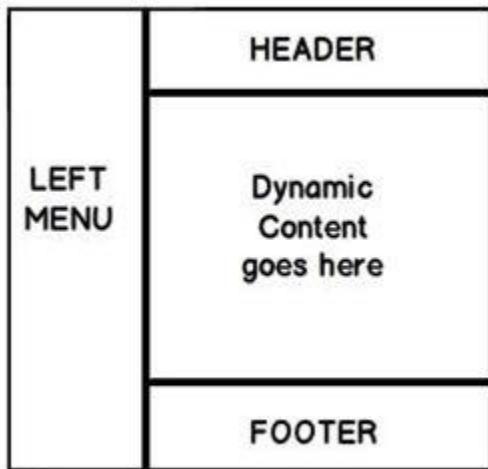
## 8.WHERE ARE THE ROUTING RULES DEFINED IN AN ASP.NET MVC APPLICATION?

In `Application_Start` event in `Global.asax`

## 9.WHAT IS PARTIAL VIEW IN MVC?

A partial view is a chunk of HTML that can be safely inserted into an existing DOM. Most commonly, partial views are used to componentize Razor views and make them easier to build and update. Partial views can also be returned directly from controller methods. In this case, the browser still receives text/html content but not necessarily HTML content that makes up an entire page. As a result, if a URL that returns a partial view is directly invoked from the address bar of a browser, an incomplete page may be displayed. This may be something like a page that misses title, script and style sheets. However, when the same URL is invoked via a script, and the response is used to insert HTML within the existing DOM, then the net effect for the end-user may be much better and nicer.

Partial view is a reusable view (like a user control) which can be embedded inside another view. For example, let's say all the pages of your site have a standard structure with left menu, header, and footer as in the following image,



```

@model IEnumerable<MvcApplication8.Models.Product>

@{
 ViewBag.Title = "Index";
}

Products

Name	Description	Price	EDD
@item.Name	@item.Description	@(item.Price/10)	Edit Details Delete


```

Let's make this piece of code reusable

---

## 10. WHAT IS TEMPDATA IN MVC?

TempData is a dictionary object to store data temporarily. It is a TempDataDictionary class type and instance property of the Controller base class.

TempData is able to keep data for the duration of a HTP request, in other words it can keep live data between two consecutive HTTP requests. It will help us to pass the state between action methods. TempData only works with the current and subsequent request. TempData uses a session variable to store the data. TempData Requires type casting when used to retrieve data.

TempDataDictionary is inherited from the IDictionary<string, object>, ICollection<KeyValuePair<string, object>>, IEnumerable<KeyValuePair<string, object>> and IEnumerable interfaces.

### Example

1. **public** ActionResult FirstRequest()
2. {
3.   List <**string**> TempDataTest = **new** List <**string**>();

```
4. TempDataTest.Add("Tejas");
5. TempDataTest.Add("Jignesh");
6. TempDataTest.Add("Rakesh");
7. TempData["EmpName"] = TempDataTest;
8. return View();
9. }
10. public ActionResult ConsecutiveRequest()
11. {
12. List < string > modelData = TempData["EmpName"]
 as List < string >;
13. TempData.Keep();
14. return View(modelData);
15. }
```

---

## 11. 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.

---

## 12. WHAT IS GET AND POST ACTIONS TYPES?

### GET

GET is used to request data from a specified resource. With all the GET request we pass the URL which is compulsory, however it can take the following overloads.

```
.get(url [, data] [, success(data, textStatus, jqXHR)] [,
dataType]).done/.fail
```

### POST

POST is used to submit data to be processed to a specified resource. With all the POST requests we pass the URL which is compulsory and the data, however it can take the following overloads.

```
.post(url [, data] [, success(data, textStatus, jqXHR)] [,
dataType])
```

---

## 13. WHAT IS RAZOR IN MVC?

Razor is one of the view engines supported in ASP.NET MVC. Razor allows you to write a mix of HTML and server-side code

using C# or Visual Basic. Razor view with visual basic syntax has .vbhtml file extension and C# syntax has .cshtml file extension.

---

## 14. 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.

---

## 15. 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

## C# Example

```

1. <!-- Single statement block -->
2. @ {
3. varmyMessage = "Hello World";
4. }
5. <!-- Inline expression or variable -->
6. <p> The value of myMessage is: @myMessage </p>
7. <!-- Multi-statement block -->
8. @ {
9. var greeting = "Welcome to our site!";
10. varweekDay = DateTime.Now.DayOfWeek;
11. vargreetingMessage = greeting + " Here in Huston
 it is: " + weekDay;
12. } <p> The greeting is: @greetingMessage </p>

```

## 16. 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,

1. <system.web>
2.   <authenticationmode="Forms">
3.     <formsloginUrl="Login.aspx" protection="All" timeOut="30" name=".ASPXAUTH" path="/" requireSSL="false" slidingExpiration="true" defaultUrl="default.aspx" cookieless="UseDeviceProfile" enableCrossAppRedirects="false"/>
4.     </authentication>
5. </system.web>

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

---

## 17. 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();
}
```

---

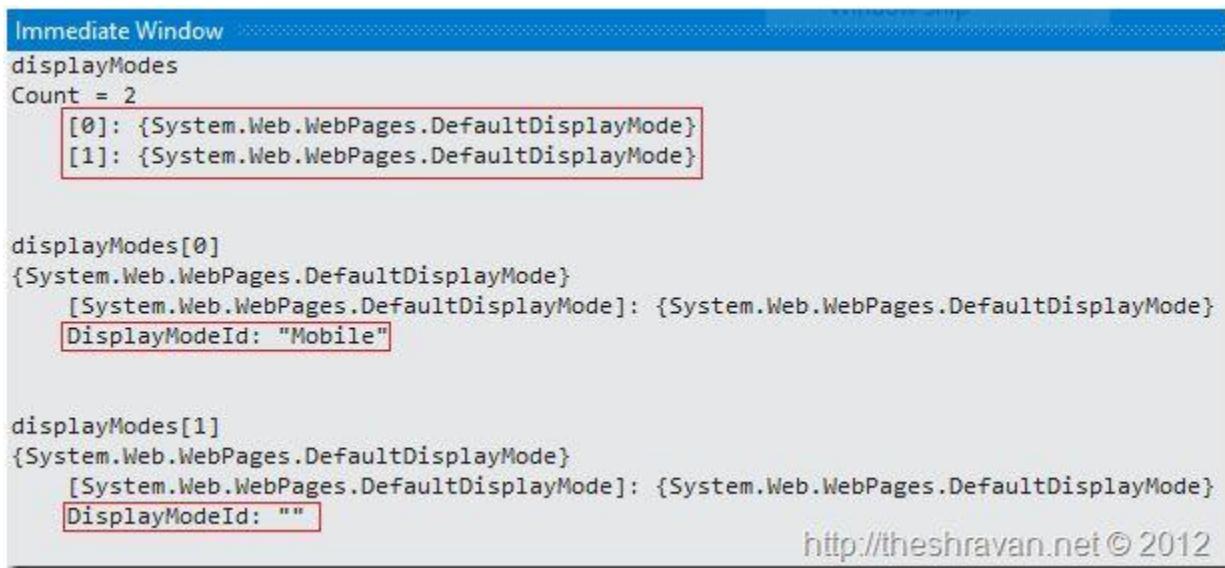
## 18. EXPLAIN THE NEED OF DISPLAY MODE IN MVC?

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.



The screenshot shows the Immediate Window of a debugger. It displays the variable `displayModes` which has a `Count = 2`. The elements `[0]` and `[1]` are both instances of `{System.Web.WebPages.DefaultDisplayMode}`. The `DisplayModeId` for `[0]` is highlighted and shows the value `"Mobile"`. The `DisplayModeId` for `[1]` is also highlighted and shows the value `""`.

```
Immediate Window
displayModes
Count = 2
[0]: {System.Web.WebPages.DefaultDisplayMode}
[1]: {System.Web.WebPages.DefaultDisplayMode}

displayModes[0]
{System.Web.WebPages.DefaultDisplayMode}
[System.Web.WebPages.DefaultDisplayMode]: {System.Web.WebPages.DefaultDisplayMode}
DisplayModeId: "Mobile"

displayModes[1]
{System.Web.WebPages.DefaultDisplayMode}
[System.Web.WebPages.DefaultDisplayMode]: {System.Web.WebPages.DefaultDisplayMode}
DisplayModeId: ""
```

<http://theshravan.net> © 2012

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.

---

## 19. 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

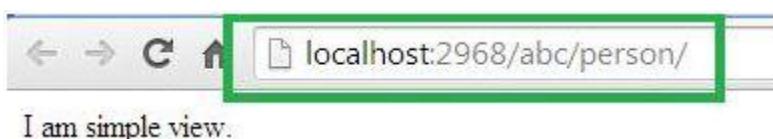
This is very necessary for when we want to add a specific constraint to our URL. Say, for example we want a [URL](#).

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 "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:



I am simple view.

---

## 20. WHAT ARE THE FOLDERS IN MVC APPLICATION SOLUTIONS?

# **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.

---

## **21. 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**

```
1. [HandleError(View = "Error")]
2. public class HomeController: Controller
3. {
4. public ActionResult Index()
5. {
6. ViewBag.Message = "Welcome to ASP.NET MVC!";
7. int u = Convert.ToInt32(""); // Error line
8. return View();
9. }
10. }
```

HandleError Attribute at Action Method Level,

```
1. [HandleError(View = "Error")]
2. public ActionResult Index()
3. {
4. ViewBag.Message = "Welcome to ASP.NET MVC!";
5. int u = Convert.ToInt32(""); // Error line
6. return View();
7. }
```

---

## 22. WHAT IS VIEWSTART?

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

```
1. @ {
2. Layout = "~/Views/Shared/_v1.cshtml";
3. }<!DOCTYPE html>
4. <html>
5. <head>
6. <meta name = "viewport"
7. content = "width=device-width" />
8. <title>ViewStart</title></head><body>
9. </body></html>
```

---

## 23. WHAT IS JSONRESULTTYPE IN MVC?

Action methods on controllers return JsonResult (JavaScript Object Notation result) that can be used in an AJAX application. This class is inherited from the "ActionResult" abstract class. Here Json is provided one argument which must be serializable. The JSON result object that serializes the specified object to JSON format.

```
1. public JsonResult JsonResultTest()
2. {
3. return Json("Hello My Friend!");
4. }
```

---

## 24. WHAT IS TEMPDATA?

### Tempdata

- TempData is a dictionary object derived from the TempDataDictionary class.
- TempData is used to pass data from the current request to a subsequent request, in other words in the case of redirection.
- The life of a TempData is very short and it retains its value for a short period of time.
- It requires typecasting for complex data type as I've used in my example:

```
@foreach (var item in
(List<MVCSample.Models.EmpRegistration>)TempData["Empl
oyeeRegistration"])
```
- You can retain its value using the Keep method for subsequent requests.

```

public ActionResult Verify()
{
 TempData["EmployeeRegistration"] = ObjEmp.GetEmpRegistrationsDetails();
 TempData.Keep("EmployeeRegistration");
 return View("Verify", TempData["EmployeeRegistration"]);
}

#region Commented Code
//return View("Verify", ViewData);
#endregion Commented Code

}

//
// GET: /Register/Details/5

```

The screenshot shows the Visual Studio debugger's watch window. A tooltip for the TempData variable is displayed, showing its structure. It has a Count of 1, with one entry under Keys. This entry has a Count of 1. Under Values, there is a list with a Count of 5. Each item in the list is labeled '[0]' through '[4]', and each points to the same MVCSample.Models.EmpRegistration object.

## 25. HOW TO USE VIEWBAG?

ViewBag is dynamic property that takes advantage of new dynamic features in C# 4.0. It's also used to pass data from a controller to a view. In short, The ViewBag property is simply a wrapper around the ViewData that exposes the ViewData dictionary as a dynamic object. Now create an action method "StudentSummary" in the "DisplayDataController" controller that stores a Student class object in ViewBag.

1. **public** ActionResult StudentSummary()
2. {
3. var student = **new** Student()
4. {
5. Name = "Sandeep Singh Shekhawat",

```
6. Age = 24,
7. City = "Jaipur"
8. };
9. ViewBag.Student = student;
10. return View();
11. }
```

Thereafter create a view StudentSummary ("StudentSummary.cshtml") that shows student object data. ViewBag does not require typecasting for complex data type so you can directly access the data from ViewBag.

```
1. @ {
2. ViewBag.Title = "Student Summary";
3. var student = ViewBag.Student;
4. }
5. < table >
6. < tr >
7. < th > Name </th> < th > Age </th> < th > City </th> </tr>
8. < tr >
9. < td > @student.Name </td> < td > @student.Age </td><
10. td > @student.City </td> </tr>
11. </table>
```

Here we used one more thing, "ViewBag.Title", that shows the title of the page.

---

## 26. 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 attribute.

1. [HandleError]
2. **public class** HomeController: Controller
3. {
4. [HandleError]
5. public ActionResult ThrowException()
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://www.w3.org/1999/xhtml">
<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.   public ActionResult ThrowException()

```
6. {
7. throw new ApplicationException();
8. }
9. }
```

---

## 27. 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 i
 s 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 valid");
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. {
11. Name: @ViewBag.Name

12. Email: @ViewBag.Email
13. }
14. } <fieldset>
15. <legend>User </legend> <div class = "editor-
16. label" >
17. @Html.LabelFor(model => model.Name) </div> <di
18. v class = "editor-field" >
19. @Html.EditorFor(model => model.Name)
20. @if(!ViewData.ModelState.IsValid)
21. {
22. <span class = "field-validation-
23. error" >@ViewData.ModelState["Name"].Errors[0].ErrorM
24. essage
25. }
26.
```

```
23. </div> <div class = "editor-label" >
24.
25. @Html.LabelFor(model => model.Email) </div> <div
26. class = "editor-field" >
27. @Html.EditorFor(model => model.Email)
28. @if(!ViewData.ModelState.IsValid)
29. {
30. <span class = "field-validation-
31. error" > @ViewData.ModelState["Email"].Errors[0].Error
32. Message
33. }
34. </div> <p >
35. <input type = "submit"
36. value = "Create" />
37. </p> </fieldset>
38. }
```

---

## 28. 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, whether 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
 e = "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 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.com) ? true : false;
7. return Json(!ifEmailExist, JsonRequestBehavior.AllowGet);
8. } catch (Exception ex)
9. {
10. return Json(false, JsonRequestBehavior.AllowGet);
11. }
12. }
```

---

## 29. WHAT IS ATTRIBUTE ROUTING IN MVC?

A route attribute is defined on top of an action method. The following is the example of a Route Attribute in which routing is defined where the action method is defined.

In the following example, I am defining the route attribute on top of the action method

```
1. public class HomeController: Controller
2. {
3. //URL: /Mvctest
4. [Route("Mvctest")]
5. public ActionResult Index()
6. ViewBag.Message = "Welcome to ASP.NET MVC!";
7. return View();
8. }
9. }
```

### Attribute Routing with Optional Parameter

We can also define an optional parameter in the URL pattern by defining a mark ("?") to the route parameter. We can also define the default value by using parameter=value.

```
1. public class HomeController: Controller
2. {
3. // Optional URI Parameter
4. // URL: /Mvctest/
5. // URL: /Mvctest/0023654
6. [Route("Mvctest /
7. {
8. customerName ?
9. }")]
10. public ActionResult OtherTest(string customerN
ame)
11. ViewBag.Message = "Welcome to ASP.NET MVC!
";
12. return View();
13. }
14. // Optional URI Parameter with default v
alue
15. // URL: /Mvctest/
16. // URL: /Mvctest/0023654
17. [Route("Mvctest /
18. {
19. customerName = 0036952
20. }")]
```

```
21. public ActionResult OtherTest(string customerName)
22. {
23. ViewBag.Message = "Welcome to ASP.NET MVC!";
24. return View();
25. }
26. }
```

---

## 30. 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>
```

---

## 31. 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.

---

## 32. 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.

---

### **33. 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.
- 

### **34. 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.

---

### **35. 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, whereas to add routes to an MVC application we use MapRoute() method.

---

## **36. 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
- 

## **37. WHICH FILTER EXECUTES FIRST IN AN ASP.NET MVC APPLICATION?**

Authorization filter

---

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

1. Action Method
2. Controller
3. Application

---

## 39. IS IT POSSIBLE TO CREATE A CUSTOM FILTER?

Yes

---

## 40. WHAT FILTERS ARE EXECUTED IN THE END?

Exception Filters

---

## 41. IS IT POSSIBLE TO CANCEL FILTER EXECUTION?

Yes

---

## 42. WHAT TYPE OF FILTER DOES OUTPUTCACHEATTRIBUTE CLASS REPRESENTS?

Result Filter

---

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

1. Razor
  2. .aspx
- 

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

@

---

## 45. 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

---

## 46. WHAT ARE SECTIONS?

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

---

## 47. WHAT ARE THE FILE EXTENSIONS FOR RAZOR VIEWS?

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

## 48. HOW DO YOU SPECIFY COMMENTS USING RAZOR SYNTAX?

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

---

## 49. 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.

---

## 50. WHAT IS SEPARATION OF CONCERNS IN ASP.NET MVC?

It's is the process of breaking the program into various distinct features which overlaps in functionality as little as possible. MVC pattern concerns on separating the content from presentation and data-processing from content.

---

## 51. 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.

---

## 52. 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.

---

### 53. 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)
{
}
}
```

---

### 54. 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.

---

## 55. 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”);
```

---

## 56. 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();
```

---

## 57. 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.

---

## 58. 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

---

## 59. 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>
```

---

## 60. 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 !!");
}
```

---

## 61. 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" });
}
```

---

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

1. RequireHttpsAttribute
  2. AuthorizeAttribute
- 

## 63. WHAT IS THE DIFFERENCE BETWEEN VIEWBAG & VIEWDATA?

ViewData is a dictionary of objects that is derived from ViewDataDictionary class and accessible using strings as keys.

ViewBag is a dynamic property that takes advantage of the new dynamic features in C# 4.0. ViewData requires typecasting for complex data type and check for null values to avoid error.

ViewBag doesn't require typecasting for complex data type.

---

## 64. 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; }
```

---

## 65. 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.

---

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

ASP.NET MVC

---

## 67. WHAT IS ENTITYFRAMEWORK?

Entity Framework is an object-relational mapper (O/RM) that enables .NET developers to work with a database using .NET objects. It eliminates the need for most of the data-access code that developers usually need to write.

---

# ASP.NET WEBFORMS

## 1.WHAT IS ASP.NET?

It is a framework developed by Microsoft on which we can develop new generation web sites using web forms(aspx), MVC, HTML, Javascript, CSS etc. Its successor of Microsoft Active Server Pages(ASP). Currently there is ASP.NET 4.0, which is used to develop web sites. There are various page extensions provided by Microsoft that are being used for web site development. Eg: aspx, asmx, ascx, ashx, cs, [vb](#), html, XML etc.

---

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

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

---

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

After the Init() and before the Page\_Load().

---

#### **4.FROM WHICH BASE CLASS ALL WEB FORMS ARE INHERITED?**

Page class.

---

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

Compare Validator control.

---

#### **6.WHAT IS VIEWSTATE?**

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

---

#### **7.HOW LONG THE ITEMS IN VIEWSTATE EXISTS?**

They exist for the life of the current page.

---

## **8.WHAT ARE THE DIFFERENT SESSION STATE MANAGEMENT OPTIONS AVAILABLE IN ASP.NET?**

- 1. In-Process**
- 2. Out-of-Process.**

In-Process stores the session in memory on the web server.

Out-of-Process Session state management stores data in an external server. The external server may be either a SQL Server or a State Server. All objects stored in session are required to be serializable for Out-of-Process state management.

---

## **9.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**
-

## 10. HOW YOU CAN ADD AN EVENT HANDLER?

Using the Attributes property of server side control.

e.g.

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

---

## 11. 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"%>
```

---

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

Yes

## 13. 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.

---

## 14. WHAT IS PROTECTED CONFIGURATION?

It is a feature used to secure connection string information.

---

## 15. 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.

---

## 16. 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.ToString());
```

---

## 17. 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.

---

## 18. 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

---

## 19. WHICH PROTOCOL IS USED TO CALL A WEB SERVICE?

HTTP Protocol

---

## 20. 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 >
```

---

## 21. 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>

---

## 22. 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.

---

## 23. 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

---

## 24. 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.

---

## 25. 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
- 

## 26. HOW DO YOU REGISTER JAVASCRIPT FOR WEBCONTROLS ?

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

---

## 27. 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.

---

## 28. LIST ALL TEMPLATES OF THE REPEATER CONTROL.

- ItemTemplate
  - AlternatingItemTemplate
  - SeparatorTemplate
  - HeaderTemplate
  - FooterTemplate
- 

## 29. LIST THE MAJOR BUILT-IN OBJECTS IN ASP.NET?

- Application
- Request

- Response
  - Server
  - Session
  - Context
  - Trace
- 

## 30. 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>
```

---

## 31. WHICH DATA TYPE DOES THE RANGEVALIDATOR CONTROL SUPPORT?

The data types supported by the RangeValidator control are Integer, Double, String, Currency, and Date.

---

## 32. WHAT IS THE DIFFERENCE BETWEEN AN HTMLINPUTCHECKBOX CONTROL AND AN HTMLINPUTRADIOBUTTON CONTROL?

In HtmlInputCheckBoxcontrol, multiple item selection is possible whereas in HtmlInputRadioButton controls, we can select only single item from the group of items.

---

## 33. WHICH NAMESPACES ARE NECESSARY TO CREATE A LOCALIZED APPLICATION?

System.Globalization

System.Resources

---

## **34. 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.

---

## **35. WHAT IS THE FILE EXTENSION OF WEB SERVICE?**

Web services have file extension .asmx.

---

## **36. WHAT IS SESSION LESS COOKIE.**

We use Session in ASP.NET application to maintain the state of the user. In Cookie-less Sessions, the values that are required to associate users with their sessions are appended to the browser's URL.

---

## **37. 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

---

### 38. 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.

---

### 39. 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.

---

## 40. 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

## 1. WHAT IS AN ADO.NET?

ADO.Net is commonly termed as ActiveX Data Objects which is a part of .Net Framework. ADO.Net framework has set of classes which are used to handle data access by connecting with different databases like SQL, Access, Oracle, etc...

---

## 2. WHAT ARE THE NAMESPACES USED IN ADO.NET TO CONNECT TO A DATABASE?

Following namespaces are used to connect to Database.

- The System.Data namespace.
  - The System.Data.OleDb namespace – A data provider used to access database such as Access, Oracle, or SQL.
  - The System.Data.SqlClient namespace – Used to access SQL as the data provider.
-

### **3. WHAT IS DATAREADER OBJECT?**

Datareader is an object of ADO.Net which provides access to data from a specified data source. It consists of classes which sequentially read data from a data source like Oracle, SQL or Access.

---

### **4. WHAT IS DATASET OBJECT?**

A Dataset is set to be collection of data with a tabular column representation. Each column in the table represents a variable and the row represents to value of a variable. This Dataset object can be obtained from the database values.

---

### **5. 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.

---

## **6. 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.

---

## **7. WHAT IS DATA VIEW?**

Data view is the representation of data in various formats and it can be requested by the users. Data can be exposed in different sort orders or filter on the user condition with the help of Data view. Data Customization is also possible through Data View.

---

## **8. WHAT IS DATA ADAPTER?**

Data Adapter is a part of ADO.NET data provider which acts as a communicator between Dataset and the Data source. This Data adapter can perform Select, Insert, Update and Delete operations in the requested data source.

---

## **9. WHAT IS THE USE OF SQLCOMMAND OBJECT?**

SQLCommand object that allows user to interact with the database. This object mainly used to query the database and it can be of different types – Select, Insert, Modify and Delete.

---

## **10. WHAT IS THE DIFFERENCE BETWEEN ADO AND ADO.NET?**

ADO works with the connected data whereas ADO.Net works in a disconnected manner. ADO has main object called Recordset which is used to reference data. But ADO.Net has various objects to access the database.

ADO allows creating client side cursors whereas ADO.Net deals with both server side and server side cursors. ADO allows persisting records in XML format and ADO.Net allows to manipulate data using XML.

---

## **11. WHAT ARE THE BENEFITS OF ADO.NET?**

Following are the benefits of ADO.Net:

- Programmability
- Maintainability

- Interoperability
  - Performance
  - Scalability
- 

## 12. IS IT POSSIBLE TO EDIT DATA IN REPEATER CONTROL?

No, it is not possible to edit data in the Repeater control.

---

## 13. WHAT ARE THE DIFFERENCES BETWEEN OLEDB AND SQLCLIENT PROVIDERS?

OLEDB provider is used to access any database and provides flexibility of changing the database at any time. SQLClient provider is used to access only SQL Server database but it provides excellent performance than OLEDB provider while connecting with SQL Server database.

---

## 14. WHAT IS THE DIFFERENCE BETWEEN DATASET.CLONE AND DATASET.COPY?

Dataset.clone object copies structure of the dataset including schemas, relations and constraints. This will not copy data in the table.

Dataset.copy – Copies both structure and data from the table.

---

## 15. 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.

---

## 16. WHAT ARE THE METHODS OF XML DATASET OBJECT?

There are various methods of XML dataset object:

`GetXml()` – Get XML data in a Dataset as a single string.

`GetXmlSchema()` – Get XSD Schema in a Dataset as a single string.

`ReadXml()` – Reads XML data from a file.

`ReadXmlSchema()` – Reads XML schema from a file.

`WriteXml()` – Writes the contents of Dataset to a file.

`WriteXmlSchema()` – Writes XSD Schema into a file.

---

## 17. WHICH KEYWORD IS USED TO ACCEPT VARIABLE NUMBER OF PARAMETERS?

`Params` keyword is used to accept variable number of parameters.

---

## 18. WHICH METHOD IN OLEDBADAPTER IS USED TO POPULATE DATASET WITH RECORDS?

`Fill` Method is used to populate dataset with records.

---

## 19. WHICH OBJECT NEEDS TO BE CLOSED?

`OLEDBReader` and `OLEDBConnection` object need to be closed. This will stay in memory if it is not properly closed.

---

## **20. WHAT ARE DIFFERENT LAYERS OF ADO.NET?**

There are three different layers of ADO.Net:

- Presentation Layer
  - Business Logic Layer
  - Database Access Layer
- 

## **21. WHICH OBJECT IS USED TO ADD RELATIONSHIP BETWEEN TWO DATATABLES?**

`DataRelation` object is used to add relationship between two or more datatable objects.

---

## **22. 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.
- 

## 23. WHAT ARE THE DATA PROVIDERS IN ADO.NET?

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

- MS SQL Server.
  - OLEDB.
  - ODBC.
- 

## 24. 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.

---

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

ExecuteScalar

---

# JAVASCRIPT

## 1.WHAT IS JAVASCRIPT?

**JavaScript** is a *scripting language*. It is different from Java language. It is object-based, lightweight, cross-platform translated language. It is widely used for client-side validation. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

---

## 2.LIST SOME FEATURES OF JAVASCRIPT.

Some of the features of JavaScript are:

- Lightweight
- Interpreted programming language
- Good for the applications which are network-centric
- Complementary to Java
- Complementary to HTML
- Open source

Cross-platform

---

### **3.LIST SOME OF THE ADVANTAGES OF JAVASCRIPT.**

Some of the advantages of JavaScript are:

- Server interaction is less
  - Feedback to the visitors is immediate
  - Interactivity is high
  - Interfaces are richer
- 

### **4.LIST SOME OF THE DISADVANTAGES OF JAVASCRIPT.**

Some of the disadvantages of JavaScript are:

- No support for multithreading
  - No support for multiprocessing
  - Reading and writing of files is not allowed
  - No support for networking applications.
- 

### **5.NAME THE TYPES OF FUNCTIONS IN JAVASCRIPT?**

The types of function are:

- Named - These type of functions contains name at the time of definition. For Example:

```
function display()
```

```
{
 document.writeln("Named Function");
}
display();
```

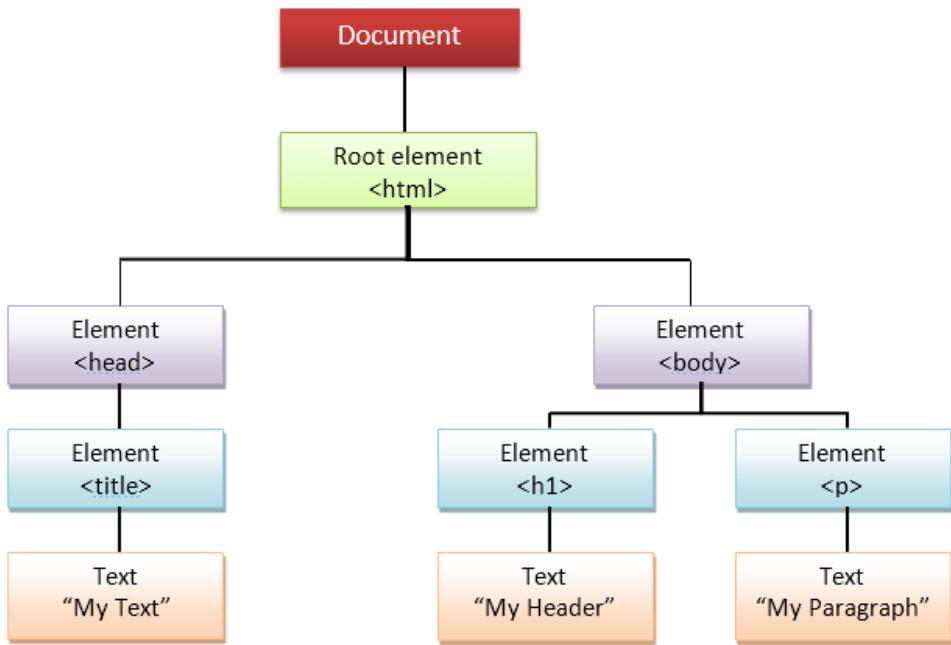
- **Anonymous** - These type of functions doesn't contain any name. They are declared dynamically at runtime.

```
var display=function()
{
 document.writeln("Anonymous Function");
}
display();
```

---

## 6.WHAT IS DOM? WHAT IS THE USE OF DOCUMENT OBJECT?

**DOM** stands for *Document Object Model*. A document object represents the HTML document. It can be used to access and change the content of HTML.



## 7.WHAT IS THE DIFFERENCE BETWEEN == AND ===?

The == operator checks equality only whereas === checks equality, and data type, i.e., a value must be of the same type.

## 8.HOW TO WRITE HTML CODE DYNAMICALLY USING JAVASCRIPT?

The innerHTML property is used to write the HTML code using JavaScript dynamically. Let's see a simple example:

```
document.getElementById('mylocation').innerHTML=<h2>This is heading using JavaScript</h2>;
```

---

## 9. HOW TO CREATE OBJECTS IN JAVASCRIPT?

There are 3 ways to create an object in JavaScript.

1. By object literal
2. By creating an instance of Object
3. By Object Constructor

Let's see a simple code to create an object using object literal.

`emp={id:102,name:"Rahul Kumar",salary:50000}`

---

## 10. WHAT DOES THE ISNAN() FUNCTION?

The isNaN() function returns true if the variable value is not a number. For example:

---

## 11. HOW TO HANDLE EXCEPTIONS IN JAVASCRIPT?

By the help of try/catch block, we can handle exceptions in JavaScript. JavaScript supports try, catch, finally and throw keywords for exception handling.

```
try {
 Block of code to try
}
catch(err) {
 Block of code to handle errors
}
finally {
 Block of code to be executed regardless of the
 try / catch result
}
```

---

## 12. WHAT IS THIS KEYWORD IN JAVASCRIPT?

The `this` keyword is a reference variable that refers to the current object.

For example:

```
var person = {
 firstName: "John",
 lastName : "Doe",
 id : 5566,
 fullName : function() {
 return this.firstName + " " + this.lastName;
 }
};
```

---

## 13. WHAT IS THE USE OF DEBUGGER KEYWORD IN JAVASCRIPT?

JavaScript debugger keyword sets the breakpoint through the code itself. The debugger stops the execution of the program at the position it is applied. Now, we can start the flow of execution manually. If an exception occurs, the execution will stop again on that particular line.. For example:

```
var x = 15 * 5;
debugger;
document.getElementById("demo").innerHTML = x;
```

---

## 14. WHAT IS THE USE OF MATH OBJECT IN JAVASCRIPT?

The JavaScript math object provides several constants and methods to perform a mathematical operation. Unlike date object, it doesn't have constructors.

For example:

```
Math.abs(-4.7); // returns 4.7
Math.ceil(4.4); // returns 5
```

```
Math.random(); // returns a random number
```

---

## 15. DEFINE A NAMED FUNCTION IN JAVASCRIPT.

The function which has named at the time of definition is called a named function. For example

1. function msg()
  2. {
  3. document.writeln("Named Function");
  4. }
  5. msg();
- 

## 16. WHAT IS THE USE OF WINDOW OBJECT?

The window object is created automatically by the browser that represents a window of a browser. It is not an object of JavaScript. It is a browser object.

The window object is used to display the popup dialog box. Let's see with description.

Method	Description

alert()	displays the alert box containing the message with ok button.
confirm()	displays the confirm dialog box containing the message and cancel button.
prompt()	displays a dialog box to get input from the user.
open()	opens the new window.
close()	closes the current window.
setTimeout()	performs the action after specified time like calling function or evaluating expressions.

## 17. DEFINE ANONYMOUS FUNCTION

It is a function that has no name. These functions are declared dynamically at runtime using the function operator instead of the function declaration. The function operator is more flexible than a function declaration. It can be easily used in the place of an expression. For example:

1. var **display**=function()
2. {

```
3. alert("Anonymous Function is invoked");
4.
5. display();
```

---

## 18. CAN AN ANONYMOUS FUNCTION BE ASSIGNED TO A VARIABLE?

Yes, you can assign an anonymous function to a variable.

---

## 19. IN JAVASCRIPT WHAT IS AN ARGUMENT OBJECT?

The variables of JavaScript represent the arguments that are passed to a function.

---

## 20. DEFINE CLOSURE.

In JavaScript, we need closures when a variable which is defined outside the scope in reference is accessed from some inner scope.

1. var num = 10;
2. function sum()

```
3. {
4. document.writeln(num+num);
5. }
6. sum();
```

---

## 21. IF WE WANT TO RETURN THE CHARACTER FROM A SPECIFIC INDEX WHICH METHOD IS USED?

The JavaScript string `charAt()` method is used to find out a character value present at the specified index. The index number starts from 0 and goes to  $n-1$ , where  $n$  is the length of the string. The index value can't be a negative, greater than or equal to the length of the string. For example:

```
1. var str="Javatpoint";
2. document.writeln(str.charAt(4));
```

---

## 22. WHAT IS THE DIFFERENCE BETWEEN JAVASCRIPT AND JSCRIPT?

Netscape provided the JavaScript language. Microsoft changed the name and called it JScript to avoid the trademark issue. In

other words, you can say JScript is the same as JavaScript, but Microsoft provides it.

---

## 23. HOW TO WRITE A HELLO WORLD EXAMPLE OF JAVASCRIPT?

A simple example of JavaScript hello world is given below. You need to place it inside the body tag of HTML.

1. **<script type="text/javascript">**
  2. **document.write("JavaScript Hello World!");**
  3. **</script>**
- 

## 24. HOW TO USE EXTERNAL JAVASCRIPT FILE?

I am assuming that js file name is message.js, place the following script tag inside the head tag.

1. **<script type="text/javascript" src="message.js"></script>**
- 

## 25. IS JAVASCRIPT CASE SENSITIVE LANGUAGE?

Yes, JavaScript is a case sensitive language. For example:

```
1. Var msg = "JavaScript is a case-
sensitive language"; //Here, var should be used to declare a v
ariable
2. function display()
3. {
4. document.writeln(msg); // It will not display the result.
5. }
6. display();
```

---

## 26. WHAT IS THE USE OF HISTORY OBJECT?

The history object of a browser can be used to switch to history pages such as back and forward from the current page or another page. There are three methods of history object.

1. history.back() - It loads the previous page.
  2. history.forward() - It loads the next page.
  3. history.go(number) - The number may be positive for forward, negative for backward. It loads the given page number.
- 

## 27. HOW TO WRITE A COMMENT IN JAVASCRIPT?

There are two types of comments in JavaScript.

1. Single Line Comment: It is represented by // (double forward slash)
  2. Multi-Line Comment: Slash represents it with asterisk symbol as /\* write comment here \*/
- 

## 28. HOW TO CREATE A FUNCTION IN JAVASCRIPT?

To create a function in JavaScript, follow the following syntax.

1. `function function_name(){`
  2. `//function body`
  3. `}`
- 

## 29. WHAT ARE THE JAVASCRIPT DATA TYPES?

There are two types of data types in JavaScript:

1. Primitive Data Types - The primitive data types are as follows:

Data Type	Description
String	represents a sequence of characters, e.g., "hello"
Number	represents numeric values, e.g., 100

Boolean	represents boolean value either false or true
Undefined	represents an undefined value
Null	represents null, i.e., no value at all

2. Non-primitive Data Types - The non-primitive data types are as follows:

Data Type	Description
Object	represents an instance through which we can access
Array	represents a group of similar values
RegExp	represents regular expression

## 30. HOW TO WRITE NORMAL TEXT CODE USING JAVASCRIPT DYNAMICALLY?

The innerText property is used to write the simple text using JavaScript dynamically. Let's see a simple example:

1. `document.getElementById('mylocation').innerText="This is te  
xt using JavaScript";`

---

## 31. HOW TO CREATE AN ARRAY IN JAVASCRIPT?

There are 3 ways to create an array in JavaScript.

1. By array literal
2. By creating an instance of Array
3. By using an Array constructor

Let's see a simple code to create an array using object literal.

1. `var emp=["Shyam","Vimal","Ratan"];`

---

## 32. WHAT ARE THE POP-UP BOXES AVAILABLE IN JAVASCRIPT?

- Alert Box
- Confirm Box
- Prompt Box

---

## 33. HOW TO SUBMIT A FORM USING JAVASCRIPT BY CLICKING A LINK?

Let's see the JavaScript code to submit the form by clicking the link.

1. <form name="myform" action="index.php">
2. Search: <input type='text' name='query' />
3. <a href="javascript: submitform()">Search</a>
4. </form>
5. <script type="text/javascript">
6. function submitform()
7. {
8. document.myform.submit();
9. }
10. </script>

---

### 34. WHAT IS THE ROLE OF A STRICT MODE IN JAVASCRIPT?

The JavaScript strict mode is used to generates silent errors. It provides "use strict"; expression to enable the strict mode. This expression can only be placed as the first statement in a script or a function. For example:

1. "use strict";
  2. x=10;
  3. console.log(x);
-

# SQL FOR DEVELOPER

## 1. WHAT IS DBMS?

A Database Management System (DBMS) is a program that controls creation, maintenance and use of a database. DBMS can be termed as File Manager that manages data in a database rather than saving it in file systems.

## 2.WHAT IS RDBMS?

RDBMS stands for Relational Database Management System. RDBMS store the data into the collection of tables, which is related by common fields between the columns of the table. It also provides relational operators to manipulate the data stored into the tables.

**Example: SQL Server.**

---

## 3.WHAT IS SQL?

SQL stands for Structured Query Language , and it is used to communicate with the Database. This is a standard language used to perform tasks such as retrieval, updation, insertion and deletion of data from a database.

Standard SQL Commands are Select.

---

## 4.WHAT IS A DATABASE?

Database is nothing but an organized form of data for easy access, storing, retrieval and managing of data. This is also known as structured form of data which can be accessed in many ways.

Example: School Management Database, Bank Management Database.

---

## 5.WHAT ARE TABLES AND FIELDS?

A table is a set of data that are organized in a model with Columns and Rows. Columns can be categorized as vertical, and Rows are horizontal. A table has specified number of column called fields but can have any number of rows which is called record.

Example:.

Table: Employee.

Field: Emp ID, Emp Name, Date of Birth.

Data: 201456, David, 11/15/1960.

---

## 6.WHAT IS A PRIMARY KEY?

A primary key is a combination of fields which uniquely specify a row. This is a special kind of unique key, and it has implicit NOT NULL constraint. It means, Primary key values cannot be NULL.

---

## 7.WHAT IS A UNIQUE KEY?

A Unique key constraint uniquely identified each record in the database. This provides uniqueness for the column or set of columns.

A Primary key constraint has automatic unique constraint defined on it. But not, in the case of Unique Key.

There can be many unique constraint defined per table, but only one Primary key constraint defined per table.

---

## 8.WHAT IS A FOREIGN KEY?

A foreign key is one table which can be related to the primary key of another table. Relationship needs to be created between two tables by referencing foreign key with the primary key of another table.

---

## 9.WHAT IS A JOIN?

This is a keyword used to query data from more tables based on the relationship between the fields of the tables. Keys play a major role when JOINS are used.

---

## **10. 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.

---

## **11. 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.

---

## **12. WHAT IS AN INDEX?**

An index is performance tuning method of allowing faster retrieval of records from the table. An index creates an entry for each value and it will be faster to retrieve data.

---

## **13. 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.

---

## 14. WHAT IS A QUERY?

A DB query is a code written in order to get the information back from the database. Query can be designed in such a way that it matched with our expectation of the result set. Simply, a question to the Database.

---

## 15. WHAT IS SUBQUERY?

A subquery is a query within another query. The outer query is called as main query, and inner query is called subquery. SubQuery is always executed first, and the result of subquery is passed on to the main query.

---

## 16. 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.

---

## 17. WHAT IS A STORED PROCEDURE?

Stored Procedure is a function consists of many SQL statement to access the database system. Several SQL statements are consolidated into a stored procedure and execute them whenever and wherever required.

---

## 18. WHAT IS A TRIGGER?

A DB trigger is a code or programs that automatically execute with response to some event on a table or view in a database. Mainly, trigger helps to maintain the integrity of the database.

Example: When a new student is added to the student database, new records should be created in the related tables like Exam, Score and Attendance tables.

---

## 19. 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.

---

## 20. 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.

---

## 21. WHAT IS AUTO INCREMENT?

Auto increment keyword allows the user to create a unique number to be generated when a new record is inserted into the

table. AUTO INCREMENT keyword can be used in Oracle and IDENTITY keyword can be used in SQL SERVER.

Mostly this keyword can be used whenever PRIMARY KEY is used.

---

## 22. 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.

---

## 23. 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.

---

## 24. WHAT IS USER DEFINED FUNCTIONS?

User defined functions are the functions written to use that logic whenever required. It is not necessary to write the same logic several times. Instead, function can be called or executed whenever needed.

---

## 25. 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.

---

## 26. 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.

---

## 27. 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.

---

## 28. 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.

---

## 29. 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.

---

## 30. 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.

---

## 31. 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.

---

## 32. 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.

---

## 33. 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.

---

## 34. 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

---

## 35. 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]

---

### 36. HOW TO SELECT UNIQUE RECORDS FROM A TABLE?

Select unique records from a table by using DISTINCT keyword.

Select DISTINCT StudentID, StudentName from Student.

---

### 37. 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

---

### 38. 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.
- 

## 39. 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.

---

## 40. WHAT IS THE DIFFERENCE BETWEEN TRUNCATE AND DROP STATEMENTS?

TRUNCATE removes all the rows from the table, and it cannot be rolled back.

DROP command removes a table from the database and operation cannot be rolled back.

---

## 41. 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_'
```

---

## 42. 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.

---

### 43. WHAT IS CHECK CONSTRAINT IN SQL SERVER?

A check constraint in SQL Server (Transact-SQL) allows you to specify a condition on each row in a table.

Example:

```
CONSTRAINT check_employee_id CHECK (employee_id
BETWEEN 1 and 10000)
```

## DESIGN PATTERNS

### 1.WHAT IS CREATIONAL DESIGN PATTERN?

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

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

-  Singleton
  -  Factory
  -  Abstract Factory
  -  Prototype
  -  Builder
- 

## 2. WHAT IS STRUCTURAL DESIGN PATTERN?

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

Structural class patterns use inheritance to compose interface or implementations.

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

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

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

### 3.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.
- 

### 4.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.

---

## 5.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
- 

## 6.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.
-

## 7.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
- 

## 8.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.
-

## 9. 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.
- 

## 10. 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 data is 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.

---

## 11. 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

## 1. WHAT IS WEB API?

WebAPI is a framework which helps you to build/develop HTTP services.

---

## 2. WHY SELECT WEB API?

- It is used to create simple, non-SOAP-based HTTP Services
  - It is also an easy method for creation with Web API. With WCF REST Services
  - It is based on HTTP and easy to define, expose and consume in a REST-ful way.
  - It is lightweight architecture and ideal for devices that have limited bandwidth like smartphones.
- 

### 3.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.

---

### 4.WHAT ARE THE ADVANTAGES OF WEB API?

Advantages of Web API are:

- OData
- Filters
- Content Negotiation
- Self-Hosting

- Routing
  - Model Bindings
- 

## 5.WHAT ARE MAIN RETURN TYPES SUPPORTED IN WEB API?

A Web API controller action can return following values:

- Void – It will return empty content
  - HttpResponseMessage - It will convert the response to an HTTP message.
  - IHttpActionResult - internally calls ExecuteAsync to create an HttpResponseMessage
  - Other types - You can write the serialized return value into the response body
- 

## 6.WEB API SUPPORTS WHICH PROTOCOL?

Web App supports HTTP protocol.

---

## 7.WHICH .NET FRAMEWORK SUPPORTS WEB API?

NET 4.0 and above version supports web API.

---

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

Web API uses Json.NET library for JSON serialization.

---

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

500 - Internal Server Error

---

## 10. WHAT IS WEB API ROUTING?

Routing is pattern matching like in MVC.

All routes are registered in Route Tables.

For example:

```
Routes.MapHttpRoute (
```

```
Name: "ExampleWebAPIRoute",

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

defaults: new { id =
RouteParameter.Optional }
```

---

## 11. 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.

---

## 12. 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.

---

## 13. 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.
- 

## 14. EXPLAIN EXCEPTION FILTERS?

It will be executed when exceptions are unhandled and thrown from a controller method. The reason for the exception can be anything. Exception filters will implement "IExceptionFilter" interface.

---

## 15. 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.

---

## 16. HOW TO REGISTER EXCEPTION FILTER GLOBALLY?

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

```
GlobalConfiguration.Configuration.Filters.Add(new
MyTestCustomerStore.NotImplExceptionFilterAttribute());
```

---

## 17. EXPLAIN WHAT IS REST AND RESTFUL?

REST represents REpresentational State Transfer; it is entirely a new aspect of writing a web app. REST is architectural style. It has defined guidelines for creating services which are scalable. REST used with HTTP protocol using its verbs GET, PUT, POST and DELETE.

RESTFUL: It is term written by applying REST architectural concepts is called RESTful services. It focuses on system resources and how the state of the resource should be transported over HTTP protocol.

## 18. HOW CAN YOU HANDLE ERRORS IN WEB API?

Several classes are available in Web API to handle errors. They are `HttpError`, `Exception Filters`, `HttpResponseException`, and `Registering Exception Filters`.

---

## 19. 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
```

## 20. HOW CAN YOU PASS MULTIPLE COMPLEX TYPES IN WEB API?

Two methods to pass the complex types in Web API –

Using ArrayList and Newtonsoft array

---

## 21. STATE DIFFERENCES BETWEEN MVC AND WEBAPI

MVC framework is used for developing applications which have User Interface. For that, views can be used for building a user interface.

WebAPI is used for developing HTTP services. Other apps can also be called the WebAPI methods to fetch that data.

---

## 22. 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. In fact, even portable devices like Mobile devices can easily consume WebAPI which is certainly the biggest advantages of this technology.

---

## 23. 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.

---

## 24. 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.
- 

## 25. 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)
- 

## 26. 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.

---

## 27. 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.

---

## 28. 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.

---

## 29. 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.

---

## 30. 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.

---

## 31. 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.

---

## 32. 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.

---

## 33. WHAT IS UDDI?

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

---

## 34. 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).

---

## **35. 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
- 

## **36. 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.

---

## **37. 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.

---

### 38. MENTION WHAT IS THE ENDPOINT IN WCF AND WHAT ARE THE THREE MAJOR POINTS IN WCF?

Every service must have an **address** that determines where the service is located, contract that defines what the service does and **binding** that tells how to communicate with the service.

- Address: It specifies the location of the service which will be like <http://Myserver/Myservice>. To communicate with our service client it will use this location
- Contract: It specifies the interface between the server and client. It's a simple interface with some attribute
- Binding: It decides how two parties will communicate with each other in terms of transport and encoding and protocols

### 39. EXPLAIN HOW MANY TYPES OF CONTRACT DOES WCF DEFINES?

WCF defines four types of Contracts

- Service Contracts
  - Data Contracts
  - Fault Contracts
  - Message Contracts
- 

## 40. WHAT ARE THE TRANSPORT SCHEMAS DOES WCF SUPPORTS?

It supports

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

## 41. 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)
- 

## 42. 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

---

## 43. 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.

---

## 44. 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.
- 

## 45. 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.

---

## 46. 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

---

## **47. 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
- 

## **48. NAME THE NAMESPACE THAT IS USED TO ACCESS WCF SERVICE?**

System.ServiceModel is used to access WCF service

---

## **49. WHAT IS SOA STANDS FOR?**

Service Oriented Architecture

---

## **50. WHAT IS WCF STANDS FOR?**

# Windows Communication Foundation.

---