freelance_Project available to buy contact on 8007592194

SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql

freelance_Project available to buy contact on 8007592194

21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql
31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql
41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48		React+Springboot+MySql
49		React+Springboot+MySql
50		React+Springboot+MySql

1.Frame work:-

1) Choose right option ('s) below statements about the .NET CLR?

1. Common Language Runtime provides a language-neutral development and execution environment.

2. Common Language Runtime ensures that an application would not be able to access memory that it is not authorized to access.

3. Common Language Runtime provides services to run managed applications.
4. Common Language Runtime The resources are garbage collected.

5. Common Language Runtime provides services to run "unmanaged" applications.

a) Only 1 and 2 b) Only 1, 2 and 4

c) 1, 2, 3, 4 d) Only 4 and 5

2) What is true about Managed Code (MC)?

a) Managed code(MC) is compiled by the JIT(Just In Time) compilers

b) Managed code(MC) where resources are Garbage Collected(GC)

c) Managed code (MC) runs on top of Windows OS.

d) Managed code (MC) written to target the services of the Common Language Runtime (CLR).

3) Dot Net Framework consists of:

a. Common language runtime b. Set of class libraries

c. Common language runtime and set of class libraries d. None of above

4) Which of the following statements are correct about JIT?

1. JIT compiler compiles instructions into machine code at run time.

2. The code compiler by the JIT compiler runs under CLR.

3. The instructions compiled by JIT compilers are written in native code.

4. The instructions compiled by JIT compilers are written in Intermediate Language (IL) code

a. 1, 2, 3 b. 2, 4 c. 3, 4 d. 1, 2

5) Which of the following is the root of the .NET type hierarchy?

a. System.Type b. System. Parent

c. System.Base d. System. Object

6) Code that targets the Common Language Runtime is known as

a. unmanaged b. Distributed

c. Native Code d. Managed Code

7) Which of the following statements correctly define .NET Framework?

a. It is an environment for developing, building, deploying and executing Desktop Applications, Web Applications and Web Services.

b. It is an environment for developing, building, deploying and executing only Web Applications.

c. It is an environment for developing, building, deploying and executing Distributed Applications.

d. It is an environment for developing, building, deploying and executing Web Services.

8) How many types of JIT compilers available under CLR?

a. 4 b. 2 c. 1 d.3

9) Which of the following statements is correct about Managed Code?

a. Managed code is the code that is compiled by the JIT compilers.

b. Managed code is the code where resources are Garbage Collected.

c. Managed code is the code that runs on top of Windows.

d. All of above

10) Name Type of assembly

a. Private, shared assembly b. Public, protected assembly

c. All the above d. None

11) In Shared Assembly every project will have local copy

a. True b. False

12) In Private Assembly every project will have local copy

a. True b. False

13) For shared Assembly you have to generate Strong name

a. True b. False

14) Strong Name includes the

a. only name of the .net assembly,

b. only version number,

c. culture identity, and a public key token

d. all the above

15) Shared Assemblies are installed at:

a. System Assembly Cache b. Global Assembly Cache

c. Machine Assembly Cache d. Windows Assembly Cache

16) To create a key pair command is

a. sn –k b. sn-k c. ns –k d. none

17) Command to move dll file in shared location

a. gacutil/i dllfilename b. gc c. gautyil/I d. none

2.Net frame work:-

1) .entrypoint .maxstack 3

.locals ([0] int32 ValueOne,

[1] int32 ValueTwo,

[2] int32 V_2,

[3] int32 V_3)

IL_0000: ldc.i4.s 10

IL_0002: stloc.0

IL_0003: ldc.i4.s 20

IL_0005: stl

a. MSIL code b. Metadata

c. Assembly Manifest d. Module Manifest e. C#

2) From which one of the following locations does the garbage collector remove objects?

a. The system registry b. The thread stack

c. The managed heap d. The global assembly cache

e. The download cache

3) How does .NET Framework alleviate "DLL Hell"?

a. The Common Language Runtime (CLR) and Assemblies specify and enforce versioning rules and allow side-by-side execution of a software component

b. The Common Language Runtime (CLR) only allows a single version of a component to be registered in the Global Assembly Cache (GAC).

c. The Common Language Runtime (CLR) does not allow administrators to change the version of a component that an Assembly references externally.

d. The Common Language Runtime (CLR) and Assemblies can only use the version of a component with which they were compiled.

4) John wants to look at a human readable representation of the metadata and intermediate language (IL) code contained in a .NET Portable Executable (PE) file. Given the above scenario, what tool from the .NET SDK should John use?

a. ilasm.exe b. ildasm.exe c. al.exe d. dumpbin.exe

5) What is the relationship between Common Type System (CTS) and Common Language Specification (CLS)?

a. NET Languages each offer a subset of the CTS and a superset of the CLS.

b. NET Languages each offer a superset of the CTS and a subset of the CLS.

c. NET Languages each offer either the CTS set or the CLS set.

d. NET Languages all offer the same superset of the CTS.

6) Where is the Class Loader located?

a. In the Common Language Runtime's (CLR) Virtual Execution Engine

b. In the .NET source code compiler

c. In the Portable Executable File

d. In the host operating system

e. In the Global Assembly Cache (GAC)

7) Which one of the following creates the metadata tables contained in a PE file?

a. Source code compiler b. JIT Compiler

c. Class Loader d. Verifier

8) Which one of the following describes the Application Base property?

a. Source code compiler b. JIT Compiler

c. Class Loader d. Verifier

9) Which one of the following statements is true about MSIL code?

a. It is source code-specific.

b. It is architecture-specific.

c. It is compiled to native code by JIT compilers.

d. It is only stored in assembly resource files.

e. It is only found in static assemblies.

10) Which one of the following statements is true regarding how the .NET Framework minimizes "DLL Hell"?

a. It enforces that only one component of a given name can run on a machine at a time.

b. It only allows multiple versions of a given component to run on a machine at a time if they all are private assemblies.

c. It allows side-by-side execution on the same machine, at the same time, or even the same process, of any version of the same shared DLL.

d. It registers all assemblies with the COM+ catalog. e. It registers all assemblies with the Global Assembly Cache (GA

11) _____ is collection of reusable classes or type.

a. Base Class Library b. File Library

c. Both a and b are true d. None of the above

12) The common language runtime can be thought of as the environment that manages code execution. It provides core services, such as_____

a. code compilation b. memory allocation

c. thread management, and garbage collection d. All of the Above

13) The .NET Framework is designed for cross-language compatibility, which means, simply, that .NET components can interact with each other no matter what supported language they were written in originally.

a. This level of cross-language compatibility is possible because of the common language runtime.

b. This level of cross-language compatibility is possible because of the common Type System

c. This level of cross-language compatibility is possible because of the Common Language Specification

d. None of the above

14) Statement A: The Common Language Specification (CLS) defines the minimum

A. standards to which .NET language compilers must conform. Statement B: CLS ensures that any source

B. code successfully compiled by a .NET compiler can interoperate with the .NET Fram

a. Statement A is True b. Statement B is true

c. Both Statements are True d. None of the above

3.C# Basic:-

1) How many Bytes are stored by 'Long' Datatype in C# .net?

a) 8 b) 4 c) 2 d) 1

2) Correct Declaration of Values to variables 'a' and 'b'?

a) int a = 32, b = 40.6; b) int a = 42; b = 40;

c) int a = 32; int b = 40; d) int a = b = 42;

3) Arrange the following datatype in order of increasing magnitude sbyte, short, long, int.

a) long < short < int < sbyte b) sbyte < short < int < long

c) Short < sbyte < int < long d) short < int < sbyte < long

4) Which datatype should be more preferred for storing a simple number like 35 to improve execution speed of a program?

a) sbyte b) short c) int d) long

5) Which Conversion function of ‘Convert.TOInt32 ()’ and ‘Int32.Parse ()’ is efficient?

1) Int32.Parse () is only used for strings and throws argument exception for null string

2) Convert.Int32 () used for datatypes and returns directly ‘0’ for null string

a) 2 b) Both 1, 2 c) 1 d) None of the mentioned

6) What is the Size of ‘Char’ datatype?

a) 8 bit b) 12 bit c) 16 bit d) 20 bit

7) Select output for the following set of code.

```
1. static void Main(string[] args)
2. {
3. int a = 5;
4. int b = 10;
5. int c;
6. Console.WriteLine(c = ++ a + b ++);
7. Console.WriteLine(b);
8. Console.ReadLine();
9. }
```

a) 11, 10 b) 16, 10 c) 16, 11 d) 15, 11

4.Enum:-

1) Which among the following cannot be used as a datatype for an enum in C#.NET?

a) short b) double c) int

2) Choose the correct output for the C#.NET code given below?

```
1. enum days:int
2. {
3. sunday = -3,
4. monday,
5. tuesday
6. }
7. Console.WriteLine((int)days.sunday);
8. Console.WriteLine((int)days.monday);
9. Console.WriteLine((int)days.tuesday);
```

a) -3 0 1 b) 0 1 2 c) -3 -2 -1 d) sunday monday tuesday

3) Choose the correct output for given set of code?

```
1. enum color:int
2. {
3. red, 4. green,
5. blue = 5,
6. cyan,
7. pink = 10,
8. brown
9. }
10. console.writeline((int)color.green);
11. console.writeline((int)color.brown);
```

a) 2 10 b) 2 11 c) 1 11 d) 1 5

4) Correct the output for the C#.NET code given below?

1. enum letters

2. {

3. a,

4. b,

5. c

6. }

7. letters l;

8. l = letters.a;

9. Console.writeline(l); advertisements

a) -1 b) 0 c) a d) letters.a

5.Structure:-

1) Which of the following is a correct statement about the C#.NET code given below?

1. struct book

2. {

3. private String name;

4. private int pages;

5. private Single price;

6. }

7. book b = new book();

a) New structure can be inherited from struct book

b) When the program terminates, variable b will get garbage collected

c) The structure variable ‘b’ will be created on the stack

d) When the program terminates, variable b will get garbage collected

2) Choose the correct statement about structures in C#.NET?

a) Structures can be declared within a procedure

b) Structures can implement an interface but they cannot inherit from another structure

c) Structure members cannot be declared as private

d) a structure can be empty

3) When does a structure variable get destroyed?

a) When no reference refers to it, it will get garbage collected

b) Depends on whether it is created using new or without new operator

c) As variable goes out of the scope

d) Depends on either we free its memory using free() or delete()

6.Polymorphisms:-

1) The capability of an object in Csharp to take number of different forms and hence display behaviour as according is known as:

a) Encapsulation b) Polymorphism

c) Abstraction d) None of the mentioned

7.Inheritance:-

1) Which procedure among the following should be used to implement a 'Is a' or a 'Kind of' relationship between two entities?

a) Polymorphism b) Inheritance c) Templates

2) In Inheritance concept, which of the following members of base class are accessible to derived class members?

a) Static b) protected c) private d) shared

3) which form of inheritance is not supported directly by C# .NET?

a) Multiple inheritance b) Multilevel inheritance

c) Single inheritance d) Hierarchical inheritance

5) If no access modifier for a member of a class is specified, then class member accessibility is defined as?

a) Public b) protected c) private d)internal

6) using System; class shape

{ public int H; public int W; public shape(int h, int w)

{ H = h; W = w; } public double area()

{ return 0; } }

class rectangle :shape

{ public rectangle(int p, int q) : base(p, q) { }

public double area() { return H * W; } }

class triangle:shape

{ public triangle(int p, int q) : base(p, q) { }

public double area() { return (H * W)/2.0; } }

class Program {

static void Main(string[] args)

{ shape s1 = new triangle(5, 5);

Console.WriteLine (s1.area()); shape s2 = new rectangle (5, 5);
Console.WriteLine(s2.area()); Console.ReadLine (); } }

What will be the output

a. 0, 0 b. 25,12 c. 12.00 d. None

7) using System; class shape

{ public int H; public int W; public shape(int h, int w)

{ H = h; W = w; }

public virtual double area()

{ return 0; } }

class rectangle :shape

{ public rectangle(int p, int q) : base(p, q) { } public double area()

{ return H * W; } } class triangle:shape

{ public triangle(int p, int q) : base(p, q) { }

public double area() { return (H * W)/2.0; } }

class Program {

static void Main(string[] args)

{ shape s1 = new triangle(5, 5);

Console.WriteLine (s1.area()); shape s2 = new r

ectangle (5, 5);

Console.WriteLine(s2.area());

Console.ReadLine (); } }

a. 25,12.00 b.12.05,25 c.0,0 d. None

8) using System; using System.Collections.Generic;

class shape { public int H; public int W; public shape(int h, int w)

{ H = h; W = w; } public virtual double area()

{ return 0; } }

class rectangle :shape

{ public rectangle(int p, int q) : base(p, q) { }

public override double area() { return H * W; } }

class triangle:shape

{ public triangle(int p, int q) : base(p, q) { }

public override double area()

{ return (H * W)/2.0; } }

class Program { static void Main(string[] args)

{ shape s1 = new rectangle(5, 5);

Console.WriteLine (s1.area()); shape s2 = new triangle(6, 6);
Console.WriteLine(s2.area()); Console.ReadLine (); } }

a. 0, 25 b. 25, 18 c.25, 18.000000 d.0

8) using System; namespace ConsoleApplication7

{ abstract class shape { public int H; public int W; public shape(int h, int w)

{ H = h; W = w; }

public virtual double area()

{ return 0; } }

class rectangle :shape

{ public rectangle(int p, int q) : base(p, q) { }

public double area() { return H * W; } }

class triangle:shape {

public triangle(int p, int q) : base(p, q) { }

public double area() { return (H * W)/2.0; } }

class Program { static void Main(string[] args)

{ shape s1 = new rectangle(5, 5);

Console.WriteLine (s1.area());

shape s2 = new triangle (6, 6);

Console.WriteLine(s2.area());

Console.ReadLine (); } }

a.0,0 b.25,1 c. 15,18.000000 d. none

9) using System; using System.Collections.Generic;

namespace ConsoleApplication7

{ abstract class shape

{ public int H; public int W; public shape(int h, int w)

{ H = h; W = w; }

public abstract double area(); }

class rectangle :shape

{ public rectangle(int p, int q) : base(p, q) { }

public override double area() { return H * W; } }

class triangle:shape { public triangle(int p, int q) : base(p, q) { }

public override double area() { return (H * W)/2.0; } }

class Program

{ static void Main(string[] args) { shape s1 = new rectangle (5, 5);
Console.WriteLine (s1.area()); shape s2 = new triangle (6, 6);
Console.WriteLine(s2.area()); Console.ReadLine (); } }

a. 25, 18 b. 0, 0 c. 0, 15 d. None

8.Method Overloading:-

1) The process of defining two or more methods within the same class that have same name but different parameters list?

a) Method overloading b) method overriding

c) Encapsulation d) None of the mentioned

2) Which of these can be overloaded?

a) Constructors b) Methods c) Both a & b d) None of the mentioned

3) What is the process of defining a method in terms of itself that is a method that calls itself?

a) Polymorphism b) Abstraction c) Encapsulation d) Recursion

4) What could be the output of the following set of code?

```
1. class Program
2. {
3. static void Main(string[] args)
4. {
5. Console.WriteLine( vol(10));
6. Console.WriteLine( vol(2.5f, 5));
7. Console.WriteLine( vol( 5l, 4, 5));
8. Console.ReadLine();
9. }
10. static int vol(int x) 11. {
12. return(x * x * x);
13. }
14. static float vol(float r, int h)
15. {
1. return(3.14f * r * r * h);
2. }
```

3. static long vol(long l, int b, int h)

4. {

5. return(l * b * h);

6. }

7. }

a) 1000 0 100 b) 0 0 100 c) compile time error d) 1000 98.125 100

5) What could be the output for the set of code?

1. class overload

2. {

3. public int x;

4. int y;

5. public int add(int a)

6. {

7. x = a + 1;

8. return x;

9. }

10. public int add(int a, int b)

11. {

12. x = a + 2;

13. return x;

14. }

15. }

16. class Program

17. {

18. static void Main(string[] args)

19. {

```
20. overload obj = new overload();
21. overload obj1 = new overload();
22. int a = 0;
23. obj.add(6);
24. obj1.add(6, 2);
25. Console.WriteLine(obj.x);
26. Console.WriteLine(obj1.x);
27. Console.ReadLine();
28. }
29. }
```

a) 8 8 b) 0 2 c) 8 10 d) 7 8

6) What will be the output for the set of code?

```
1. static void Main(string[] args)
2. {
3. int i = 5;
4. int j = 6;
5. add(ref i);
6. add(6);
7. Console.WriteLine(i);
8. Console.ReadLine();
9. }
10. static void add(ref int x)
11. {
12. x = x * x;
13. }
14. static void add(int x)
```

15. {

16. Console.WriteLine(x * x * x);

17. }

a) Compile time error b) 25 0 c) 216 0 d) 216 25

9.Over rider:-

1) Which keyword is used to declare a base class method while performing overriding of base class methods?

a) This b) virtual c) override d) extend

2. The process of defining a method in subclass having same name & type signature as a method in its superclass is known as?

a) Method overloading b) Method overriding c) none of the mentioned

3) Which of the given modifiers can be used to prevent Method overriding?

a) Static b) Constant c) Sealed d) final

4) What will be the output for the given set of code?

class A

{ public virtual void display()

{ Console.WriteLine("A"); } }

class B: A { public override void display()

{ Console.WriteLine(" B "); } }

class Program

{ static void Main(string[] args)

{ A obj1 = new A();

B obj2 = new B();

A r; r = obj1; r.display();

r = obj2; r.display();

Console.ReadLine(); } }

a) A, A b) B, B c) Compile time error d) A, B

5) The modifier used to hide the base class methods is?

a) Virtual b) New c) Override d) Sealed

6) What will be the output for the given set of code?

class a

{ public void fun()

{ Console.WriteLine("base method"); } }

class b: a { public new void fun()

{ Console.WriteLine(" derived method "); } }

class Program { static void Main(string[] args)

{ b k = new b(); k.fun(); Console.ReadLine(); } }

a) base method b) derived method

c) Code runs successfully prints nothing d) Compile time error

7) What will be the output for the given set of code?

class A

{ public virtual void display()

{ Console.WriteLine("A"); } }

class B: A { public override void display()

{ Console.WriteLine(" B "); } }

class Program

{ static void Main(string[] args)

{ A obj1 = new A();

B obj2 = new B();

A r; r = obj1; r.display();

r = obj2; r.display();

Console.ReadLine(); } }

a) A, A b) B, B c) Compile time error d) A, B

10.Constructor Overloading:-

1) What will be the output of the given set of code?

```
1. class maths
2. {
3. public int length;
4. public int breadth;
5. public maths(int x, int y)
6. {
7. length = x;
8. breadth = y;
9. Console.WriteLine(x + y);
10. }
11. public maths(double x, int y)
12. {
13. length = (int)x;
14. breadth = y;
15. Console.WriteLine(x * y);
16. }
```

```
17. }
18. class Program
19. {
20. static void Main(string[] args)
21. {
22. maths m = new maths(20, 40);
23. maths k = new maths(12.0, 12);
24. Console.ReadLine();
25. }
26. }
```

a) 60, 24 b) 60, 0 c) 60, 144 d) 60, 144.0

2) What will be the output of the given set of code?

```
1. class maths
2. {
3. public int length;
4. public int breadth;
5. public maths(int x)
6. {
7. length = x + 1;
8. }
9. public maths(int x, int y)
10. {
11. length = x + 2;
12. }
13. }
14. class Program
```

```
15. {
16. static void Main(string[] args)
17. {
18. maths m = new maths(6);
19. maths k = new maths(6, 2);
20. Console.WriteLine(m.length);
21. Console.WriteLine(k.length);
22. Console.ReadLine();
23. }
24. }
```

a) 8, 8 b) 0, 2 c) 8, 10 d) 7, 8

3) What will be the output of the given set of code?

```
1. class maths
2. {
3. int i;
4. public maths(int x)
5. {
6. i = x;
7. Console.WriteLine(" hello: ");
8. }
9. }
10. class maths1 : maths
11. {
12. public maths1(int x) :base(x)
13. {
14. Console.WriteLine("bye");
```

15. }

16. }

17. class Program

18. {

19. static void Main(string[] args)

20. {

21. maths1 k = new maths1(12);

22. Console.ReadLine();

23. }

24. }

a) hello bye b) 12 hello ` c) bye 12 d) Compile time error

11.Property getset:-

1) Select the correct statement about properties of read and write in C#.NET?

a) A property can simultaneously be read or write only

b) A property can be either read only or write only

c) A write only property will only have get accessor

d) A read only property will only have set accessor

2) What will be the output of the following snippet of code?

1. class number

2. {

3. int length = 50;

4. public int number1

5. {

```
6. get
7. {
8. return length;
9. }
10. set
11. {
12. length = value;
13. }
14. }
15. }
16. class Program
17. {
18. public static void Main(string[] args)
19. {
20. number p = new number();
21. p.number1 = p.number1 + 40;
22. int k = p.number1 * 3 / 9;
23. Console.WriteLine(k);
24. Console.ReadLine();
25. }
26. }
```

a) 0 b) 180 c) 30 d) Compile time error

3) What will be the output of the following snippet of code?

```
1. class number
2. {
3. int length = 60;
```

```
4. public int number1
5. {
6. get
7. {
8. return length;
9. }
10. }
11. }
12. class Program
13. {
14. public static void Main(string[] args)
15. {
16. number p = new number();
17. int l;
18. l = p.number1 + 40;
19. int k = l * 3 / 4;
20. Console.WriteLine(k);
21. Console.ReadLine();
22. }
23. }
```

a) 30 b) 75 c) 80 d) 0

4) What will be the output of following snippet of code?

```
1. class number
2. {
3. private int num1;
4. private int num2;
```

```
5.public int anumber
6. { get
7. { return num1; }
8. set
9. { num1 = value; }
10. }
11. public int anumber1
12. {
13. get
14. { return num2; }
15. set
16. { num2 = value; }
17. }
18. }
19. class Program
20. {
21. public static void Main(string[] args)
22. {
23. number p = new number();
24. p.anumber = 20;
25. number k = new number();
26. k.anumber1 = 40;
1. int m = p.anumber;
2. int t = k.anumber1;
3. int r = p.anumber + k.anumber1;
4. Console.WriteLine("number = " +m);
```

5. Console.WriteLine("number = " +t);

6. Console.WriteLine("sum = " +r);

7. Console.ReadLine();

8. }

9. }

a) 0 b) Compile time error c) 60 d) none of the above mentioned

5) Consider a class maths and we had a property called as sum.b is a reference to a maths object and we want the statement b.sum = 10 to fail.Which of the following is the correct solution to ensure this functionality?

a) Declare sum property with both get and set accessors

b) Declare sum property with only get accessor

c) Declare sum property with get, set and normal accessors

d) None of the mentioned

6) Consider a class maths and we had a property called as sum. b which is the reference to a maths object and we want the statement Console. WriteLine (b.sum) to fail. Which among the following is the correct solution to ensure this functionality?

a) Declares sum property with only get accessor

b) Declares sum property with only set accessor

c) Declares sum property with both set and get accessor

d) Declares sum property with both set, get and normal accessor

7. Consider a class maths and we had a property called as sum.b is a reference to a maths object and we want the code below to work. Which is the correct solution to ensure this functionality? b. maths = 10;

Console. WriteLine(b.maths);

a) Declare maths property with get and set accessors

b) Declare maths property with only get accessors

c) Declare maths property with only set accessors

d) Declare maths property with only get, set and normal accessors

12.Interface:-

1) Which statement correctly defines Interfaces in C#.NET?

a) Interfaces cannot be inherited

b) Interfaces consists of data static in nature and static methods

c) Interfaces consists of only method declaration

d) None of the mentioned

2) A class consists of two interfaces with each interface consisting of three methods. The class had no instance data. Which of the following indicates the correct size of object created from this class?

a) 12 bytes b) 16 bytes c) 0 bytes d) 24 bytes

3) Select the correct statement among the given statements?

a) One class could implement only one interface

b) Properties could be declared inside an interface

c) Interfaces cannot be inherited

d) None of the above mentioned

4) Which of the following is the correct way of implementing an interface addition by class maths?

a) class maths : addition {}

b) class maths implements addition {}

c) class maths imports addition {}

d) None of the mentioned

5) Does C#.NET support partial implementation of interfaces?

a) True b) False c) Can't Say d) None of the above mentioned

6) Access specifiers which can be used for an interface are?

a) Public b) Protected c) Private d) All of the mentioned

7) using System; interface I1

{ void A(); } interface I2 { void A(); }

class C : I1, I2 { public void A()

{ Console.WriteLine("C.A()"); } }

Class entry { static void main()

{ C c = new C(); I1 i1 = (I1)c;

I2 i2 = (I2)c; } c.A(); I1.A();

I2.A(); }

What will be the output of the program.

A. C.A() C.A() A() B. c.A(); i2.A(); i1.A();

C.Error D. None

8) using System; interface I1

{ void A(); } interface I2 { void A(); }

class C : I1, I2 { public void A() {

Console.WriteLine("C.A()"); }

void I1.A() {

Console.WriteLine("I1.A()"); } }

Class entry

{ static void main()

{ C c = new C(); c.A(); I2 i2 = c; c.A();}}

a. C.A(),C.A() b. C.A(),I1.A() c.I1.A(), C.A() d. None

9) using System; interface I1

{ void A();} interface I2

{ void A();}

class C : I1, I2

{ void I1.A() {

Console.WriteLine("I1.A()"); } }

Class entry { static void main()

{ C c = new C(); c.A(); }}

a. Compile time Error b. I2.A () c. Run time Error d. none

10) using System; interface I1

{ void A();} interface I2

{ void A();} class C : I1, I2

{ void I2.A() {

Console.WriteLine("I2.A()"); }

Void A() { Console.WriteLine("I1.A()");} }

Class entry { static void main()

{ C c = new C(); I2 x=new C(); c.A(); x.A();}}

a. I1.A(),I2.A() b. I2.A(),I1.A() c. Error d. None

11) using System; interface I1

{ void A();} interface I2

{ void I1.A();} class C : I1, I2

{ void I2.A() {

Console.WriteLine("I2.A()"); }

Void A() { Console.WriteLine("I1.A()");} }

Class entry { static void main()

{ I1 x=new C(); I2 p=new c(); x.A(); p.A(); }}

a. I1.A(),I2.A() b.I1.A(); c. Error d. None

12) using System;

class Test

{ static void Main()

{ A. F(); B. F() } }

class A { static A()

{ Console.WriteLine("Init A"); }

public static void F()

{ Console.WriteLine("A.F"); } }

class B { static B() {

Console.WriteLine("Init B"); }

public static void F() {

Console.WriteLine("B.F"); } }

What will be the output

A.Init A A.F Init B B.F B.None C. Init B B.F Init A A.F D.Error

13) Constant variable are by default static

a. True b. false

14) Readonly variable must be either initialised at time of declaration or in constructor

a. True b. False

15) Readonly variable must be initialised at time of declaration

a. true b. false

16) Readonly variable required data at compile time

a. True b. False

17)Const variable required data at runtime

a. True b. False

18) class program

{ const int a=5;

Static void Main()

{ program C= new program();

Console.WriteLine(C.a); } }

a. Compile t b.5 c. none d. run time error

19) class program

{ const int a=5;

Static void main()

{ program C= new program();

Console.WriteLine(program.

a); } }

a. Compile ti b.5 c. none d. run time error

20) You cannot declare static variable inside method

a. True b. False

21) You cannot declare constan variable inside method

a. true b. false

13.Exception:-

1) Which among the following is NOT an exception?

a) Stack Overflow b) Arithmetic Overflow or underflow

c) Incorrect Arithmetic Expression d) All of the above mentioned

2) Select the statements which describe the correct usage of exception handling over conventional error handling approaches?

a) As errors can be ignored but exceptions cannot be ignored

b) Exception handling allows separation of program's logic from error handling logic making software more reliable and maintainable

c) try – catch – finally structure allows guaranteed clean up in event of errors under all circumstances

d) All of the above mentioned

3) Select the correct statement about an Exception?

a) It occurs during loading of program

b) It occurs during Just-In-Time compilation

c) It occurs at run time

d) All of the above mentioned

4) Which of these keywords is not a part of exception handling?

a) Try b) finally c) thrown d) catch

5) Which of these keywords must be used to monitor exceptions?

a) try b) finally c) throw d) catch

6) Which of these keywords is used to manually throw an exception?

a) try b) finally c) throw d) catch

7) Choose the correct output for the given set of code:

```
1. class program
2. {
3. static void main(string[] args)
4. {
5. int i = 5;
6. int v = 40;
7. int[] p = new int[4];
8. try
9. {
10. p[i] = v;
11. }
12. catch(IndexOutOfRangeException e)
13. {
```

14. Console.WriteLine("Index out of bounds");

15. }

16. Console.WriteLine("Remaining program");

17. }

18. }

a) Value 40 will be assigned to a[5];

b) The output will be : Index out of bounds Remaining program

c) The output will be : Remaining program

d) None of the above mentioned

8) Choose the correct output for the given set of code:

1. static void Main(string[] args)

2. {

3. try

4. {

5. Console.WriteLine("csharp" + " " + 1/Convert.ToInt32(0));

6. }

7. catch(ArithmeticException e)

8. {

9. Console.WriteLine("Java");

10. }

11. Console.ReadLine();

12. }

a) csharp b) java c) Run time error d) csharp 0

9) Which of the following is the correct statement about exception handling in C#.NET?

a) Finally clause is compulsory

b) A program can contain multiple finally clauses

c) The statement in final clause will get executed no matter whether an exception occurs or not

d) All of the above mentioned

10) Choose the correct output for given set of code:

```
1. class Program
2. {
3. static void Main(string[] args)
4. {
5. try
6. {
7. Console.WriteLine("csharp" + " " + 1/0);
8. }
9. finally
10. {
11. Console.WriteLine("Java");
12. }
13. Console.ReadLine();
14. }
15. }
```

a) csharp 0 b) Run time Exception generation c) Compile time error d) Java

11) What will be the output of given code snippet?

```
1. {
2. try
3. {
4. int []a = {1, 2, 3, 4, 5};
5. for (int i = 0; i < 7; ++i)
```

6. Console.WriteLine(a[i]);

7. }

8. catch(IndexOutOfRangeException e)

9. {

10. Console.WriteLine("0");

11. }

12. Console.ReadLine();

13. }

a) 12345 b) 123450 c) 1234500 d) Compile time error

12. When no exception is thrown at runtime then who will catch it?

a) CLR b) Operating System c) Loader d) Compiler

15.Basic C:-

1) _____ pops up a list of methods that can be called on that object,instead of typing the full method name.

a) Intelligence b) intelligence c) goodsense d) intellisense

2) C# supports all the key object oriented concepts such as encapsulation inheritence and polimorphism

a. True b. False

3) C# is case sensitive language.

a. True b. False

4) We use the _____ function to write a string to the screen.

a. Console.Wrileln() b. Console.WriteLine()

c. Console.Write() d. Console.PrintLine()

5) _____ method can by called creating an instance of a class.

a. Non Static b. Static c. All of the above d.None of the Above

6) C# provides us with two predefined references types _____ and _____.

a. int ,float b. Object, String c. bool , int

7) In an enum type each of the name constant should necessarily have an ___ type.

a. int b. double c. string

8) A reference to a reference-type instance requires how many bytes?

a. 2 bytes b. 4 bytes c.8 bytes d.16 bytes

9) In C#, what character is used to indicate a verbatim string literal?

a. @ b. ! c. “ d. #

10) What is boxing?

A. The conversion of a value type to an object instance

B. The conversion of an object instance to a value type.

C. The conversion of a value type to reference type.

D. The conversion of a reference type to a value type

11) What is the work of JIT compiler in .NET framework?

a. It complies the assemblies.

b. It translates the code in the machine language.

c. It translates the MSIL code of assembly and uses the CPU architecture of the machine to execute a .NET application.

d. It translates the assembly code of MSIL and uses the CPU architecture of the machine to execute a .NET application.

12) Benefit of .NET framework is/are_____.

a. Consistent programming Model

b. Language Interoperability

c. Automatic management of resources

d. All of the above

13) Which statement is used to replace multiple if statement in code?

a. Select – Case b.Switch – Case c.Both 1 and 2 d.None of the above

14) Which of the following is not a method of System. Object?

a. GetType b.ToString c.Equals d.Clone

15) What will be the o/p ?

```
class Test
{ static void Main(string[] args)
{ int a; a = 1; if(a) System.Console.WriteLine("I can use c# wrongly");
else System.Console.WriteLine("I can take chances"); } }
```

a. I can use c# wrongly b. I can take chances

c. Runtime Error d. Compile Time Error

16.Deligate Lambda:-

1) To Handel exception in C# you must use

a. Try catch block b. Only try c. Try – finally d. None

2) All Exceptions derived from

a. Exception class b. Application exception c. System Exception

3) A. An anonymous method cannot access ref or out parameters of the defining method.

B. An anonymous method cannot have a local variable with the same name as a local

a. Only B is true b. only A is true

c. none d. both statements are true

4) delegate void CountIt(int end);

class AnonMethDemo3 { static void Main()

{ int result;

CountIt count = delegate (int end)

{ int sum = 0;

for(int i=0; i <= end; i++)

{ Console.WriteLine(i);

sum += i; }

return sum;

// return a value from an anonymous method };

result = count(3);

Console.WriteLine("Summation of 3 is " + result); } }

a. 6 b. 0 c. Error d. None

5) delegate int CountIt(int end);

class AnonMethDemo3

{ static void Main()

{ int result; CountIt count = delegate (int end)

{ int sum = 0; for(int i=0; i <= end; i++)

{ Console.WriteLine(i); sum += i; }

return sum;

// return a value from an anonymous method };

result = count(3);

Console.WriteLine(result); } }

a. 1,2,3 6 b. none c. 1,2,3,

6) delegate int addition(int x, int y); class myclass

{ public int add(int p, int q) { return p + q; }

public int mul(int p, int q)

{ return p * q; } }

class Program

{ static void Main(string[] args)

{ myclass m = new myclass();

addition a =delegate(int p,int q)

{int r; r=p+q; return r;};

a += delegate(int p, int q)

{ int r; r = p * q; return r; };

Console.WriteLine(a.GetInvocationList().Length);

int invo = a(3, 5);

Console.WriteLine(invo); Console.ReadLine(); } } }

a.2, 15 b.15,8 c. Error d. none

7) delegate int Incr(int v);

class SimpleLambdaDemo

{ static void Main()

{ Incr incr = count => count + 2; int x = -5;

while(x <= 0) { Console.Write(x + " "); x = incr(x);

// increase x by 2

} } }

a. -5,-3,-1 b. None C. 5,3,1, D. Error

8) deligate for this lambda expression n => n % 2 ==

A.delegate true deli(); B. deligate bool deli(); V.IMP

C. deligate int deli(); D. none

9).Using system;

Delegat bool isEven(int x);

Class myclass

{ public static void Main()

{ isEven isEven = n => n % 2 == 0;

// Now, use the isEven lambda expression Console.WriteLine("Use isEven lambda expression: ");

for(int i=1; i <= 3; i++) if(isEven(i))

Console.WriteLine(i + " is even."); } }

a. 2 b. none c. 1,2,3 d. Error

10) Data written before => is known as

a. input parameter b. output parameter

c. represent return value d. None Name

16.Name Method:-

1) using System; class Program

{ static void Main(string[] args)

{ mycall("vita"); mycall("vita",55);

Console.ReadLine(); }

static void mycall(string message, int age =25)

{ Console.WriteLine("{0}", message);

Console.WriteLine("{0}", age); } }

a. Vita, 25 ,vita, 55 b.Vita,vita,55

c. Error d.Vita,55,vita,25

2) using System;

class Program

{ static void Main(string[] args)

{ DisplayFancyMessage(message: "vita", age: 25,addr: "juhu");
Console.ReadLine(); }

static void DisplayFancyMessage(int age,string message, string addr)

{ Console.WriteLine(message); Console.WriteLine("{0} {1}",age,addr); } } }

a. vita,25,juhu b. error C.juhu,vita,25 d. runtime error

3) using System;

class Program

{ static void Main(string[] args)

{ DisplayFancyMessage(message= "vita", age= 25,addr= "juhu");
Console.ReadLine(); }

static void DisplayFancyMessage(int age,string message, string addr)

{ Console.WriteLine(message);

Console.WriteLine("{0} {1}",age,addr); } } }

a. vita,juhu,25 b. Error c. juhu,vita,25 d. runtime error

4) IClonable interface has abstract method

a. Clone ===clone b. memberwiseclone c. both d. None

5) class Program

{ static void Main(string[] args)

{ DisplayFancyMessage("Wow! Very Fancy indeed!", 50, name:"raj");
DisplayFancyMessage("geeta", message: "hello",50); Console.ReadLine(); }
static void DisplayFancyMessage(string message, int number, string name,)

{ Console. WriteLine("{0},{1},{2}",number, name, message); } }

a. Error b.50, geeta, hello c.hello,geeta,50 d. none

6) foreach loop internally calling

a. Iclonable b. IEnumerable c. both d. none

7) using System; class Program

{ static void Main(string[] args)

{ EnterLogData(message:”Error”,string owner = "Programmer", DateTime timeStamp = DateTime.Now)

Console.ReadLine(); }

static void EnterLogData(string message,string owner = "Programmer", DateTime timeStamp = DateTime.Now)

{ Console.Beep();

Console.WriteLine("{0}", message);

Console.WriteLine("{0}", owner);

Console.WriteLine("{0}", timeStamp); }

a. Error b. Error, Programmer c. none d. Programmer, Error

8) IComparable has abstract method

a. compareTo b. compare c. comparer d. all the above

9) IComparer has abstract method

a. Clone b. compare c. comparer d. none

10) Which statement is true

A. when you implement interface and use abstract method you must use public access modifier.

B. when you implement interface and use abstract method you may use public access modifier

a. only A is true b. both are true c. only b is true d. none

11) Which statement is true

A. MemberwiseClone() method copy value type bit by bit and for reference type use shallow copy

B. MemberwiseClone() method copy value type and reference type as shallow copy

a. onlyb is true b. only a is true c. none d. both

12) To short array you have

a. static sort() method in Array class b. virtual sort() method in Array class

c. user have to write own algorithm d. none

13) What will be the output using System;

delegate int addition();

class myclass

{ int a, b; public int add()

{ return a + b;

} public myclass(int a, int b) { a = a;b = b; } }

class Program {

static void Main(string[] args)

{ myclass m = new myclass(6,6);

addition a=m.add; int r = a(); Console.WriteLine(r);

Console.ReadLine(); } } }

a. 0 b. 12 c. Error d. None

14) using System;

delegate int addition();

class myclass

{ int a, b; public int add()

{ return a + b; }

public myclass(int a, int b)

{this. a = a;this.b = b; } }

class Program

{ static void Main(string[] args)

{ myclass m = new myclass(6,6);

addition a=m.add; int r = a();

Console.WriteLine(r); Console.ReadLine(); } } }

a. 12 b. None c. Error d. 0

15) delegate int addition(int x,int y); class myclass

{ public int add(int p,int q)

{ return p + q; }

public int mul(int p,int q)

{ return p * q;

} }

class Program

{ static void Main(string[] args)

{ myclass m = new myclass();

addition a=m.add;

addition b = m.mul; addition tot = a + b;

int r = tot(3,5); Console.WriteLine(r);

Console.ReadLine(); } } }

a. 15 b. Error c. 8, 15 d. none

16) deligate is derived from

a. System. Deligat b. System . MulticastDelegate

c. none d. from both

17) int invocationCount = d1.GetInvocationList().GetLength(0); the above code assume d1 is variable of a type deligate

a. This method give length of method bind with deligate

b. This method give list of method

c. None

d. This method give list of parameter of method

18) readonly key are internally static

a. True b. False

19) readonly key can not be used in method

a. true b. False

20) Which statement is true

A.as operator is like a cast,if conversion not possible it will return null instead of raising exception

B as operator is like a cast,if conversion not possible it will raise exception

a. only A b. only B c. both true d. both false

21) Array.Sort() method use a.

a. Quicksort algorithm. b. Heapsort algorithm

c. insertion sort algorithm. d. all three depend on size of data

22) as operator perform only

a. reference conversion b. nullable conversion

c. boxing conversion d. all the above

17.Assembly Coll Basic:-

1) Name Type of assembly

a. Private ,shared assembly b. Public ,protected assembly

c. All the above d. None

2) In Shared Assembly every project will have local copy

a. True b. False

3) In Private Assembly every project will have local copy

a. True b. False

4) For shared Assembly you have to generate Strong name

a. true b. False

5) Strong Name includes the

a. only name of the .net assembly, b. only version number,

c. culture identity, and a public key token d. all the above

6) Shared Assemblies are installed at:

a. System Assembly Cache b. Global Assembly Cache

c. Machine Assembly Cache d. Windows Assembly Cache

7) To create a key pair command is

a. sn –k b. sn-k c. ns -k d. none

8) Command to move dll file in shared location

a. gacutil/i dllfilename b. gc c. gautyil/I d. none

9) What are delegates?

a. Value Pointer b. Function Pointer

c. Pass By Reference d. Pass By Value

10) Generics provide better performance because they do not result in boxing or unboxing penalties when storing value types.

a. True b. False

11) Generics are not type safe because they can contain different type you specify.

a. True b. False

12) To help overcome the limitations of a simple array, the .NET base class libraries ship with

a. Thread class b. Collection class c. None d. Connection class

13) Collection classes are built to dynamically resize themselves on the fly as you insert or remove items

a. True b. False

14) Array List is in namespace

a. System b. System.Collection c. System.Collection.Generic d. none

15) When creating a C# Class Library project, what is the name of the supplementary file that Visual Studio.NET creates that contains General Information about the assembly?

a. AssemblyInfo.xml b. AssemblyInfo.cs

c. AssemblyInformation.cs d. AssemblyAttributes.cs

16) Which of the following is a value type, and not a reference type?

a. array b. delegate c. enum d. class

17) What is the difference between Overridding and Overloading?

a. Overridding, same name with different return type and overloading same name with different argument

b. Overridding is dynamic, overloading is static

c. Overridding, same signature with different definition, overloading has different signature

d. All the above

Q18) A reference to a reference-type instance requires how many bytes?

a. 2 bytes b. 4 bytes c.8 bytes d. 16 bytes

19) Which of the following is the C# escape character for Null?

a. \n b. \0 c.\f d. \v

20) Which keyword is used in C# to prevent a class from being inherited by another class?

a. override b. protected c. sealed d. NotInheritable

21) C# types are defined in _____, organized by _____, compiled into _____, and then grouped into _____.

a. files, modules, namespaces, assemblies

b. files, namespaces, assemblies, modules

c. files, assemblies, namespaces, modules

d. files, namespaces, modules, assemblies

22) What is Boxing?

a. The conversion of a value type to an object instance

b. The conversion of an object instance to a value type.

c. The conversion of a value type to reference type.

d. The conversion of a reference type to a value type

23) Which of the following is true for a special member of the class namely 'this'

a. this cannot be used in a static method

b. this cannot be used in a class A to access a member of class B

c. The this member can never be declared: it is automatically implied when you create a class

d. All the above are correct

e. None of above

18.ADO _NET:-

1) To use the .NET Framework Data Provider for SQL Server, an application must reference the _____ namespace.

a) System.Data.Client b) System.Data.SqlClient

c) System.Data.Sql d) None of the mentioned

2) _____object is used to fill a DataSet/DataTable with query results in ADO.net.

a) DataReader b) Dataset c) DataAdapter d) DataTables

3) classes used to access a SQL Server database in the managed space.Valid Code for Creating a SqlConnection Object would be :

a) SqlConnection conn = new SqlConnection("Data Source=(local);Initial Catalog=Northwind;Integrated Security=SSPI");

b)SqlConnect conn = new SqlConnection("Data Source=(local);Initial Catalog=Northwind;Integrated Security=SSPI");

c)SqlConnection conn = new SqlConnect("Data Source=(local);Initial Catalog=Northwind;Integrated Security=SSPI"); advertisements

d) All of the mentioned

4). Syntax for closing and opening the connection in ADO.net is :

a) sqlConn.Open() and sqlConn.close()

b) sqlConn.open() and sqlConn.Close()

c) sqlConn.Open() and sqlConn.Close()

d) None of the mentioned

5) The DataSet object is a _____ storage.

a. connected b. Disconnected c. polling d. None

6) _____ is a bridge between a DataSet and data source for retrieving and saving data.

1. DataControler 2. DataCommand

3. DataAdapter 4. None

7) How instantiate the DataReader ?

a. by calling a Command object's ExecuteReader.

b. by calling a Query object's ExecuteQuery.

c. by calling a new().

d. None

8) When we need to retrieve only a single value from the Database,which Method is efficient

a. ExecuteReader() b. ExecuteScalar()

c. ExecuteNonQuery() d. ExecuteXmlReader()

9) If we are not returning any records from the database which method is used

a. ExecuteReader () b. ExecuteScalar ()

c. ExecuteScalar () d. ExecuteNonQuery()

10) To populate the data set, which methord of DataAdapter is used

a. GetData() b. FillData() c. FillDataset() d.Fill()

11) What does ADO stand for?

a. Advanced Data Object b. Active Data Objects

c. ActiveX Directory Objects d. ActiveX Data Objects

19.ASP:-

1) What does ASP stand for?

a. All Standard Pages b. Active Server Pages

c. A Server Page d. Active Standard Pages

2) What attribute must be set on a validator control for the validation to work?

a.Validate b. ValidateControl

c. ControlToBind d. ControlToValidate

3) What is the Web.config file used for ?

a. To store the global information and variable definitions for the application

b. To store the global information and variable definitions for the application

c. To configure the web server

d. To configure the web browser

4) What is the file extension used for ASP.NET files?

a. ASP b. ASPX c. Web d. None of the above

5) The first event triggers in an aspx page is.

a. Page_Init() b. Page_Load() c. Page_click()

6) What class does the ASP.NET Web Form class inherit from by default?

a. System.Web.UI.Page b. System.Web.UI.Form

c. System.Web.GUI.Page d. System.Web.Form

7) We can manage states in asp.net application using

a. Session Objects b. Application Objects

c. Viewstate d. All of the above

8) Caching type supported by ASP.Net

a. Output Caching b. DataCaching c. a and b d. none of the above

9) What is used to validate complex string patterns like an e-mail address?

a. Extended expressions b. Basic expressions

c. Regular expressions d. Irregular expressions

10) An alternative way of displaying text on web page using

a. asp:label b. asp:listitem c. asp:button

11) Default Session data is stored in ASP.Net.

a. StateServer b. Session Object c. InProcess d. all of the above

12) How do you get information from a form that is submitted using the "post" method?

a. Request.QueryString b. Request.Form

c. Response.write d. Response.writeln

13) Which object can help you maintain data across users?

a. Application object b. Session object

c. Response object d. Server object

14) Which of the following ASP.NET object encapsulates the state of the client?

a. Session object b. Application object

c. Response object d. Server object

15) Which of the following control is used to validate that two fields are equal?

a. RegularExpressionValidator b. CompareValidator

c. equals() method d. RequiredFieldValidator

16) Which of the following transfer execution directly to another page?

a. Server.Transfer b. Response.Redirect

c. Both A. and B. d. None of the Above

17) The type of code found in Code-Behind class is _____?

a. Server-side code b. Client-side code

c. Both A. and B. d. None of the above

18) When an .aspx page is requested from the web server, the out put will be rendered to browser in following format.

a. HTML b. XML c. WML d. JSP

19) Which of the following is true?

a. IsPostBack is a method of System.UI.Web.Page class

b. IsPostBack is a method of System.Web.UI.Page class

c. IsPostBack is a readonly property of System.Web.UI.Page class

20) Does the EnableViewState allows the page to save the users input on a form?

a. Yes b. No

21) Explain the significance of Server .MapPath

a. Returns the Virtual Path of the web folder

b. Maps the specified virtual path to Physical path

c. Returns the physical file path that corresponds to virtual specified path d. All the above

22) By default, ASP.NET store SessionIDs in _____.

a. Cookies b. Cache c. Database d. Global variable

20.ENTITY:-

1) Which of the following is True?

a. Entity Framework is an ORM framework.

b. Entity Framework is an open source ORM framework.

c. Entity Framework is database mapping tool.

d. Entity Framework is object mapping tool.

2) A pattern of loading related data where a query for one type of entity also loads related entities as part of the query is called:

a. Lazy loading b. Eager loading

c. Explicit loading d. Quick Loading

3) Which of the following development approaches are supported in Entity Framework?

a. Code First b. Database First

c. Model First d. All of the above

4) What window in Visual Studio display CSDL, MSL and SSDL of Entity Framework?

a. Model window b. Model Browser

c. EDM Designer d. Solution Explorer

5) Which of the following is TRUE?

a. DbContext can not be used in Code First approach

b. ObjectContext is a wrapper around DBContext

c. DbContext is a wrapper around ObjectContext

d. DbContext is a sealed class which cannot be override.

6) CSDL stands for

a. Common Schema Definition Language

b. Conceptual Schema Definition Language

c. Conceptual Store Definition Language

d. Conceptual Storage Definition Language

7) Which of the following query syntax can be used to query EDM?

a. LINQ-to-Entity b. Entity SQL

c. Native SQL d. All of the above

8) An XML-based language that describes the storage model of an Entity Framework application is called

a. SSDL b. CSDL c. EDM d. MSL

9) An XML-based language that describes the mapping between the conceptual model and storage model of an Entity Framework Application is called

a. SSDL b. CSDL c. EDM d. MSL

10) Which of the following is NOT a type of entity?

a. POCO b. POCO Proxy c. EntityObject d. D: DBSet

11) Which of the following is NOT TRUE about the Entity Framework?

a. It automatically generates the classes from the model and updates these classes dynamically when the model is changed.

b. It takes care of database connectivity.

c. It provides query syntax for querying the model

d. It does not provide any mechanism to track changes to the model's objects.

12) Which of the following is responsible for change tracking management?

a. DBContextManager b. ObjectContextManager

c. ObjectStateManager d. EntityObjectManager

13) How to disable Lazy loading using DBContext?

a. myDBContext.Database.LazyLoadingEnabled = false;

b. myDBContext.Configuration.LazyLoadingEnabled = false;

c. myDBContext.LazyLoadingEnabled = false;

d. myDBContext.Students.LazyLoadingEnabled = false;

14) Which interface you have to implement to get the reference of ObjectContext from DBContext?

a. IObjectContextAdapter b. IDBContextAdapter

c. IEntityObjectContext d. IObjectContext

15) An API that can be used to configure a Code First model is called:

a. Fluent API b. CLR API c. POCO API d. T4 Template

21.MVC:-

1) MVC stands for _____.

a. Model, Vision & Control b. Model, View & Controller

c. Model, ViewData & Controller d. Model, Data & Controller

2) Which of following is TRUE?

a. The controller redirects incoming request to model.

b. The controller executes an incoming request.

c. The controller controls the data.

d. The controller render html to view.

3) The model is a _____.

a. Shape of data b. Html content

c. Collection of data d. Type of data.

4) Which of the following is a type of view in MVC?

a. Partial view b. Executable view

c. Data view d. Designer view

5) Which of the followings are Action Selectors?

a. ActionName b. NonAction

c. ActionVerbs d. All of the above

6) Which is the default http method for an action method?

a. HttpPost b. HttpGet

c. HttpPut d. HttpDelete

7) Which of the following view file types are supported in MVC?

a. cshtml b. vbhtml c. aspx d. All of the above

8) HtmlHelper class _____.

a. Generates html elements b. Generates html view

c. Generates html help file d. Generates model data

9) _____ attributes can be used for data validation in MVC.

a. DataAnnotations b. Fluent API

c. DataModel d. HtmlHelper

10) Which of the following view contains common parts of UI?

a. Partial view b. Html View

c. Layout view d. Razor view

11) How to transfer data from controller to view?

a. Using model object b. Using ViewBag

c. Using ViewData d. All of the above

12) TempData is useful to _____.

a. Transfer data from view to controller

b. Transfer data from one page to another page

c. Transfer data from controller to controller

d. Store data permanently.

13) What is action filters?

a. Action filter executes before and after action method executes.

b. Action filter executes before action method executes.

c. Action filter executes after action method executes.

d. Action filter executes parallel to action method.

14) Bundling allows _____.

a. Loading of multiple images in single request

b. Loading of multiple view files in single request.

c. Loading of caching of multiple script files

d. Loading of multiple script files in single request.

15) Which of the following is a default route pattern in MVC?

a."/{action}/{controller}/{id}" b."{controller}/{id}"

c."{controller}/{action}/{id}" d."{controller}/{action}"

16) Which of the following default class is used to configure all the routes in MVC?

a. FilterConfig b. RegisterRouteConfig

c. RouteConfig d. MVCRoutes

17) Which of the following method of html helper generates html control based on the data type of specified property?

a. Html.TextBox b. Html.Password

c. Html.Editor d. Html.Display

18) Which is the best approach to assign a session in MVC?

A) System.Web.HttpContext.Current.Session["LoginID"] =7;

B) Current.Session["LoginID"] =7;

C) Session["LoginID"] =7;

D) None

19) RedirectToActionPermanent() Method for which Status code represents?

A) 304 B) 302 C) 301 D) 300 E) None

20) RedirectToAction() Method for which Status code represents?

A) 304 B) 302 C) 301 D) 300 E) None

21) What is ActionResult() ?

A) It is an abstract Class B) It is a Concrete Class

C) Both A and B D) None

22) What is ViewResult() ?

A) It is an abstract Class B) It is a Concrete Class

C) Both A and B D) None

23) return View() works like in ASP.Net MVC C# as

A) Server.Transfer() B) Response.Redirect()

C) Both A and B D) None

24) RedirectToAction() works like in ASP.Net MVC C# as

A) Server.Transfer() B) Response.Redirect()

C) Both A and B D) None

25) In which format data can be return from XML into table ?

A) DataSet B) Datatable

C) A and B D) None

26) Can we use view state in MVC ?

A) Yes B) No C) Both A & B D) None

27) What Request Processing technique follows ASP.Net ?

A) Top-Down B) Down-Up C) Pipeline D) Water fall

28) What is DRY principle in ASP.Net ?

A) Don't repeat yourself. B) Don't revise yourself.

C) both a and b D) None

29) What is default authentication in Internet Information Services (IIS)?

A) Standard User B) Administrator C) Anonymous D) None

30) What is the extension of MVC view when using C#?

A) cshtml B) vbhtml C) None D) Both A & B

31) What is the extension of MVC view when using vb.net?

A) cshtml B) vbhtml

C) None D) Both A & B

32) How can you comment using Razor Syntax?

A) *@ Comment me *@

B) @* Comment me *@

C) @* Comment me @*

D) *@ Comment me @*

33) Which Namespace is used for Razor View Engine ?

A) System.Web.Razor B) System.Web.Mvc.WebFormViewEngine

C) Both A & B D) None

34) Which Namespace is used for ASPX View Engine ?

A) System.Web.Razor B) System.Web.Mvc.WebFormViewEngine

C) Both A & B D) None

35) The Razor View Engine uses to render server side content.

A) @ B) C) Both A & B D) None

36) The ASPX View Engine uses to render server side content.

A) @ B) <%=%> C) Both A & B D) None

37) Which is more faster between ASPX View Engine and Razor View Engine.

A) ASPX View Engine B) Razor View Engine

C) Both A & B D) None

38) Does Razor Engine supports for TDD ?

A) Yes B) No C) None

39) Does ASPX View Engine supports for TDD ?

A) Yes B) No C) None

40) How to Print value from Controller to View in MVC ?

A) ViewBag.ECMDetail = "my message"; and in view @ViewBag.ECMDetail

B) ViewBag.ECMDetail = "my message"; and in view ViewBag.ECMDetail

C) ViewBag.ECMDetail = "my message"; and in view ViewBag.Title

D) None

41) What are the advantages of using ASP.NET routing?

Answer: Clean URLs is originally brought from Ruby on Rails. http://www.technologycrowds.com?abc=10 , now clean URL in MVC ASP.Net will be work like http://www.technologycrowds.com/abc/10

42) What is the significance of ASP.NET routing?

Answer: Default Route Name: "{controller}/{action}/{id}", // URL with parameters By default routing is defined under Global.asax file. MVC ASP.Net uses routing to map between incoming browser request to controller action methods.

43) Can be it possible to share single view across multiple controllers in MVC?
Answer: We can put the view under shared folder, it will automatically view the across the multiple controllers.

44) Are MVC and Web API merged into one in MVC 6?

A) Yes B) No C) Both A & B D) None

45) Does MVC 6 introduced new JSON project based structure?

A) Yes B) No C) Both A & B D) None

46) Does MVC 6 allow only save change, hitting the save but then refreshing the browser to reflect changes?

A) Yes B) No C) Both A & B D) None

47) Does vNext is now Open Sourced via the .NET Foundation and open to public contributions.

A) Yes B) No C) Both A & B D) None

48) Can vNext runs on both Mac and Linux today (Mono Version)?

A) Yes B) No C) Both A & B D) None

49) What is the difference between MVC (Model View Controller) and MVP (Model View Presenter)?

Answer: MVC controller handles all the requests, MVP handles as the handler and also handles the all requests as well.

50) How does work Viewstart in MVC (ASP.Net)?

A) Viestart is used to layout of the application.

B) Viewstart is used like Masterpage in traditional forms (ASP.Net pages).

C) Viewstart render first in the views.

D) A, B and C.

E) None

51) Viewstart comes under which folder name ?

A) Views B) Account C) Shared D) Home

52) Does Viewstart override all Views layout/template under "Views" folder in MVC ?

A) Yes B) No C) Both A & B D) None

53) What is the name of default Viewstart Page in ASP.Net MVC ?

A) _ViewStart.cshtml B) _Layout.cshtml

C) _Login.cshtml D) None

54) Can we use third party View Engine using ASP.Net MVC Engine ?

Yes, below are the top five alternative

ASP.Net MVC View Engines

1. Spark (Castle MonoRail framework projects), Open Sourced, it is popular as MVCContrib library.

2. NHaml works like inline page templating.

3. NDjango uses F# Language.

4. Hasic uses VB.Net, XML.

5. Bellevue for ASP.NEt view, It respects HTML class first.

55) What is scaffolding using ASP.Net MVC Engine?

Answer: Scaffolding helps us to write CRUD operations blend using Entity Framework, It helps developer to write down simply even yet complex business logic.

56) What is life cycle in ASP.Net MVC Engine?

Step 1: Fill Route (Global.asax file will hit first).

Step 2: Fetch Route: It will gether information about controller and action to invoke.

Step 3: Request context

Step 4: Controller instance: it calls Controller class and method.

Step 5: Executing Action: It determines which action to be executed

Step 6: Result (View): Now Action method executed and returns back response to view in differentiating forms like Json, View Result, File Result etc.

57) Which is the way to render Partial View using ASP.Net MVC Razor Engine?
A) @Html.Partial("_PartialHeader")

B) @Html.PartialView("_PartialHeader")

C) @Html.PartialHtml("_PartialHeader")

D) B and C

E) None

58) Which Namespace is used to "Display" in Data Annotation using MVC ?

A) System.ComponentModel B) System.ComponentModel.DataAnnotations
C) Both A and B D) None

59) Which Namespaces are required to Data Annotation using MVC ?

A) System.ComponentModel B) System.ComponentModel.DataAnnotations
C) Both A and B D) None

60) Are both TempData/ViewData require typecasting in MVC?

A) Both (TempData/ViewData) requires type casting to avoid null exception.

B) No, these (TempData/ViewData) does not require type casting.

C) Both A) & B)

D) None

61) Is ViewBag slower than ViewData in MVC?

A) Yes B) No C) Both A) & B) D) None

62) Is ViewData faster than ViewBag in MVC?

A) Yes B) No C) Both A) & B) D) None

63) Are both TempData/ViewData property of Controller base class in MVC?

A) Yes B) No C) Both A) & B) D) None

64) Does TempData used to pass data from one page to another page in MVC?
A) Yes B) No C) Both A) & B) D) None

65) Can ASP.Net Web API specialize to XML or JSON ?

A) Yes B) No C) None

66) Does Web API (ASP.Net) supports to non SOAP based like XML or JSON ?

A) Yes B) No C) None

67) Does Web API (ASP.Net) supports to both version mobile apps and others ?
A) Yes B) No C) Both A & B D) None

68) Can ASP.Net Web API, it works HTTP standard verbs like POST, GET, PUT, DELETE (CRUD Operations) ?

A) Yes B) No C) Both A & B D) None

69) Can ASP.Net Web API ability to both self hosting (outside of IIS) and IIS ?

A) Yes B) No C) None

70) Can ASP.Net Web API has ability to transport non HTTP protocols like TCP, UDP, Named Pipes etc ?

A) Yes B) No C) None

71) What is AuthConfig.cs in ASP.Net MVC ?

A) AuthConfig.cs is used to configure route settings

B) AuthConfig.cs is used to configure security settings including sites oAuth Login.

C) None

D) All

72) What is BundleConfig.cs in ASP.Net MVC ?

A) BundleConfig.cs in MVC is used to register filters for different purposes.

B) BundleConfig.cs in MVC is used to register bundles used by the bundling and minification, serveral bundles are added by default like jQuery, jQueryUI, jQuery validation, Modernizr, default CSS references.

C) All

D) None

73) What is FilterConfig.cs in ASP.Net MVC ?

A) FilterConfig.cs is used to register global MVC filters, HandleErrorAttribute is registered by default filter. We can also register other filters.

B) FilterConfig.cs is used to register global MVC bundles.

C) None

D) All

74) What is RouteConfig.cs in ASP.Net MVC?

A) RouteConfig.cs is used to register MVC config statements, route config.

B) RouteConfig.css is used to register global MVC bundles.

C) None

D) All

75) What is the difference between HtmlTextbox and HtmlTextboxFor using ASP.Net MVC Razor Engine?

A) @Html.TextBox is not strongly typed, @Html.TextBoxFor is strongly typed that is why should be use @Html.TextBoxFor in MVC Razor Engine.

B) @Html.TextBox is strongly typed, @Html.TextBoxFor is not strongly typed that is why should be use @Html.TextBox in MVC Razor Engine.

C) None

D) Both A and B

76) What is the benefits of Html.RenderPartial using ASP.Net MVC Razor Engine?

A) @Html.RenderPartial Returns response, moreover requires to create action.
B) @Html.RenderPartial Returns nothing (void), it is faster than @Html.Partial, moreover requires not to create action.

C) None D) Both A and B

77) What is the benefits of Html.Partial using ASP.Net MVC Razor Engine?

A) @Html.RenderPartial Returns response, moreover requires to create action.
B) @Html.RenderPartial Returns string value, it is slower than @Html.RenderPartial, moreover requires not to create action.

C) None D) Both A and BSyntax@Html.Partial(" viewname");

78) How to check Request coming from which controller using MVC ASP.Net?
A) var _controller = HttpContext.Current.Request.RequestContext.Values["Controller"].ToString();
B) var _controller = HttpContext.Current.Request.RequestContext.RouteData.Values["Controller"].ToString();

C) var _controller = RouteData.Values["Controller"].ToString();

D) None

79) For which ModelState.IsValid Validate ?

A) It checks for Entityframework Model state.

B) It checks for valid Model State using DataAnnotations.

C) It checks for SQL database state.

D) None

80) Which Name space is used to create chart using ASP.Net MVC?

A) using System.Web.MVC; B) using System.Web.Helpers;

c) using System.Web.Chart; D) All

81) How can we write Chart output to MVC View?

A) .Write(bmp); B) Write("bmp");

C) .Write("bmp"); D) All

82) Which name space using can send email in ASP.Net MVC?

A) using System.Net.Mail; B) using System.Net;

C) using System.Mail; D) None

83) If Razor View Engine need to add JQuery function and contain @ special character then how we can write it in Razor View?

A) Replace @ to @@@ (tripple)

B) Replace @ to @@ (double)

C) None

D) Both (A & B)

84) How to set Default Value to Hidden Input Box using ASP.Net MVC?

A) @Html.HiddenFor(m => m.Name, "Jack")

B) @Html.HiddenFor(m => m.Name, new { Value = "Jack"})

C) @Html.Hidden(m => m.Name, new { Value = "Jack"})

D) None

85) How to check all errors of Model using ASP.Net MVC?

A) var errors = Model.Values.SelectMany(v => v.Errors);

B) var errors = ModelState.SelectMany(v => v.Errors);

C) var errors = ModelState.Values.SelectMany(v => v.Errors);

D) None

86) AuthConfig.cs file is under in which App folder ?

A) App_Data B) App_Start C) Content D) Filters

87) BundleConfig.cs file is under in which App folder ?

A) App_Data B) App_Start C) Content D) Filters

88) FilterConfig.cs file is under in which App folder ?

A) App_Data B) App_Start C) Content D) Filters

89. RouteConfig.cs file is under in which App folder ?

A) App_Data B) App_Start C) Content D) Filters

90) WebApiConfig.cs file is under in which App folder ?

A) App_Data B) App_Start C) Content D) Filters

91) Can you list the main types of result using ASP.Net MVC?

ANSWER:-

There are total 10 main types of result, ActionResult is main type and others are sub types of results as listed below:

1• System.Web.Mvc.ActionResult

2• System.Web.Mvc.ContentResult

3• System.Web.Mvc.EmptyResult

4• System.Web.Mvc.FileResult

5• System.Web.Mvc.HttpStatusCodeResult

6• System.Web.Mvc.JavaScriptResult

7• System.Web.Mvc.JsonResult

8• System.Web.Mvc.RedirectResult

9• System.Web.Mvc.RedirectToRouteResult

10• System.Web.Mvc.ViewResultBase

92) Which filter will be execute at first using ASP.Net MVC?

A) Action filters B) Authorization filters

C) Response filters D) Exception filters

93) Which filter will be execute at last using ASP.Net MVC?

A) Action filters B) Authorization filters

C) Exception filters D) Response filters

22.WCF:-

1. Which of the following is NOT true?

A) A WCF Service can be consumed by Windows applications

B) A WCF Service can be consumed by Web applications

C) A WCF Service can perform calculations

D) A WCF Service cannot return a dataset

2. WCF services can communicate with _____.

A) all programming languages

B) XML

C) only the languages included with Visual Studio .NET

D) multiple platforms and multiple languages

3. The standard method for storing data that can be transferred easily from one machine or platform to another is _____.

A) XML B) SOAP C) WSDL D) WCF

4. One of the advantages of using _____ is that data are transmitted in a text format rather than a binary format.

A) XML B) SOAP C) WSDL D) WCF

5. Data that is in _____ format can pass through many firewalls that _____ cannot penetrate.

A) binary, text B) text, binary C) SOAP, WCF D) WCF, SOAP

6. _____ is a popular standard that includes a set of rules for handling requests and responses including class names, method names, and parameters.

A) XML B) WCF C) WSDL D) SOAP

7. The information about the names of the methods, the parameters that can be passed, and the values that are returned from the functions is controlled in some Web services by a description specified in _____.

A) XML B) SOAP C) WSDL D) WCF

8. Always end your URI (or URL) with a _____ to avoid an extra trip to the server to determine that it is a site rather than a directory.

A) hyphen B) slash C) backslash D) double slash

9. A resource on the Web is uniquely identified by its URI, which means _____.

A) Uniform Resource Identifier B) Universal Registered Identifier

C) Uniform Registered Identifier D) Universal Resource Identifier

10. To add a WCF Service, select the solution name in the Solution Explorer and select _____ from File menu.

A) Add / New Solution B) Add / New Web Site

C) Add / New Service D) Add / New Library

11. You have created a new service based on Windows Communication Foundation and also a client application to test the service. You want add an endpoint in the web.config file of the client application to use the new service. Which values should you include in you service element of the web.config file?
A. Address B. Contract C. Binding D. All of these

12. A service contract specifies what an endpoint communicates to the outside world. At a more concrete level, it is a statement about a set of specific messages organized into basic message exchange patterns (MEPs), such as

request/reply, one-way, and duplex. Which of the following is NOT part of the Service Contract specification:

A. The data types of messages

B. The specific protocols and serialization formats

C. The location of the operations

D. The frequency of messages per second

13. Services are groups of operations. To create a service contract you usually model operations and specify their grouping. In Windows Communication Foundation (WCF) applications, developers define the operations by creating a method and marking it with the which attribute?

A. ServiceContractAttribute B. DataMemberAttribute

C. DataContractAttribute D. OperationContractAttribute

14. You've created a new class in your .NET project that contains a wide variety of operations grouped together to form part of a Server Contract. You now need to add an attribute to the class to define it as a service contract. Which attribute should you use?

A. OperationContractAttribute B. DataMemberAttribute

C. DataContractAttribute D. ServiceContractAttribute

15. Both classes and interfaces represent a grouping of functionality and, therefore, both can be used to define a WCF service contract. However, it is recommended that you use interfaces because they directly model service contracts. Without an implementation, interfaces do no more than define a grouping of methods with certain signatures. Which of the following is a benefit of using interfaces to define Service Contracts?

A. Service contract interfaces can extend any number of other service contract interfaces.

B. You can modify the implementation of a service contract by changing the interface implementation, while the service contract remains the same

C. A single class can implement any number of service contracts by implementing those service contract interfaces.

D. All of these

16. You have created a new class which will be the basis for a Service Contract. You have used ServiceContractAttribute and OperationContractAttribute to decorate the class and the methods. Which of the following is NOT an advantage of using classes instead of interfaces for Service Contracts?

A. Speed

B. All of these are disadvantages

C. Simplicity

D. Multiple Inheritance

17. Which of the following is TRUE regarding service operations and references to objects?

A. Objects must be serializable

B. You can't return values from service operations

C. Objects are passed as references

D. You can't pass parameters to service operations

18. You've created a new class and decorated it with the DataContractAttribute so that it forms a Data Contract for WCF. This class contains several attributes that you want to make available as part of the Data Contract. Currently these attributes are declared as private. What do you need to do to ensure these attributes are serializable?

A. Add the DataMemberAttribute and change the type to public

B. Add the DataContractAttribute to the attribute

C. Change the type to internal

D. Add the DataMemberAttribute or change the type to public

19. A developer has designed a service that contains a method called TakeAction which is decorated with the following attribute: [OperationContractAttribute(IsOneWay=true)] Another client application will invoke the TakeAction operation and continue processing after WCF writes the message to the network. What must the developer of the TakeAction method do to ensure the client action can call this method?

A. Use object as the return type

B. Use FaultException as the return type

C. Remove all parameters from the method signature

D. Use void as the return type

20. The signature of a service operation dictates a certain underlying message exchange pattern (MEP) that can support the data transfer and the features an operation requires. You want to adopt a pattern that supports the sending and receiving of messages by both the service and client. Which patter should you choose?

A. one-way B. none of these C. request/reply D. duplex

21. Study the following line of code: OperationContext.Current.GetCallbackChannel(); The ICalendarDuplexCallback interface is defined as the CallbackContract property in the Service Contract. In which class should you use this line of code?

A. Client B. You should never use this C. Both D. Service

22. A client application interacts with a new Service that calculates interest rates for the banks customers. The Service Contract contains BasicHttpBinding as the binding type in the endpoint configuration. The service contains some methods that return sensitive information such as customers names and addresses. You want to ensure that these methods are encrypted. What should you do?

A. Set the ProtectionLevel to None in the ServiceContractAttribute

B. Set the ProtectionLevel in the OperationContractAttribute to EncryptAndSign for each of the sensitive methods

C. Nothing, all messages will be encrypted and signed already

D. Set the ProtectionLevel to Sign in the ServiceContractAttribute

23. Which of the following is FALSE regarding the WSHttpBinding class?

A. Provides WS-Addressing

B. Provides un-encrypted messages by default

C. Provides reliable messaging

D. Provides transactions

24. In Windows Communication Foundation (WCF) applications, which of the following is FALSE regarding Sessions?

A. Messages delivered during a session are processed in the order in which they are received

B. They are explicitly initiated and terminated by the receiving application

C. There is no general data store associated with a WCF session

D. Sessions correlate a group of messages into a conversation

25. The instancing behaviour (set by using the System.ServiceModel.ServiceBehaviorAttribute.InstanceContextMode property) controls how the InstanceContext is created in response to incoming messages. You have created a new WCF service and set the InstanceContextMode to PerCall. What is the behaviour of the InstanceContext in this mode?

A. A new InstanceContext is created for each call

B. A new InstanceContext is created for each channel

C. A new InstanceContext is created for all calls

D. A new InstanceContext is never created

26. When configuring a WCF service using Visual Studio, you can use either a Web.config file or an App.config file to specify the settings. The choice of the configuration file name is determined by the hosting environment you choose for the service. Where does the endpoint configuration element lie in a .NET configuration file?

A. System.ServiceModel – bindings – endpoint

B. System.ServiceModel – services – service – endpoint

C. System.ServiceModel – endpoint

D. System.ServiceModel – behaviors – behavior – endpoint

27. The System.ServiceModel.Channels namespace contains the DeliveryFailure enumeration. DeliveryFailure specifies the possible types of delivery failure for a message read from the queue. Which of the following elements is a valid DeliveryFailure?

A. BadSignature B. AccessDenied C. ReceiveTimeout D. All of these

28. Which class in WCF represents the unit of communication between endpoints in a distributed environment?

A. RequestContext B. Message C. Binding D. ChannelBase

28. Windows Communication Formats (WCF) is Microsoft's technology for communicating between applications on the same computer system, on a network, or across the

A) True B) False

29. It is possible for a single application to be both a client and a service.

A) True B) False

30. WCF cannot communicate with other platforms that support SOAP and simple XML.

A) True B) False

31. An endpoint indicates where messages can be sent (address).

A) True B) False

32. Data that is in binary format can pass through any firewall.

A) True B) False

33. WSDL contains information about the names of the methods, the parameters that can be passed, and the values that are returned from the functions.

A) True B) False

34. For technical specifications, the industry standard term URL is preferred to URI.

A) True B) False

35. The transport protocol used by SOAP is HTTP.

A) True B) False

36. To rename a Web Service you need to change only the name in the Solution Explorer.

A) True B) False

37. When a new project is added to a WCF Service solution, the projects are saved independently.

A) True B) False

TOP 500 +IMP INTERVIEW QUESTIONS:-

"MICROSOFT .NET"

"MICROSOFT .NET _INTERVIEW'S QUESTIONS

→REFERENCE: JAVAT POINT:-

→1.NET Framework:-

.NET is a framework to develop software applications. It is designed and developed by Microsoft and the first beta version released in 2000.

It is used to develop applications for web, Windows, phone. Moreover, it provides a broad range of functionalities and support.

This framework contains a large number of class libraries known as Framework Class Library (FCL). The software programs written in .NET are executed in the execution environment, which is called CLR (Common Language Runtime). These are the core and essential parts of the .NET framework.

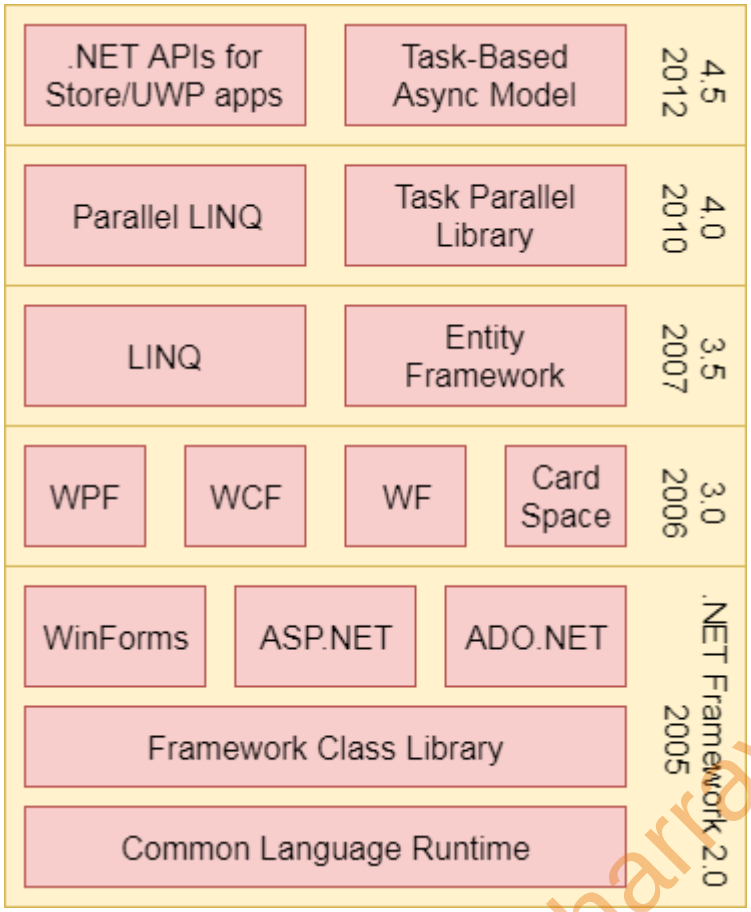
This framework provides various services like memory management, networking, security, memory management, and type-safety.

The .Net Framework supports more than 60 programming languages such as C#, F#, VB.NET, J#, VC++, JScript.NET, APL, COBOL, Perl, Oberon, ML, Pascal, Eiffel, Smalltalk, Python, Cobra, ADA, etc.

Following is the .NET framework Stack that shows the modules and components of the Framework.

The .NET Framework is composed of four main components:

1. Common Language Runtime (CLR)
2. Framework Class Library (FCL),
3. Core Languages (WinForms, ASP.NET, and ADO.NET), and
4. Other Modules (WCF, WPF, WF, Card Space, LINQ, Entity Framework, Parallel LINQ, Task Parallel Library, etc.)

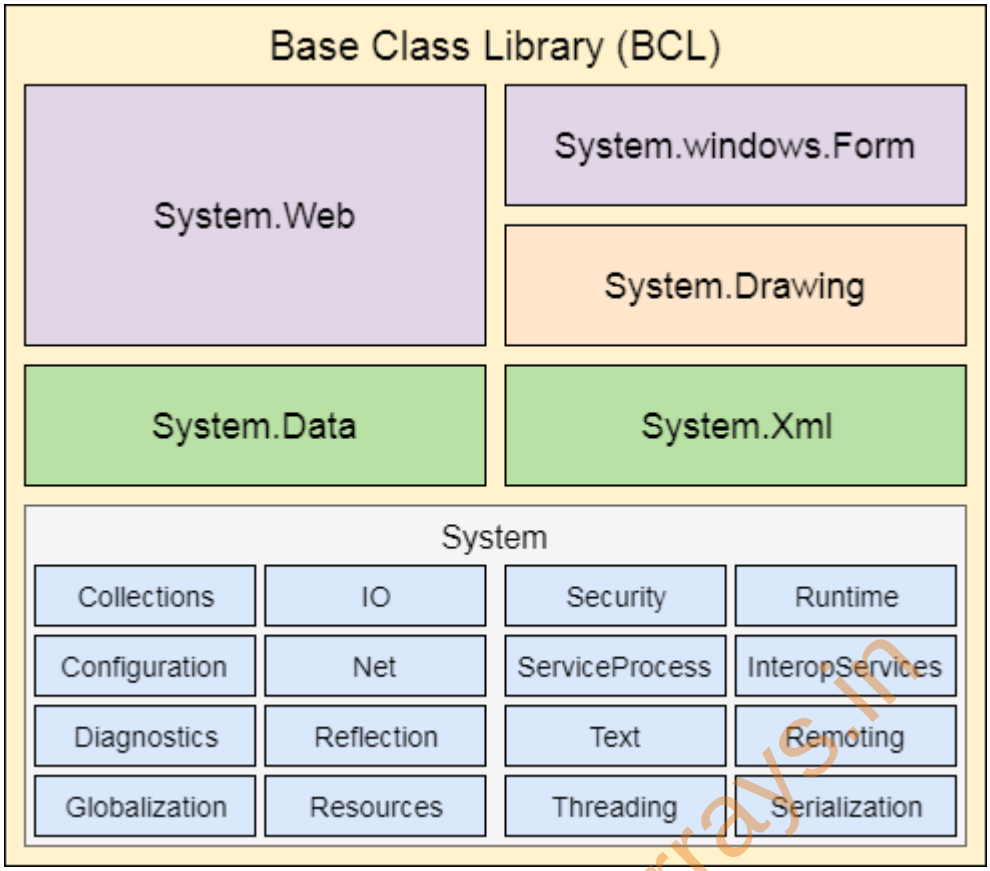


CLR (Common Language Runtime)

It is a program execution engine that loads and executes the program. It converts the program into native code. It acts as an interface between the framework and operating system. It does exception handling, memory management, and garbage collection. Moreover, it provides security, type-safety, interoperability, and portablility. A list of CLR components are given below:

FCL (Framework Class Library)

It is a standard library that is a collection of thousands of classes and used to build an application. The BCL (Base Class Library) is the core of the FCL and provides basic functionalities.



WinForms

Windows Forms is a smart client technology for the .NET Framework, a set of managed libraries that simplify common application tasks such as reading and writing to the file system.

ASP.NET

ASP.NET is a web framework designed and developed by Microsoft. It is used to develop websites, web applications, and web services. It provides a fantastic integration of HTML, CSS, and JavaScript. It was first released in January 2002.

ADO.NET

ADO.NET is a module of .Net Framework, which is used to establish a connection between application and data sources. Data sources can be such as SQL Server and XML. ADO .NET consists of classes that can be used to connect, retrieve, insert, and delete data.

WPF (Windows Presentation Foundation)

Windows Presentation Foundation (WPF) is a graphical subsystem by Microsoft for rendering user interfaces in Windows-based applications. WPF, previously known as "Avalon", was initially released as part of .NET Framework 3.0 in 2006. WPF uses DirectX.

WCF (Windows Communication Foundation)

It is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another.

WF (Workflow Foundation)

Windows Workflow Foundation (WF) is a Microsoft technology that provides an API, an in-process workflow engine, and a rehostable designer to implement long-running processes as workflows within .NET applications.

LINQ (Language Integrated Query)

It is a query language, introduced in .NET 3.5 framework. It is used to make the query for data sources with C# or Visual Basics programming languages.

Entity Framework

It is an ORM based open source framework which is used to work with a database using .NET objects. It eliminates a lot of developers effort to handle the database. It is Microsoft's recommended technology to deal with the database.

Parallel LINQ

Parallel LINQ or PLINQ is a parallel implementation of LINQ to objects. It combines the simplicity and readability of LINQ and provides the power of parallel programming.

It can improve and provide fast speed to execute the LINQ query by using all available computer capabilities.

Apart from the above features and libraries, .NET includes other APIs and Model to improve and enhance the .NET framework.

→1.NET INTERVIEW QUESTIONS:-

→REFERENCE:-INTERVIEW BIT:-

Introduction to .NET and .NET Core Framework

.NET framework is developed by Microsoft, provides an environment to run, debug and deploy code onto web services and applications by using tools and functionalities like libraries, classes, and APIs. This framework uses object-oriented programming.

You can use different languages like C#, Cobol, VB, F#, Perl, etc. for writing .NET framework applications. This Framework supports services, websites, desktop applications, and many more on Windows. It provides functionalities such as generic types, automatic memory management, reflection, concurrency, etc. These functionalities will help to make the development easier and efficiently build high-quality web as well as client applications.

.NET Core is a newer version of the .NET framework and it is a general-purpose, cost-free, open-source development platform developed by Microsoft. .NET Core is a cross-platform framework that runs an application on different operating systems such as Windows, Linux, and macOS operating systems. This framework can be used to develop various kinds of applications like mobile, web, IoT, cloud, microservices, machine learning, game, etc.

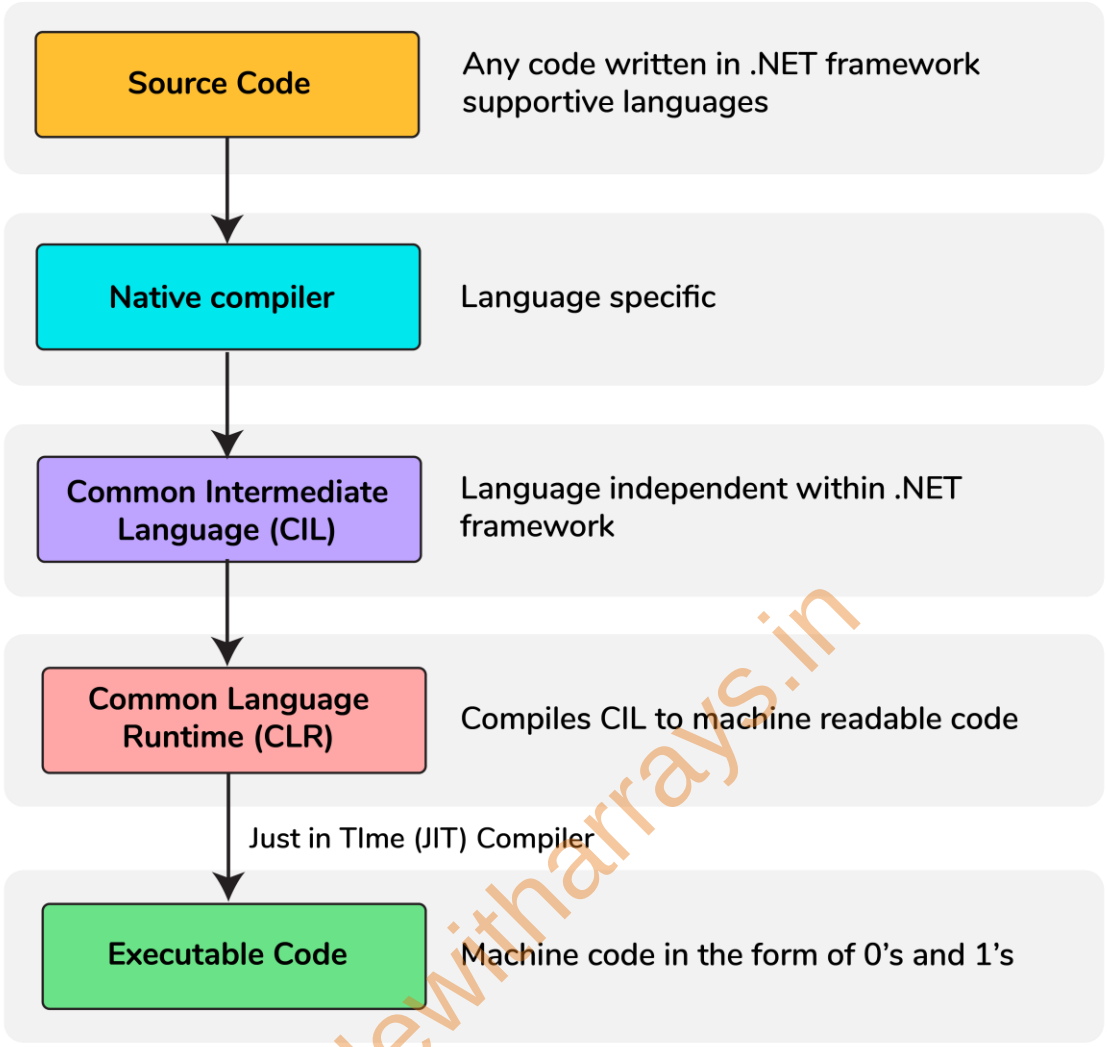
Characteristics of .NET Core:

- **Free and open-source**: .NET Core source code project can be obtained from Github. It is free and licensed under the MIT and Apache licenses.
- **Cross-platform**: .NET Core is supported by different operating systems like Windows, macOS, and Linux.
- **Sharable**: A single consistent API model that is written in .NET Standard will be used by .NET Core and is common for all the .NET applications. The same library or API can be used on multiple platforms with different languages.
- **Friendly**: The .NET Core is compatible with .NET Framework, Mono, and Xamarin, through .NET Standard. It also supports working with different Web frameworks and libraries such as Angular, React, and JavaScript.
- **Fast**: .NET Core 3.0 is faster compared to the .NET Framework, .NET Core 2.2 and previous versions. It is also much faster than other server-side frameworks like Node.js and Java Servlet.

→1.NET Interview Questions:-

1. How does the .NET framework work?

- .NET framework-based applications that are written in supportive languages like C#, F#, or Visual basic are compiled to Common Intermediate Language (CIL).
- Compiled code is stored in the form of an assembly file that has a .dll or .exe file extension.
- When the .NET application runs, Common Language Runtime (CLR) takes the assembly file and converts the CIL into machine code with the help of the Just In Time(JIT) compiler.
- Now, this machine code can execute on the specific architecture of the computer it is running on.



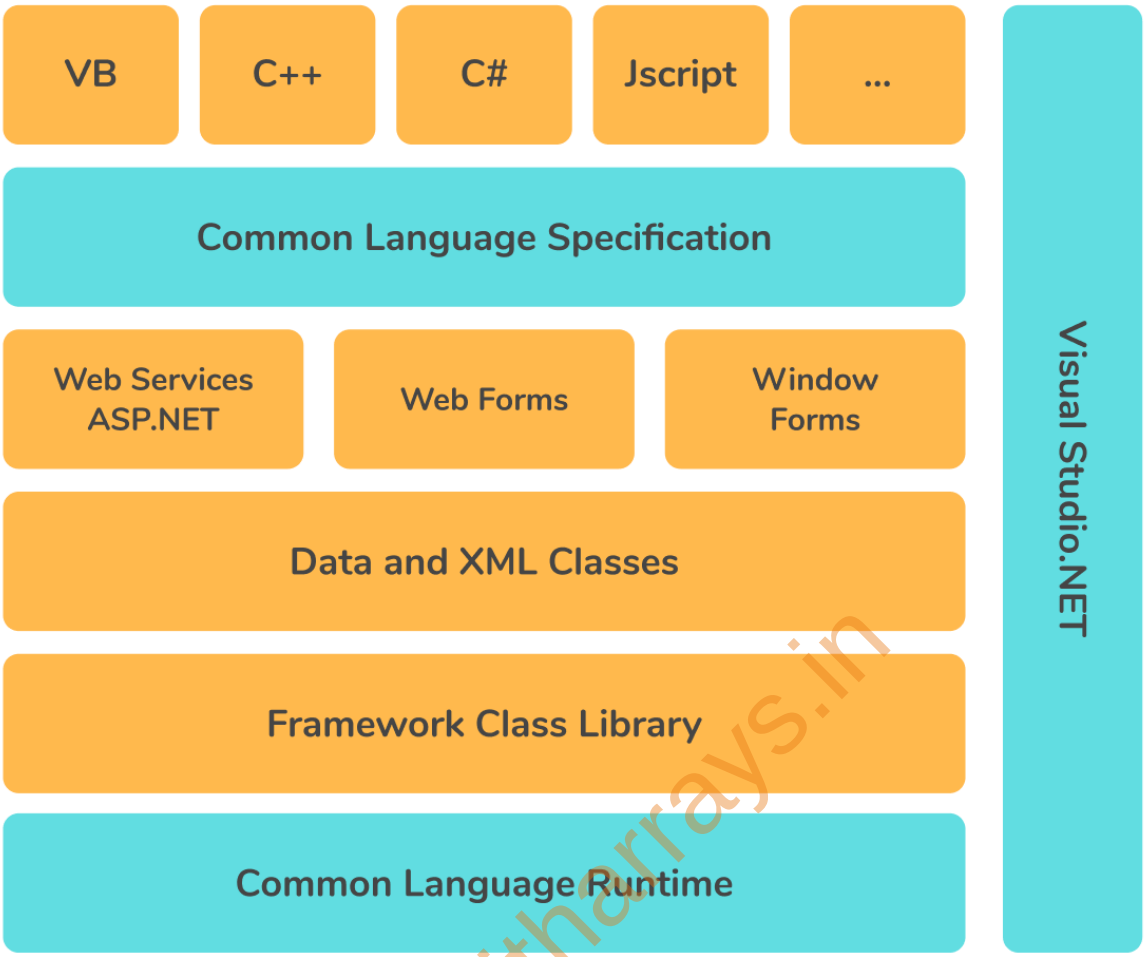
How .NET works?

2. Explain about major components of the .NET framework.

The major components .NET framework are given below:

- **Common Language Runtime(CLR):**
  - It is an execution engine that runs the code and provides services that make the development process easier.
  - Services provided by CLR are memory management, garbage collection, type safety, exception handling, security, and thread management. It also makes it easier for designing the applications and components whose objects interact across the languages.
  - The programs written for the .NET Framework are executed by the CLR regardless of programming language. Every .NET Framework version is having CLR.
- **Framework Class Library(FCL):**
  - It has pre-defined methods and properties to implement common and complex functions that can be used by .NET applications. It will also provide types for dates, strings, numbers, etc.

    - This class library includes APIs for database connection, file reading and writing, drawing, etc.
- **Base Class Library(BCL):**
    - The Base Class Library(BCL) has a huge collection of libraries features and functions that are helpful in implementing various programming languages such as C#, F#, Visual C++, etc., in the .NET Framework.
    - BCL is divided into two parts. They are:
        - **User-defined class library:** It includes Assemblies.
            - *Assembly*: A .NET assembly is considered as the major building block of the .NET Framework. An assembly in the CLI(Common Language Infrastructure) is a logical unit of code, which is used for security, deployment, and versioning. Assembly can be defined in two forms namely Dynamic Link Library(.dll) and executable(.exe) files.
              When compilation of the .NET program takes place, metadata with Microsoft Intermediate Language(MSIL) will be generated and will be stored in a file called Assembly.
        - **Predefined class library:** It contains namespace.
            - *Namespace:* It is the collection of pre-defined methods and classes that are present in the .Net Framework. A namespace can be added to a .NET program with the help of "using system", where using represents a keyword and system represents a namespace.
- **Common Type System(CTS):**
    - CTS specifies a standard that will mention which type of data and value can be defined and managed in memory during runtime.
    - It will make sure that programming data defined in different languages should interact with each other for sharing the information. For example, in VB.NET we define datatype as integer, while in C# we define int as a data type.
    - It can be used to prevent data loss when you are trying to transfer data from a type in one language to its equivalent type in another language.
- **Common Language Specification (CLS):**
    - Common Language Specification (CLS) is a subset of CTS and defines a set of rules and regulations to be followed by every .NET Framework's language.
    - A CLS will support inter-operability or cross-language integration, which means it provides a common platform for interacting and sharing information. For example, every programming language(C#, F#, VB .Net, etc.) under the .NET framework has its own syntax. So when statements belonging to different languages get executed, a common platform will be provided by the CLS to interact and share the information.



InterviewBit

3. What is an EXE and a DLL?

EXE and DLLs are assembly executable modules.

EXE is an executable file that runs the application for which it is designed. An EXE is produced when we build an application. Therefore the assemblies are loaded directly when we run an EXE. However, an EXE cannot be shared with the other applications.

Dynamic Link Library (DLL) is a library that consists of code that needs to be hidden. The code is encapsulated inside this library. An application can consist of many DLLs which can be shared with the other programs and applications.

You can download a PDF version of Dot Net Interview Questions.

Download PDF

4. What is CTS?

CTS stands for Common Type System. It follows a set of structured rules according to which a data type should be declared and used in the program code. It is used to describe all the data types that are going to be used in the application.

We can create our own classes and functions by following the rules in the CTS. It helps in calling the data type declared in one programming language by other programming languages.

5. Explain CLS

Common Language Specification (CLS) helps the application developers to use the components that are inter-language compatible with certain rules that come with CLS. It also helps in reusing the code among all of the .NET-compatible languages.

6. What is JIT?

JIT stands for **Just In Time**. It is a compiler that converts the intermediate code into the native language during the execution.

7. What is the difference between int and Int32?

There is no difference between int and Int32. Int32 is a type provided by the .NET framework class whereas int is an alias name for Int32 in the C# programming language.

8. Explain the differences between value type and reference type.

The main differences between value type and reference type are given below:

- A Value Type holds the actual data directly within the memory location and a reference type contains a pointer which consists of the address of another memory location that holds the actual data.
- Value type stores its contents on the stack memory and reference type stores its contents on the heap memory.
- Assigning a value type variable to another variable will copy the value directly and assigning a reference variable to another doesn't copy the value, instead, it creates a second copy of the reference.
- Predefined data types, structures, enums are examples of value types. Classes, Objects, Arrays, Indexers, Interfaces, etc are examples of reference types.

9. What is the difference between managed and unmanaged code?

The main difference between managed and unmanaged code is listed below:

Managed Code	Unmanaged Code
It is managed by CLR.	It is not managed by CLR.
.NET framework is a must for execution.	Does not require a .NET framework for the execution.
Memory management is done through garbage collection.	Runtime environment takes care of memory management.

10. Explain Microsoft Intermediate Language

MSIL is the Microsoft Intermediate Language, which provides instructions for calling methods, memory handling, storing and initializing values, exception handling, and so on.

The instructions provided by MSIL are platform-independent and are generated by the language-specific compiler from the source code. JIT compiler compiles the MSIL into machine code based on the requirement.

11. What is an assembly?

An assembly is a file that is automatically generated by the compiler which consists of a collection of types and resources that are built to work together and form a logical unit of functionality. We can also say, assembly is a compiled code and logical unit of code.

Assemblies are implemented in the form of executable (.exe) or dynamic link library (.dll) files.

12. Is ASP.NET different from ASP? If yes, explain how?

Yes, ASP.NET and ASP(Active Server Pages) both are different. Let's check how they are different from each other.

- ASP.NET uses .NET languages such as C# and VB.NET, which are compiled to Microsoft Intermediate Language (MSIL). ASP uses VBScript. ASP code is interpreted during the execution.
- ASP.NET which is developed by Microsoft is used to create dynamic web applications while ASP is Microsoft's server-side technology used to create web pages.
- ASP.NET is fully object-oriented but ASP is partially object-oriented.
- ASP.NET has full XML Support for easy data exchange whereas ASP has no built-in support for XML.
- ASP.NET uses the ADO.NET technology to connect and work with databases. ASP uses ADO technology.

13. Explain role-based security in .NET

Role-based security is used to implement security measures in .NET, based on the roles assigned to the users in the organization. In the organization, authorization of users is done based on their roles.

For example, windows have role-based access like administrators, users, and guests.

14. Explain the different types of assembly.

Assemblies are classified into 2 types. They are:

Private Assembly:

- It is accessible only to the application.
- We need to copy this private assembly, separately in all application folders where we want to use that assembly. Without copying, we cannot access the private assembly.
- It requires to be installed in the installation directory of the application.

Shared or Public Assembly:

- It can be shared by multiple applications.
- Public assembly does not require copying separately into all application folders. Only one copy of public assembly is required at the system level, we can use the same copy by multiple applications.
- It is installed in the Global Assembly Cache(GAC).

15. What is the order of the events in a page life cycle?

There are eight events as given below that take place in an order to successfully render a page:

- Page_PreInit
- Page_Init
- Page_InitComplete
- Page_PreLoad
- Page_Load
- Page_LoadComplete
- Page_PreRender
- Render

16. What is a garbage collector?

Garbage collector frees the unused code objects in the memory. The memory heap is partitioned into 3 generations:

- Generation 0: It holds short-lived objects.
- Generation 1: It stores medium-lived objects.
- Generation 2: This is for long-lived objects.

Collection of garbage refers to checking for objects in the generations of the managed heap that are no longer being used by the application. It also performs the necessary operations to reclaim their memory. The garbage collector must perform a collection in order to free some memory space.

During the garbage collection process:

- The list of live objects is recognized.
- References are updated for the compacted objects.
- The memory space occupied by dead objects is recollected. The remaining objects are moved to an older segment.

`System.GC.Collect()` method is used to perform garbage collection in .NET.

17. What is caching?

Caching means storing the data temporarily in the memory so that the data can be easily accessed from the memory by an application instead of searching for it in the original location. It increases the speed and performance efficiency of an application.

There are three types of caching:

- Page caching
- Data caching
- Fragment caching

18. Can we apply themes to ASP.NET applications?

Yes. By modifying the following code in the `web.config` file, we can apply themes to ASP.NET applications:

```
<configuration>
   <system.web>
       <pages theme="windows"/>
   </system.web>
</configuration>
```

19. Explain MVC.

MVC stands for Model View Controller. It is an architecture to build .NET applications. Following are three main **logical components of MVC**: the model, the view, and the controller.

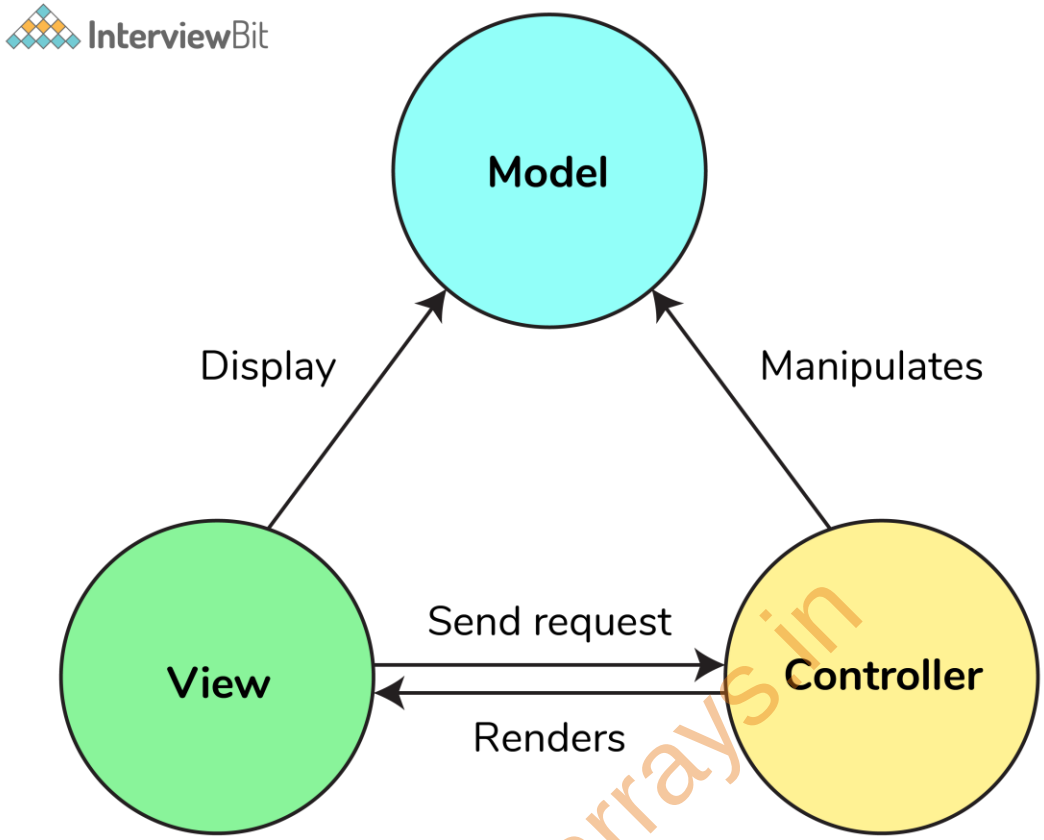


Fig: Components of MVC

Components of MVC

Model: They hold data and its related logic. It handles the object storage and retrieval from the databases for an application. For example:
A Controller object will retrieve the employee information from the database.
It manipulates employee data and sends back to the database or uses it to render the same data.

View: View handles the UI part of an application. They get the information from the models for their display. For example, any employee view will include many components like text boxes, dropdowns, etc.

Controller: They handle the user interactions, figure out the responses for the user input and also render the final output. For instance, the Employee controller will handle all the interactions and inputs from the Employee View and update the database using the Employee Model.

20. What is cross-page posting?

Whenever we click on a submit button on a webpage, the data is stored on the same page. But if the data is stored on a different page and linked to the current one, then it is known as a cross-page posting. Cross-page posting is achieved by `POSTBACKURL` property.

To get the values that are posted on this page to which the page has been posted, the FindControl method can be used.

21. What is a delegate in .NET?

A delegate is a .NET object which defines a method signature and it can pass a function as a parameter.

Delegate always points to a method that matches its specific signature. Users can encapsulate the reference of a method in a delegate object.

When we pass the delegate object in a program, it will call the referenced method. To create a custom event in a class, we can make use of delegate.

22. What are security controls available on ASP.NET?

Following are the five security controls available on ASP.NET:

- `<asp: Login>` Provides a login capability that enables the users to enter their credentials with ID and password fields.
- `<asp: LoginName>` Used to display the user name who has logged-in.
- `<asp: LoginView>` Provides a variety of views depending on the template that has been selected.
- `<asp: LoginStatus>` Used to check whether the user is authenticated or not.
- `<asp: PasswordRecovery>` Sends an email to a user while resetting the password.

23. What is boxing and unboxing in .NET?

Boxing is the process of converting a value type into a reference type directly. Boxing is implicit.

Unboxing is the process where reference type is converted back into a value type. Unboxing is explicit.

An example is given below to demonstrate boxing and unboxing operations:

```
int a = 10;       // a value type
object o = a;      // boxing
int b = (int)o;    // unboxing
```

24. What is MIME in .NET?

MIME stands for Multipurpose Internet Mail Extensions. It is the extension of the e-mail protocol which lets users use the protocol to exchange files over emails easily.

Servers insert the MIME header at the beginning of the web transmission to denote that it is a MIME transaction.

Then the clients use this header to select an appropriate ‘player’ for the type of data that the header indicates. Some of these players are built into the web browser.

25. What is the use of manifest in the .NET framework?

Manifest stores the metadata of the assembly. It contains metadata which is required for many things as given below:

- Assembly version information.
- Scope checking of the assembly.
- Reference validation to classes.
- Security identification.

26. Explain different types of cookies available in ASP.NET?

Two types of cookies are available in ASP.NET. They are:

- **Session Cookie:** It resides on the client machine for a single session and is valid until the user logs out.
- **Persistent Cookie:** It resides on the user machine for a period specified for its expiry. It may be an hour, a day, a month, or never.

27. What is the meaning of CAS in .NET?

Code Access Security(CAS) is necessary to prevent unauthorized access to programs and resources in the runtime. It is designed to solve the issues faced when obtaining code from external sources, which may contain bugs and vulnerabilities that make the user's system vulnerable.

CAS gives limited access to code to perform only certain operations instead of providing all at a given point in time. CAS constructs a part of the native .NET security architecture.

28. What is the appSettings section in the web.config file?

We can use the appSettings block in the web.config file, if we want to set the user-defined values for the whole application. Example code given below will make use of ConnectionString for the database connection throughout the project:

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

29. What is the difference between an abstract class and an interface?

The main difference between an abstract class and an interface are listed below:

Abstract Class	Interface
Used to declare properties, events, methods, and fields as well.	Fields cannot be declared using interfaces.
Provides the partial implementation of functionalities that must be implemented by inheriting classes.	Used to declare the behavior of an implementing class.
Different kinds of access modifiers like private, public, protected, etc. are supported.	Only public access modifier is supported.
It can contain static members.	It does not contain static members.
Multiple inheritances cannot be achieved.	Multiple inheritances are achieved.

30. What are the types of memories supported in the .NET framework?

Two types of memories are present in .NET. They are:

Stack: Stack is a stored-value type that keeps track of each executing thread and its location. It is used for static memory allocation.

Heap: Heap is a stored reference type that keeps track of the more precise objects or data. It is used for dynamic memory allocation.

31. Explain localization and globalization.

Localization is the process of customizing our application to behave as per the current culture and locale.

Globalization is the process of designing the application so that it can be used by users from across the globe by supporting multiple languages.

32. What are the parameters that control the connection pooling behaviors?

There are 4 parameters that control the connection pooling behaviours. They are:

- Connect Timeout
- Min Pool Size
- Max Pool Size
- Pooling

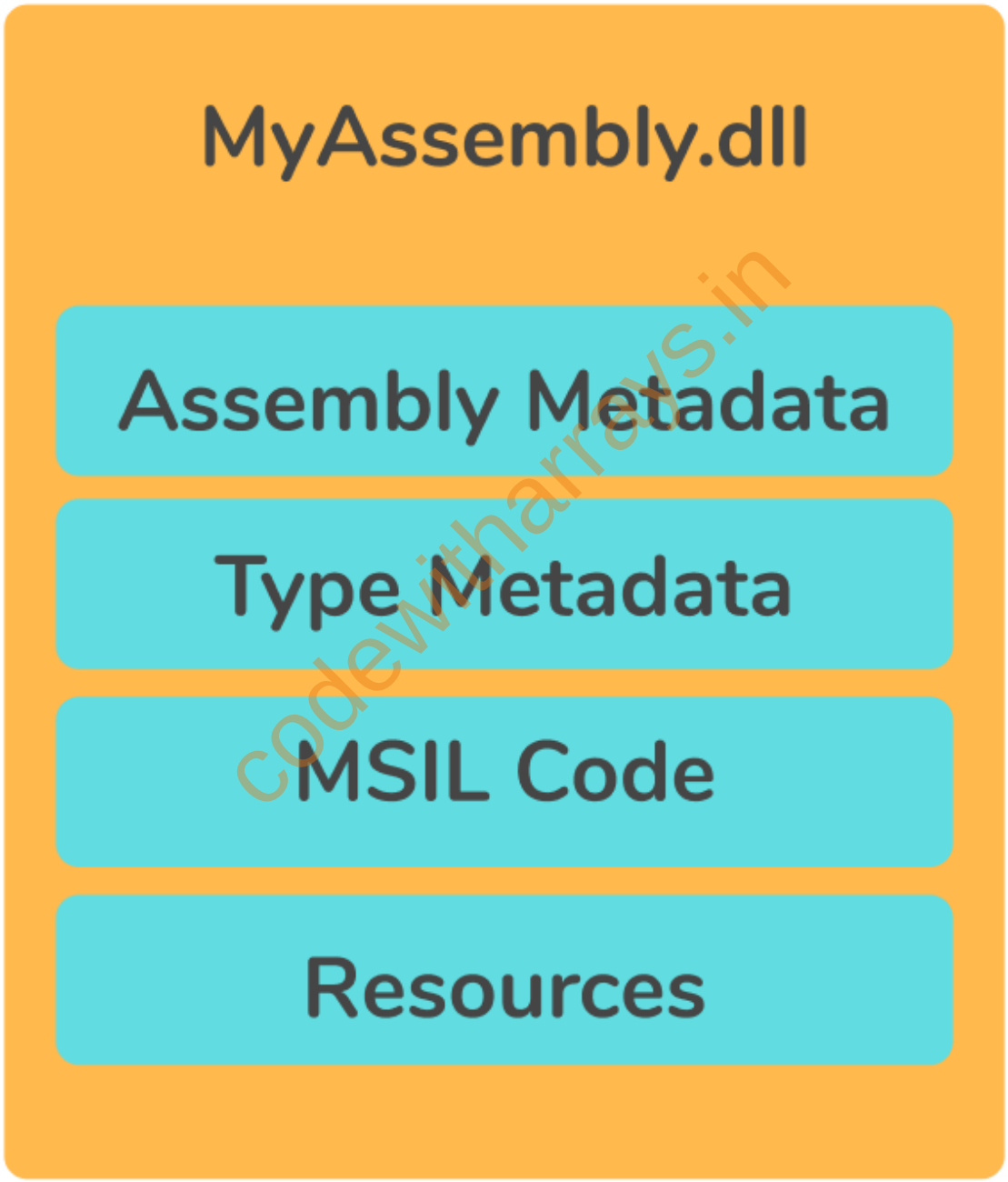
33. What are MDI and SDI?

MDI (Multiple Document Interface): An MDI allows you to open multiple windows, it will have one parent window and as many child windows. The components are shared from the parent window like toolbar, menubar, etc.

SDI (Single Document Interface): SDI opens each document in a separate window. Each window has its own components like a toolbar, menubar, etc. Therefore it is not constrained to the parent window.

34. Explain the different parts of an Assembly.

The different parts of an assembly are:



The different parts of an assembly are:

- **Manifest** – Every static or dynamic assembly holds a data collection that gives details about how the elements in the assembly relate to each other. An assembly manifest consists of complete metadata required to specify version requirements and security identity of an assembly, and also the metadata required for defining the assembly scope and resolving references to classes and resources.
  The assembly manifest will be stored in either a standalone PE(Portable Executable) file that holds only assembly manifest information, or in a PE file (a .exe or .dll) with MSIL(Microsoft intermediate language) code.
- **Type Metadata** – Metadata gives you additional information such as types, type names, method names, etc about the contents of an assembly. Metadata will be automatically generated by the Compilers from the source files and the compiler will embed this metadata within target output files like .exe, .dll, or a .netmodule(in the case of multi-module assembly).
- **MSIL** – Microsoft Intermediate Language(MSIL) is a code that implements the types. It includes instructions to load, store, initialize, and call the methods on objects. Along with this, it also includes instructions for control flow, direct memory access, arithmetic and logical operations, exception handling, etc. This is generated by the compiler using one or more source code files. During the runtime, the JIT(Just In Time) compiler of CLR(Common Language Runtime) converts the MSIL code into native code to the Operating System.
- **Resources** – Resources can be a list of related files such as .bmp or .jpg files. These resources are static, which means they do not change during run time. Resources are not executable items.

.NET Core Interview Questions

35. What is .NET core?

.NET Core can be said as the newer version of the .NET Framework. It is a cost-free, general-purpose, open-source application development platform provided by Microsoft. It is a cross-platform framework because it runs on various operating systems such as Windows, Linux, and macOS. This Framework can be used to develop applications like mobile, web, IoT, machine learning, game, cloud, microservices, etc.

It consists of important features like a cross-platform, sharable library, etc., that are necessary for running a basic .NET Core application. The remaining features are supplied in the form of NuGet packages, that can be added to your application according to your needs. Like this we can say, the .NET Core will boost up the performance of an application, decreases the memory footprint, and becomes easier for maintenance of an application. It follows the modular approach, so instead of the entire .NET Framework installation, your application can install or use only what is required.

36. What is Dot NET Core used for?

- .NET Core is useful in the server application creations, that run on various operating systems like Windows, Mac, and Linux. Using this, developers can write libraries as well as applications in C#, F#, and VB.NET in both runtimes.
- Generally, it is used for cloud applications or for modifying large enterprise applications into microservices.

- .NET Core 3.0 supports cross-development between WPF, UWP, and Windows Forms.
- .NET Core supports microservices, which permits cross-platform services to work with the .NET Core framework including services developed with .NET Framework, Ruby, Java, etc.
- .NET Core's features like lightweight, modularity, and flexibility make it easier to deploy .NET Core applications in containers. These containers can be deployed on any platform, Linux, cloud, and Windows.

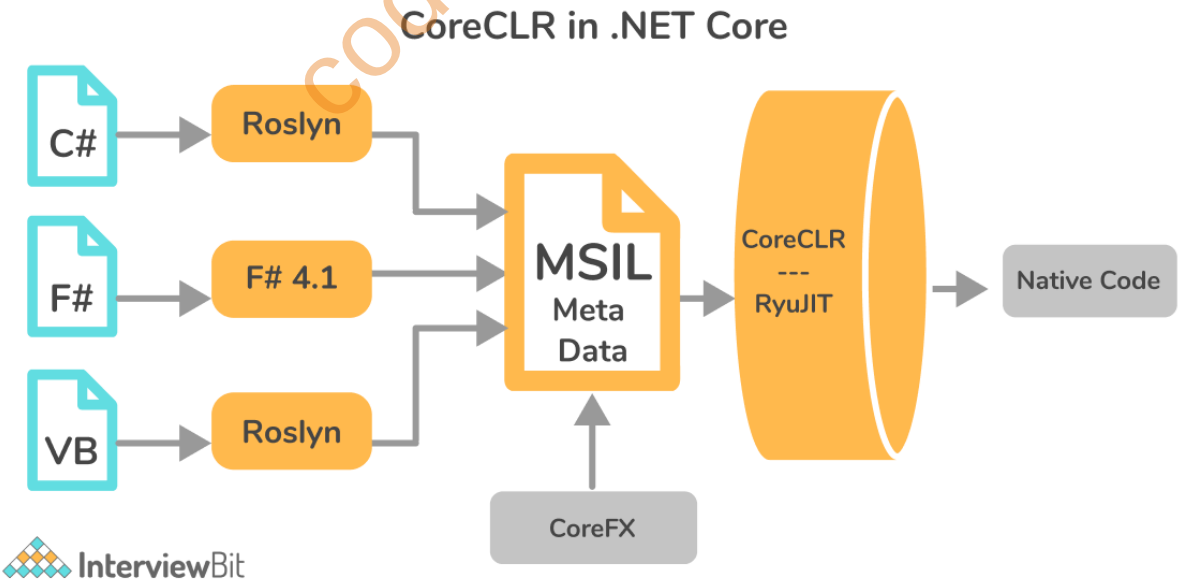
37. What are C# and F#?

C# is a general-purpose and object-oriented programming language from Microsoft that runs on the .NET platform. It is designed for CLI(Common Language Infrastructure), which has executable code and a runtime environment that allows for the usage of different high-level languages on various computer platforms and architectures. It is mainly used for developing web applications, desktop applications, mobile applications, database applications, games, etc.

F# is an open-source, functional-first, object-oriented and, cross-platform programming language that runs on a .NET platform and is used for writing robust, succinct, and performant code. We can say that F# is data-oriented because here code involves transforming data with functions. It is mainly used in making scientific models, artificial intelligence research work, mathematical problem solving, financial modelling, GUI games, CPU design, compiler programming, etc.

38. What is CoreCLR?

CoreCLR is the run-time execution engine provided by the .NET Core. It consists of a JIT compiler, garbage collector, low-level classes, and primitive data types. .NET Core is a modular implementation of .NET, and can be used as the base stack for large scenario types, ranging from console utilities to web applications in the cloud.



Here, various programming languages will be compiled by respective compilers(Roslyn can compile both C# and VB code as it includes C# and VB compilers) and Common

Intermediate Language(CIL) code will be generated. When the application execution begins, this CIL code is compiled into native machine code by using a JIT compiler included within CoreCLR. This CoreCLR is supported by many operating systems such as Windows, Linux, etc.

39. What is the purpose of webHostBuilder()?

WebHostBuilder function is used for HTTP pipeline creation through `webHostBuilder.Use()` chaining all at once with `WebHostBuilder.Build()` by using the builder pattern. This function is provided by `Microsoft.AspNet.Hosting namespace`. The Build() method's purpose is building necessary services and a `Microsoft.AspNetCore.Hosting.IWebHost` for hosting a web application.

40. What is Zero Garbage Collectors?

Zero Garbage Collectors allows you for object allocation as this is required by the Execution Engine. Created objects will not get deleted automatically and theoretically, no longer required memory is never reclaimed.

There are two main uses of Zero Garbage Collectors. They are:

- Using this, you can develop your own Garbage Collection mechanism. It provides the necessary functionalities for properly doing the runtime work.
- It can be used in special use cases like very short living applications or almost no memory allocation(concepts such as No-alloc or Zero-alloc programming). In these cases, Garbage Collection overhead is not required and it is better to get rid of it.

41. What is CoreFx?

CoreFX is the set of class library implementations for .NET Core. It includes collection types, console, file systems, XML, JSON, async, etc. It is platform-neutral code, which means it can be shared across all platforms. Platform-neutral code is implemented in the form of a single portable assembly that can be used on all platforms.

42. What is the IGCToCLR interface?

IGCToCLR interface will be passed as an argument to the InitializeGarbageCollector() function and it is used for runtime communication. It consists of a lot of built-in methods such as RestartEE(), SuspendEE(), etc.

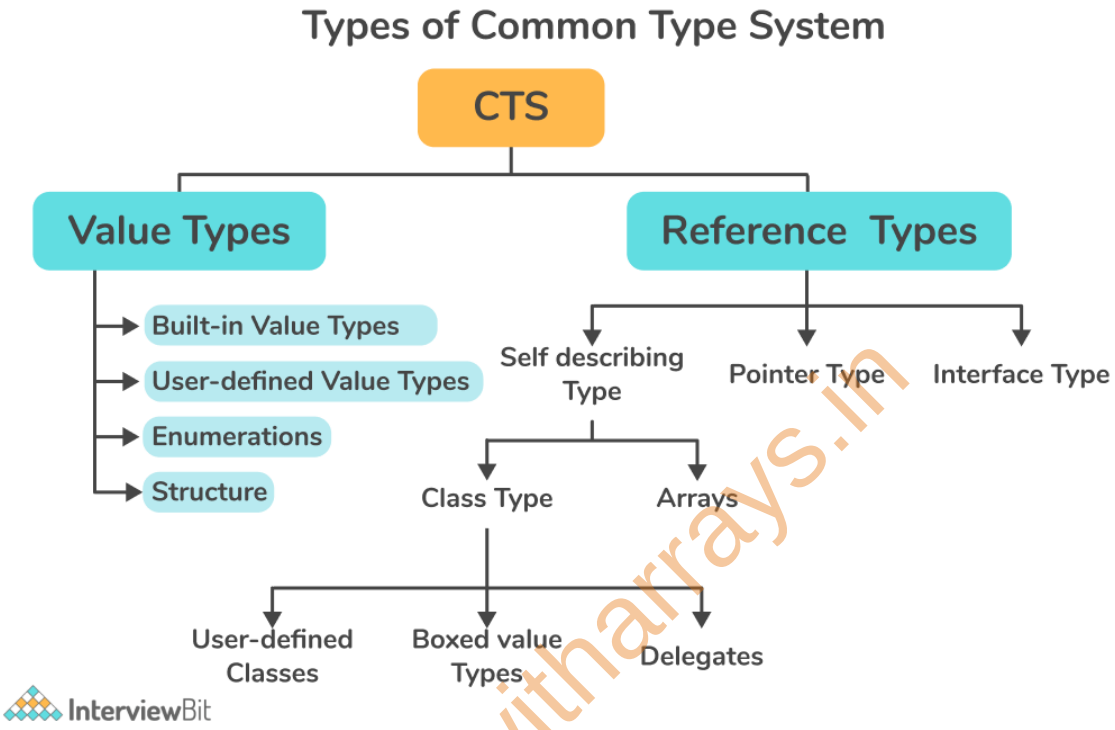
43. What is the use of generating SQL scripts in the .NET core?

It's useful to generate a SQL script, whenever you are trying to debug or deploy your migrations to a production database. The SQL script can be used in the future for reviewing the accuracy of data and tuned to fit the production database requirement.

44. Explain about types of Common Type System(CTS).

Common Type System(CTS) standardizes all the datatypes that can be used by different programming languages under the .NET framework.

CTS has two types. They are:



1. **Value Types:** They contain the values that are stored on a stack or allocated inline within a structure. They are divided into :
    - Built-in Value Types - It includes primitive data types such as Boolean, Byte, Char, Int32, etc.
    - User-defined Value Types - These are defined by the user in the source code. It can be enumeration or structure.
    - Enumerations - It is a set of enumerated values stored in the form of numeric type and are represented by labels.
    - Structures - It defines both data(fields of the structure) and the methods(operations performed on that data) of the structure. In .NET, all primitive data types like Boolean, Byte, Char, DateTime, Decimal, etc., are defined as structures.
2. **Reference Types:** It Stores a reference to the memory address of a value and is stored on the heap. They are divided into :
    - Interface types - It is used to implement functionalities such as testing for equality, comparing and sorting, etc.
    - Pointer types - It is a variable that holds the address of another variable.
    - Self-describing types - It is a data type that gives information about themselves for the sake of garbage collectors. It includes arrays(collection of variables with the same datatype stored under a single name) and class types(they define the operations like methods, properties, or events that are performed by an object and the data that the object contains) like user-defined classes, boxed value types, and delegates(used for event handlers and callback functions).

45. Give the differences between .NET Core and Mono?

.NET Core	Mono
.Net Core is the subset of implementation for the .NET framework by Microsoft itself.	Mono is the complete implementation of the .Net Framework for Linux, Android, and iOS by Xamarin.
.NET Core only permits you to build web applications and console applications.	Mono permits you to build different application types available in .NET Framework, including mobile applications, GUI-enabled desktop apps, etc.
.NET Core does not have the built-in capability to be compiled into WebAssembly-compatible packages.	Mono has the built-in capability to be compiled into WebAssembly-compatible packages.
.NET Core is never intended for gaming. You can only develop a text-based adventure or relatively basic browser-based game using .NET Core.	Mono is intended for the development of Games. Games can be developed using the Unity gaming engine that supports Mono.

46. What is Transfer-encoding?

Transfer-encoding is used for transferring the payload body(information part of the data sent in the HTTP message body) to the user. It is a hop-by-hop header, that is applied not to a resource itself, but to a message between two nodes. Each multi-node connection segment can make use of various Transfer-encoding values.

Transfer-encoding is set to “Chunked” specifying that Hypertext Transfer Protocol’s mechanism of Chunked encoding data transfer is initiated in which data will be sent in a form of a series of “chunks”. This is helpful when the amount of data sent to the client is larger and the total size of the response will not be known until the completion of request processing.

47. Whether ‘debug’ and ‘trace’ are the same?

No. The Trace class is used for debugging as well as for certain build releases. It gives execution plan and process timing details. While debug is used mainly for debugging. Debug means going through the program code flow during execution time.

Debug and trace allow for monitoring of the application for errors and exceptions without VS.NET IDE.

48. What is MSBuild in the .NET Core?

MSBuild is the free and open-source development platform for Visual Studio and Microsoft. It is a build tool that is helpful in automating the software product creation process, along with source code compilation, packaging, testing, deployment, and documentation creation. Using MSBuild, we can build Visual Studio projects and solutions without the need of installing the Visual Studio IDE.

In the Universal Windows Platform(UWP) app, if you open the folder named project, you will get to see both files namely `project.json` and `*.csproj`. But if you open our previous Console application in .NET Core, you will get to see `project.json` and `*.xproj` files.

49. What are Universal Windows Platform(UWP) Apps in .Net Core?

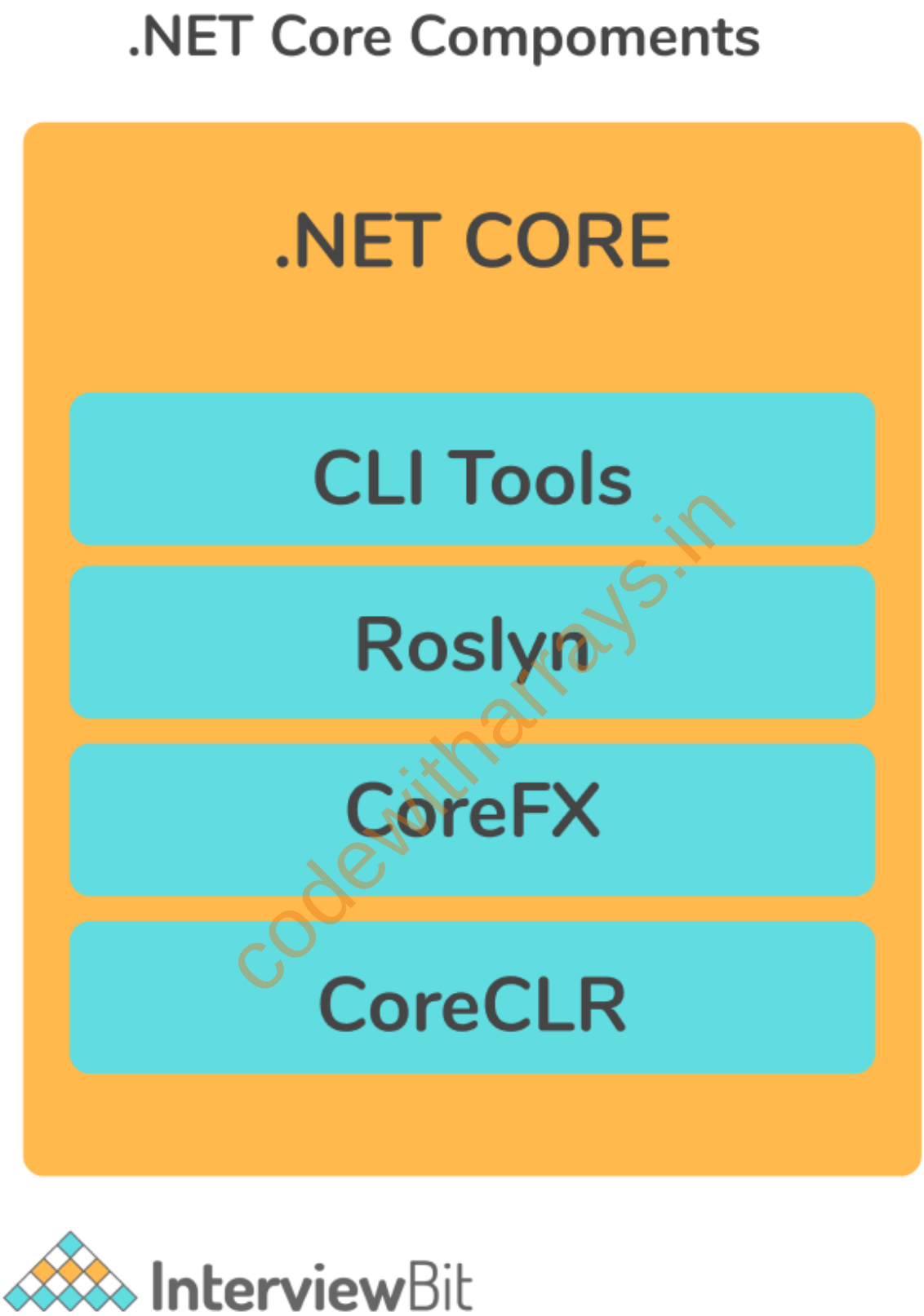
Universal Windows Platform(UWP) is one of the methods used to create client applications for Windows. UWP apps will make use of WinRT APIs for providing powerful UI as well as features of advanced asynchronous that are ideal for devices with internet connections.

Features of UWP apps:

- Secure: UWP apps will specify which resources of device and data are accessed by them.
- It is possible to use a common API on all devices(that run on Windows 10).
- It enables us to use the specific capabilities of the device and adapt the user interface(UI) to different device screen sizes, DPI(Dots Per Inches), and resolutions.
- It is available on the Microsoft Store on all or specified devices that run on Windows 10.
- We can install and uninstall these apps without any risk to the machine/incurring “machine rot”.
- Engaging: It uses live tiles, user activities, and push notifications, that interact with the Timeline of Windows as well as with Cortana’s Pick Up Where I Left Off, for engaging users.
- It can be programmable in C++, C#, Javascript, and Visual Basic. For UI, you can make use of WinUI, HTML, XAML, or DirectX.

50. Explain about .NET Core Components.

The .NET Core Framework is composed of the following components:



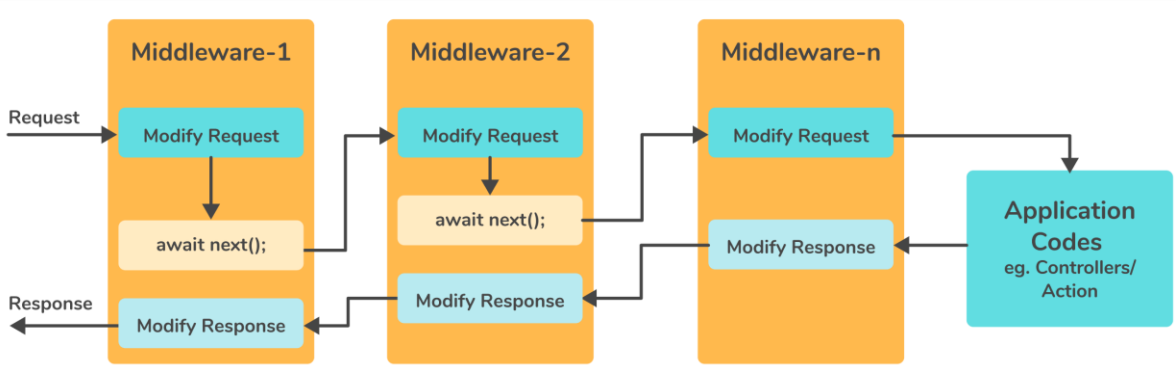
- **CLI Tools:** Command Line Interface(CLI) tools is a cross-platform tool for developing, building, executing, restoring packages, and publishing. It is also capable of building Console applications and class libraries that can run on the entire .NET framework. It is installed along with .NET Core SDK for the selected platforms. So it does not require separate

installation on the development machine. We can verify for the proper CLI installation by typing `dotnet` on the command prompt of Windows and then pressing Enter. If usage and help-related texts are displayed, then we can conclude that CLI is installed properly.

- **Roslyn(.NET Compiler platform):** It is a set of an open-source language compiler and also has code analysis API for the C# and Visual Basic (VB.NET) programming languages. Roslyn exposes modules for dynamic compilation to Common Intermediate Language(CLI), syntactic (lexical) and semantic code analysis, and also code emission.
- **CoreFX:** CoreFX is a set of framework libraries. It consists of the new BCL(Base Class Library) i.e. `System.*` things like `System.Xml`, `System.Collections`, etc.
- **CoreCLR:** A JIT(Just In Time) based CLR (Command Language Runtime). CoreCLR is the runtime implementation that runs on cross-platform and has the GC, RyuJIT, native interop, etc.

51. What is middleware in .NET core?

- Middleware is software assembled into an application pipeline for request and response handling. Each component will choose whether the request should be passed to the next component within the pipeline, also it can carry out work before and after the next component within the pipeline.
- For example, we can have a middleware component for user authentication, another middleware for handling errors, and one more middleware for serving static files like JavaScript files, images, CSS files, etc.
- It can be built-in into the .NET Core framework, which can be added through NuGet packages. These middleware components are built as part of the configure method's application startup class. In the ASP.NET Core application, these Configure methods will set up a request processing pipeline. It contains a sequence of request delegates that are called one after another.
- Normally, each middleware will handle the incoming requests and passes the response to the next middleware for processing. A middleware component can take the decision of not calling the next middleware in the pipeline. This process is known as short-circuiting the pipeline or terminating the request pipeline. This process is very helpful as it avoids unnecessary work. For example, if the request is made for a static file such as a CSS file, image, or JavaScript file, etc., these static files middleware can process and serve the request and thus short-circuit the remaining pipeline.



Processing the request through middleware

InterviewBit

Here, there are three middlewares are associated with an ASP.NET Core web application. They can be either middleware provided by the framework, added through NuGet, or your

own custom middleware. The HTTP request will be added or modified by each middleware and control will be optionally passed to the next middleware and a final response will be generated on the execution of all middleware components.

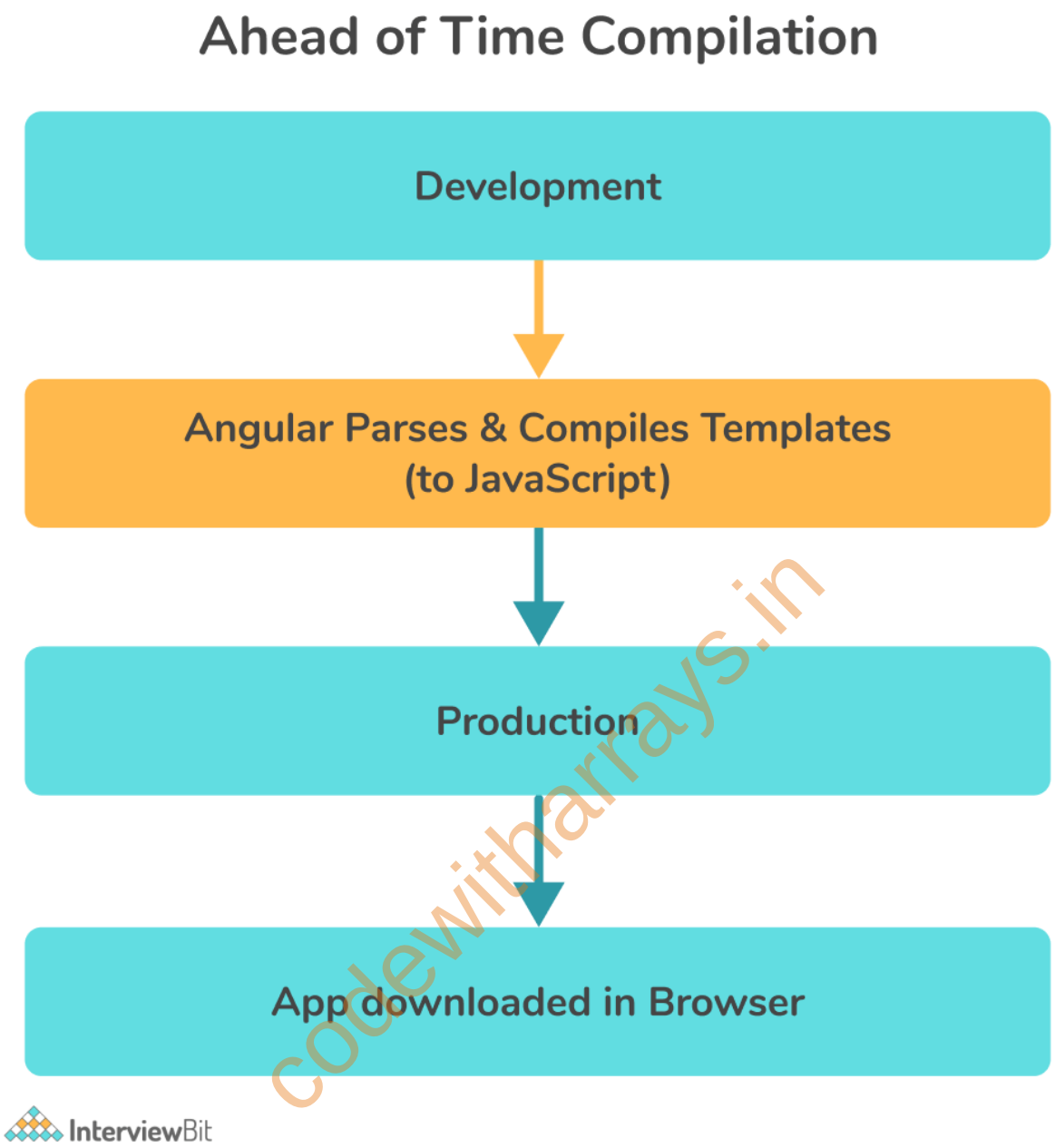
52. Differentiate .NET Core vs .NET framework.

Features	.NET Core	.NET framework
Compatibility	It works based on the principle of "build once, run anywhere". It is cross-platform, so it is compatible with different operating systems such as Linux, Windows, and Mac OS.	This framework is compatible with the Windows operating system only. Even though, it was developed for supporting software and applications on all operating systems.
Installation	Since it is cross-platform, it is packaged and installed independently of the OS.	It is installed in the form of a single package for Windows OS.
Application Models	It does not support developing the desktop application and it focuses mainly on the windows mobile, web, and windows store.	It is used for developing both desktop and web applications, along with that it also supports windows forms and WPF applications.
Performance and Scalability	It provides high performance and scalability.	It is less effective compared to .Net Core in terms of performance as well as scalability of applications.
Support for Micro-Services and REST Services	It supports developing and implementing the micro-services and the user is required to create a REST API for its implementation.	It does not support the microservices' development and implementation, but it supports REST API services.
Packaging and Shipping	It is shipped as a collection of Nugget packages.	All the libraries that belong to the .Net Framework are packaged and shipped all at once.
Android Development	It is compatible with open-source mobile app platforms like Xamarin, via .NET Standard Library. Developers can make use of tools of Xamarin for configuring the mobile application for particular mobile devices like Android, iOS, and Windows phones.	It does not support the development of mobile applications.
CLI Tools	For all supported platforms, it provides lightweight editors along with command-line tools.	This framework is heavy for CLI(Command Line Interface) and developers usually prefer to work on the lightweight CLI.

Features	.NET Core	.NET framework
Deployment Model	Updated version of the .NET Core gets initiated on one machine at a time, which means it gets updated in new folders/directories in the existing application without affecting it. Thus, we can say that .NET Core has a very good flexible deployment model.	When the updated version is released, it is deployed only on the Internet Information Server at first.

53. Explain Explicit Compilation (Ahead Of Time compilation).

- Ahead-of-time(AOT) compilation is the process of compiling a high-level language into a low-level language during build-time, i.e., before program execution. AOT compilation reduces the workload during run time.
- AOT provides faster start-up time, in larger applications where most of the code executes on startup. But it needs more amount of disk space and memory or virtual address space to hold both IL(Intermediate Language) and precompiled images. In this case, the JIT(Just In Time) Compiler will do a lot of work like disk I/O actions, which are expensive.
- The explicit compilation will convert the upper-level language into object code on the execution of the program. Ahead of time(AOT) compilers are designed for ensuring whether the CPU will understand line-by-line code before doing any interaction with it.
- Ahead-of-Time (AOT) compilation happens only once during build time and it does not require shipping the HTML templates and the Angular compiler into the bundle. The source code generated can begin running immediately after it has been downloaded into the browser, earlier steps are not required. The AOT compilation will turn the HTML template into the runnable code fragment. AOT will analyze and compile our templates statically during build time.



Benefits of AOT Compilation:

- Application size is smaller because the Compiler itself isn't shipped and unused features can be removed.
- Template the parse errors that are detected previously(during build time)
- Security is high (not required to dynamically evaluate templates)
- Rendering of a component is faster (pre-compiled templates)
- For AOT compilation, some tools are required to accomplish it automatically in the build process.

54. What is MEF?

The MEF(Managed Extensibility Framework) is a library that is useful for developing extensible and lightweight applications. It permits application developers for using extensions

without the need for configuration. It also allows extension developers for easier code encapsulation and thus avoiding fragile hard dependencies. MEF will let you reuse the extensions within applications, as well as across the applications. It is an integral part of the .NET Framework 4. It improves the maintainability, flexibility, and testability of large applications.

55. In what situations .NET Core and .NET Standard Class Library project types will be used?

.NET Core library is used if there is a requirement to increase the surface area of the .NET API which your library will access, and permit only applications of .NET Core to be compatible with your library if you are okay with it.

.NET Standard library will be used in case you need to increase the count of applications that are compatible with your library and reduce surface area(a piece of code that a user can interact with) of the .NET API which your library can access if you are okay with it.

56. What is CoreRT?

- CoreRT is the native runtime for the compilation of .NET natively ahead of time and it is a part of the new .NET Native (as announced in April 2014).
- It is not a virtual machine and it does not have the facility of generating and running the code on the fly as it doesn't include a JIT. It has the ability for RTTI(run-time type identification) and reflection, along with that it has GC(Garbage Collector).
- The type system of the CoreRT is designed in such a way that metadata for reflection is not at all required. This feature enables to have an AOT toolchain that can link away unused metadata and can identify unused application code.

57. What is .NET Core SDK?

.NET Core SDK is a set of tools and libraries that allows the developer to create a .NET application and library for .NET 5 (also .NET Core) and later versions. It includes the .NET CLI for building applications, .NET libraries and runtime for the purpose of building and running apps, and the dotnet.exe(dotnet executable) that runs CLI commands and runs an application. Here's the link to download.

58. What is Docker?

- Docker is an open-source platform for the development of applications, and also for shipping and running them. It allows for separating the application from the infrastructure using containers so that software can be delivered quickly. With Docker, you will be able to manage the infrastructure in the same ways you used to manage your applications.
- It supports shipping, testing, and deploying application code quickly, thus reducing the delay between code writing and running it in production.
- The Docker platform provides the ability of packaging and application execution in a loosely isolated environment namely container. The isolation and security permit you for running multiple containers at the same time on a given host. Containers are lightweight and they include every necessary thing required for running an application, so you need not depend on what is currently installed within the host.

59. What is Xamarin?

- Xamarin is an open-source platform useful in developing a modern and efficient application for iOS, Android, and Windows with .NET. It is an abstraction layer used to manage the communication of shared code with fundamental platform code.
- Xamarin runs in a managed environment that gives benefits like garbage collection and memory allocation.
- Developers can share about 90% of their applications over platforms using Xamarin. This pattern permits developers for writing entire business logic in a single language (or reusing existing app code) but accomplish native performance, look and feel on each platform. The Xamarin applications can be written on Mac or PC and then they will be compiled into native application packages, like a .ipa file on iOS, or .apk file on Android.

60. How can you differentiate ASP.NET Core from .NET Core?

.NET Core is a runtime and is used for the execution of an application that is built for it. Whereas ASP.NET Core is a collection of libraries that will form a framework for developing web applications. ASP.NET Core libraries can be used on .NET Core as well as on the "Full .NET Framework".

An application using the tools and libraries of ASP.NET Core is normally referred to as "ASP.NET Core Application", which in theory doesn't say whether it is built for .NET Framework or .NET Core. So an application of "ASP.NET Core" can be considered as a ".NET Core Application" or a ".NET Framework Application".

61. Write a program to calculate the addition of two numbers.

The steps are as follows:

1. You need to create a new ASP.NET Core Project "CalculateSum". Open Visual Studio 2015, goto **File–> New–> Project**. Select the option Web in Left Pane and go for the option **ASP.NET Core Web Application** (.NET Core) under the central pane. Edit the project name as "CalculateSum" and click on OK.

2. In the template window, select Web Application and set the Authentication into "No Authentication" and click on OK.

3. Open "Solution Explorer" and right-click on the folder "Home" (It is Under Views), then click on **Add New Item**. You need to select **MVC View Page Template** under ASP.NET Section and rename it as "addition.cshtml" and then click on the **Add** button.

4. Open addition.cshtml and write the following code:

```
 @{
    ViewBag.Title = "Addition Page";
}

<h1>Welcome to Addition Page</h1>

<form asp-controller="Home" asp-action="add" method="post">
```

```
    <span>Enter First Number : </span> <input id="Text1" type="text"
name="txtFirstNum" /> <br /><br />
    <span>Enter Second Number : </span> <input id="Text2" type="text"
name="txtSecondNum" /> <br /><br />
    <input id="Submit1" type="submit" value="Add" />
</form>

<h2>@ViewBag.Result</h2>
```

Here, we have created a simple form that is having two text boxes and a single Add Button. The text boxes are named as `txtFirstNum` and `txtSecondNum`. On the controller page, we can access these textboxes using:

```
<form asp-controller="Home" asp-action="add" method="post">
```

This form will indicate all the submissions will be moved to HomeController and the method add action will be executed.

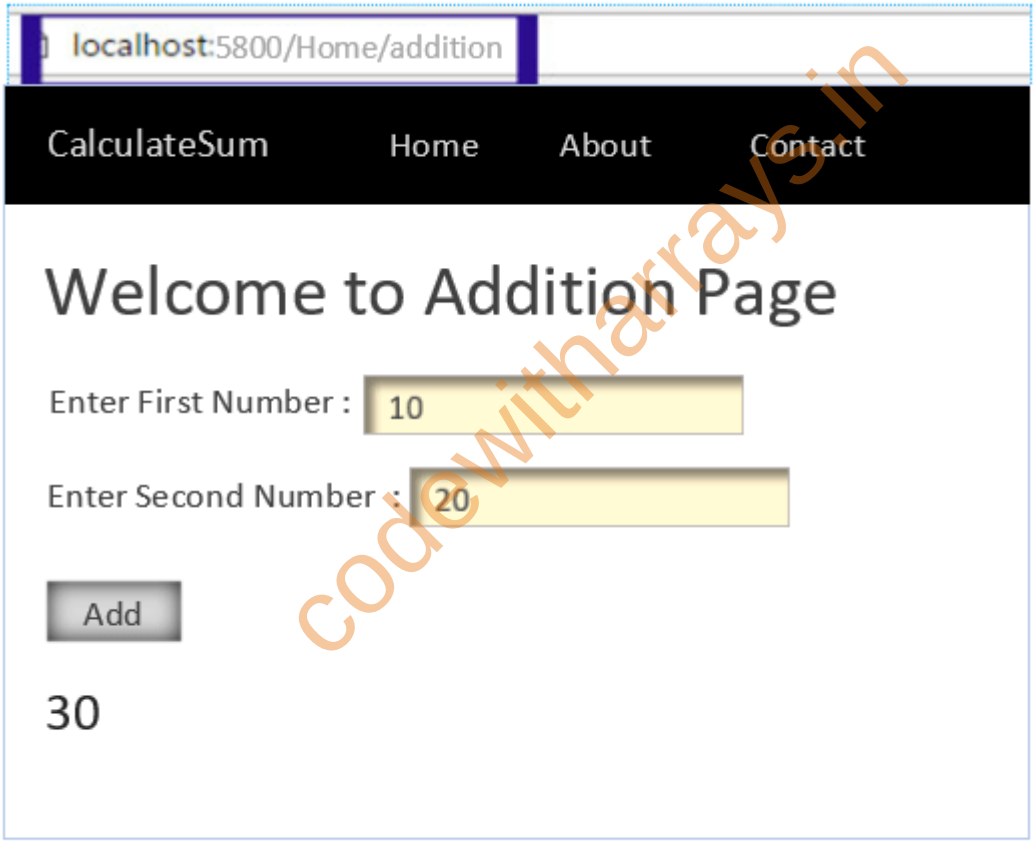
5. Open the `HomeController.cs` and write the following code onto it:

```
 using System;
using Microsoft.AspNetCore.Mvc;

namespace CalculateSum.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }

        public IActionResult About()
        {
            ViewData["Message"] = "Application description page.";
            return View();
        }

        public IActionResult Contact()
        {
            ViewData["Message"] = "Contact page.";
            return View();
        }

        public IActionResult Error()
        {
            return View();
        }

        public IActionResult addition()
        {
            return View();
        }

        [HttpPost]
        public IActionResult add()
        {
            int number1 =
Convert.ToInt32(HttpContext.Request.Form["txtFirstNum"].ToString());
```

```
            int number2 =
Convert.ToInt32(HttpContext.Request.Form["txtSecondNum"].ToString());
            int res = number1 + number2;
            ViewBag.Result = res.ToString();
            return View("addition");
        }
    }
}
```

In this program, we have added two IAction Methods addition() and add(). Addition() method will return the addition view page and add() method obtains input from the browser, processes it, and results will be kept in ViewBag.Result and then returned to the browser.

Now, press Ctrl+F5 for running your program. This will launch an ASP.NET Core website into the browser. Add `/Home/addition` at the end of the link and then hit on enter. The output format is given below:



Conclusion

The .NET is a full-stack software development framework, which is essentially used to build large enterprise-scale and scalable software applications. The .NET framework has wide scope in the market. It is a flexible and user-friendly framework, that goes well along with other technologies.

The .NET Core was developed in response to the surge in Java popularity. The .NET Core is normally used in low-risk projects. Some of the .NET components can be used in .NET core applications (but not the other way around). This article mainly concentrates on the framework concepts of .Net and .NET Core. We are sure that it would give you sufficient information and a fair knowledge of the common questions that will be asked during an interview.

.NET MCQ

1.

int keyword refers to which .NET type?

- System.Int32
- System.Int8
- System.Int16
- System.Int64

2.

Boxing allows the user to convert what?

- Integer type to double
- Value type to a reference type
- Reference type to a value type
- Double type to integer

3.

An Event has ____ as the default return type.

- No return type for events
- Integer
- String
- Double

4.

If the exception does not occur, will the final block get executed?

- There is no such block as finally
- Yes
- No
- Both catch and finally block will be executed.

5.

If a variable is declared inside a method, then it is called as _____

- Static
- Serial
- Local
- Private

6.

Runtime environment provided by .NET is called as _____

- RMT
- CLR
- RC
- RCT

7.

In which file .Net assembly's metadata is stored?

- manifest
- .dll
- .exe
- core

8.

Common Type System (CTS) can manage _____

- Value types
- All data types in .net
- Reference types
- Communication between multiple languages

9.

Specify root namespace used for fundamental types in .Net framework.

- System.Web
- System.IO
- System.Object
- System.File

10.

Which method is used to force garbage collection to run?

- GC.Run() method
- GC.Collection() method
- GC.Finalize() method
- GC.Collect() method

11.

What is the functionality of Response.Output.Write()?

- Response.Output.Write() allows you to buffer output
- Response.Output.Write() allows you to flush output
- Response.Output.Write() allows you to write formatted output
- Response.Output.Write() allows you to stream output

12.

__ is the latest version of .NET core as per 2021.

- .NET Core 3.1
- .NET Core 3.0
- .NET 6
- None of the above

13.

Docker is used for _____.

- Developing the applications
- Shipping the applications
- Running the applications
- All of the above

14.

.NET Core framework is composed of _____.

- CLI tools
- Roslyn
- CoreFX and CoreCLR
- All of the above

15.

What is the use of microservices?

- Mix technologies across the service boundary
- Debugging
- Creating and deploying a container
- None of the above

16.

.NET core is _____.

- Windows only
- Cross-platform
- Linux only
- MacOS only

17.

What is CoreFX?

- Toolset for the development of applications
- Compiler
- Foundational class libraries for .NET Core
- Docker container

18.

What is included within CoreCLR?

- Garbage Collector
- JIT Compiler
- Primitive data types
- All of the above

19.

What is Roslyn?

- .NET Compiler platform
- Container
- Library
- None of the above

20.

`System.Linq` namespace contains classes and interfaces that support _____.

- use of a memory-mapped file
- queries that use LINQ(Language-Integrated Query)
- to configure an assembly
- immutable collections

21.

Using which package we can share the libraries in .NET Core?

- NuGet package
- LINQ
- System
- Dapper

→2.ASP.NET INTERVIEW QUESTIONS:-

→REFERENCE → INTERVIEW BIT:-

ASP.NET is a web application framework developed by Microsoft which was released as part of the .NET framework. It makes it easy to build dynamic web applications. Some other examples of similar web application frameworks include Ruby on Rails (Ruby), Django (Python), and Express (JavaScript).

ASP.NET Core is a **cross-platform**, **high-performance**, and **open-source** web application framework. Microsoft released the first version of ASP.NET Core in 2016. It enables developers to create modern, cloud-enabled applications.

- **Cross-Platform:** The main advantage of ASP.NET Core is that it's not tied to a Windows operating system, like the legacy ASP.NET framework. You can develop and run production-ready ASP.NET Core apps on Linux or a Mac.
- **High Performance:** It's designed from scratch, keeping performance in mind. It's now one of the fastest web application frameworks.
- **Open Source:** Finally, it's open-source and actively contributed by thousands of developers all over the world.

Both the ASP.NET and ASP.NET Core run on C#, an object-oriented, general-purpose programming language. ASP.NET Core inherits many concepts and features from its ASP.NET heritage, but it's fundamentally a new framework.

Though Microsoft is going to support the legacy ASP.NET framework, it's not going to develop it actively. The new ASP.NET Core framework will include all the new features and enhancements. Going forward, Microsoft is recommending developers to build all the new web applications with ASP.NET Core instead of the legacy ASP.NET framework.

In this article, we will focus on both ASP.NET and ASP.NET Core interview questions. To limit the article's scope, we assume that you have programmed in the C# programming language. A basic understanding of common object-oriented concepts and front-end technologies such as HTML, CSS, and JavaScript is also expected.

We have divided the interview questions into two sections. The basic interview questions cover the fundamentals and focus on understanding the application structure of a basic ASP.NET project. Then, we cover the more advanced concepts such as dependency injection, routing, and model binding in the advanced interview questions.

→2.Basic ASP.NET Interview Questions:-

1. What is a web application?

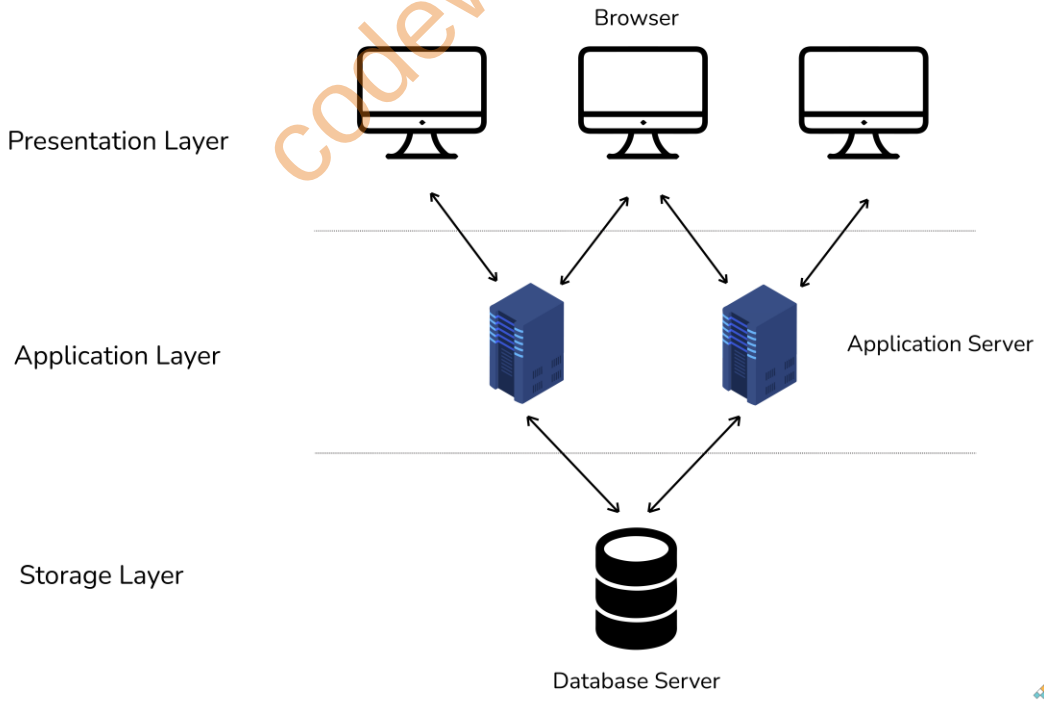
A Web application is a software that the users can access through a web browser such as Chrome or Firefox. The browser makes an HTTP request for a specific URL for the web application. The web application server intercepts and processes the request to build a dynamic HTML response sent to the user. Some examples of popular web applications include StackOverflow, Reddit, Google, etc.

A web application is different from a typical website. A website is static. When you go to the website, it returns an HTML page without doing any processing to build the contents of that HTML page. You will see the same page if you reload the browser. In contrast, a web application might return a different response each time you visit.

For example, let's say you ask a question on Stack Overflow. Initially, you will see only your question when you visit the URL. However, if another user answers your question, the browser will display that answer on your next visit to the same URL.

A web application consists of multiple separate layers. The typical example is a three-layered architecture made up of presentation, business, and data layers. For example, the browser (presentation) talks to the application server, which communicates to the database server to fetch the requested data.

The following figure illustrates a typical Web application architecture with standard components grouped by different areas of concern.



2. What is a web application framework, and what are its benefits?

Learning to build a modern web application can be daunting. Most of the web applications have a standard set of functionality such as:

- Build a dynamic response that corresponds to an HTTP request.
- Allow users to log into the application and manage their data.
- Store the data in the database.
- Handle database connections and transactions.
- Route URLs to appropriate methods.
- Supporting sessions, cookies, and user authorization.
- Format output (e.g. HTML, JSON, XML), and improve security.

Frameworks help developers to write, maintain and scale applications. They provide tools and libraries that simplify the above recurring tasks, eliminating a lot of unnecessary complexity.

3. What are some benefits of ASP.NET Core over the classic ASP.NET?

- **Cross-Platform:** The main advantage of ASP.NET Core is that it's not tied to a Windows operating system, like the legacy ASP.NET framework. You can develop and run production-ready ASP.NET Core apps on Linux or a Mac. Choosing an open-source operating system like Linux results in significant cost-savings as you don't have to pay for Windows licenses.
- **High Performance:** It's also designed from scratch, keeping performance in mind. It's now one of the fastest web application frameworks.
- **Open Source:** Finally, it's open-source and actively contributed by thousands of developers all over the world. All the source code is hosted on GitHub for anyone to see, change and contribute back. It has resulted in significant goodwill and trust for Microsoft, notwithstanding the patches and bug-fixes and improvements added to the framework by contributors worldwide.
- **New Technologies:** With ASP.NET Core, you can develop applications using new technologies such as Razor Pages and Blazor, in addition to the traditional Model-View-Controller approach.

4. When do you choose classic ASP.NET over ASP.NET Core?

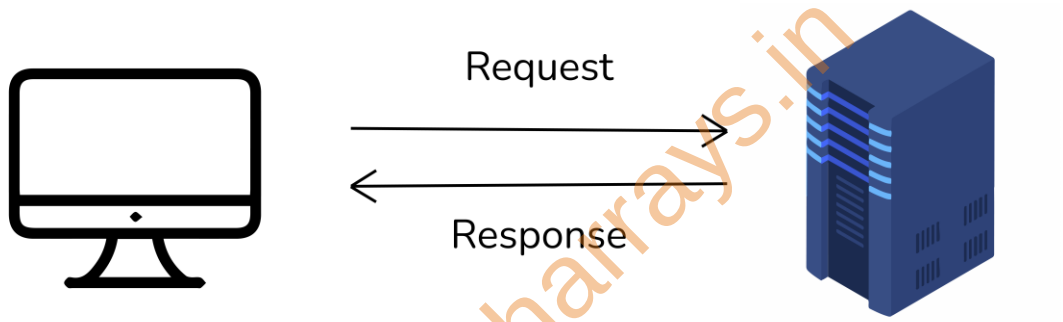
Though it's a better choice in almost all the aspects, you don't have to switch to ASP.NET Core if you are maintaining a legacy ASP.NET application that you are happy with and that is no longer actively developed.
ASP.NET MVC is a better choice if you:

- Don't need cross-platform support for your Web app.
- Need a stable environment to work in.
- Have nearer release schedules.
- Are already working on an existing app and extending its functionality.
- Already have an existing team with ASP.NET expertise.

5. Explain how HTTP protocol works?

Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It handles communication between web browsers and web servers. HTTP follows a classical client-server model. A client, such as a web browser, opens a connection to make a request, then waits until it receives a response from the server.

HTTP is a protocol that allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web, and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser.

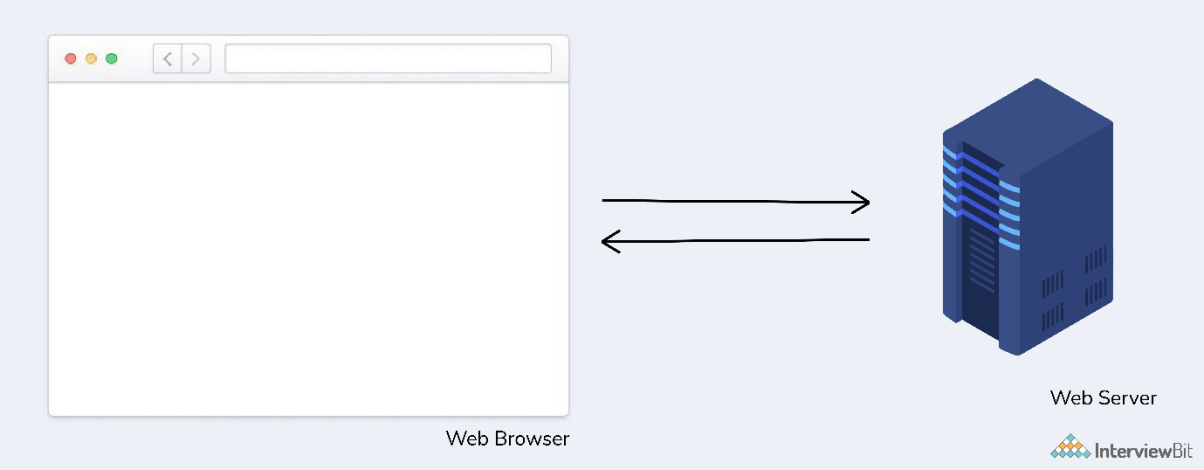


InterviewBit

6. What is a web server?

The term web server can refer to both hardware or software, working separately or together.

On the hardware side, a web server is a computer with more processing power and memory that stores the application's back-end code and static assets such as images and JavaScript, CSS, HTML files. This computer is connected to the internet and allows data flow between connected devices.

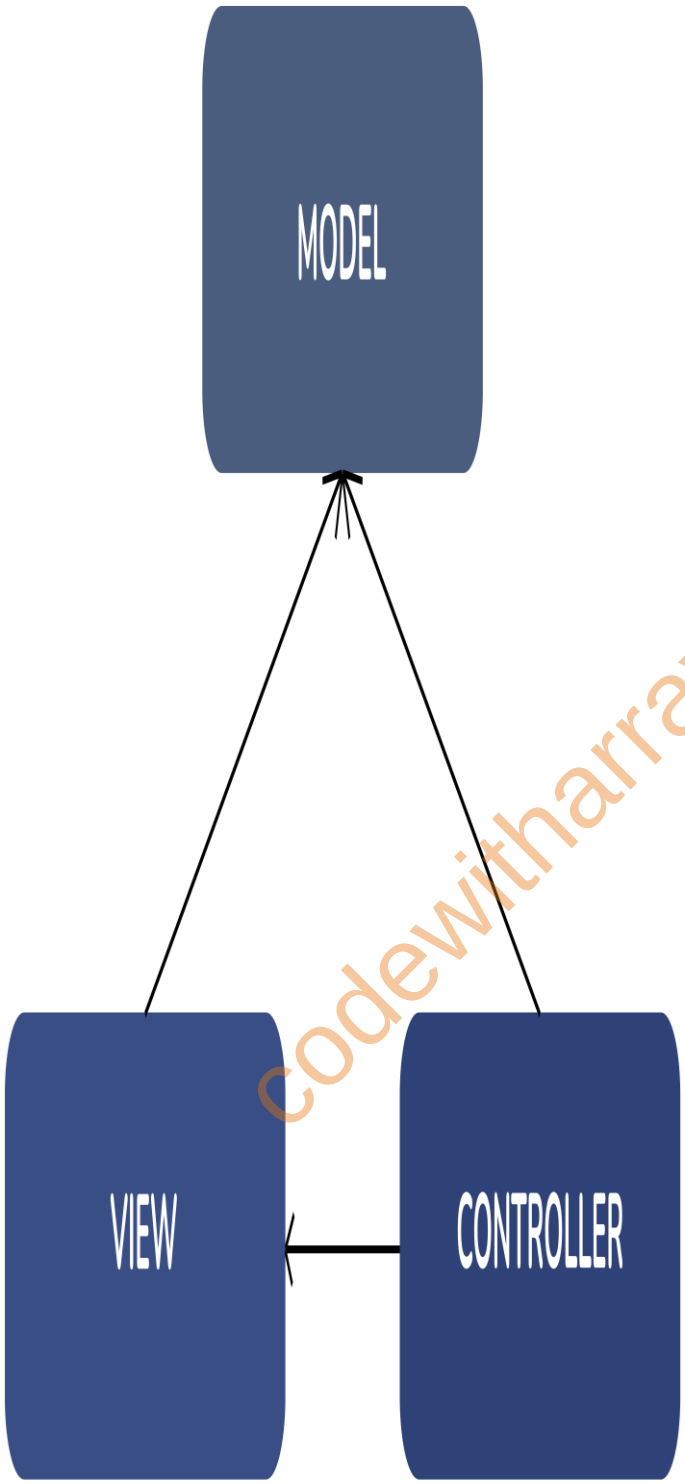


On the software side, a web server is a program that accepts HTTP requests from the clients, such as a web browser, processes the request, and returns a response. The response can be static, i.e. image/text, or dynamic, i.e. calculated total of the shopping cart.

Popular examples of web servers include Apache, Nginx, IIS.

7. What is the MVC pattern?

The **Model-View-Controller** (MVC) is an architectural pattern that separates an application into three main logical components: the **model**, the **view**, and the **controller**. It's different to note that this pattern is unrelated to the layered pattern we saw earlier. MVC pattern operates on the software side, while the layered pattern dictates how and where we place our database and application servers.



In an application that follows the MVC pattern, each component has its role well specified. For example, model classes only hold the data and the business logic. They don't deal with HTTP requests. Views only display information. The controllers handle and respond to user input and decide which model to pass to which view. This is known as the separation of responsibility. It makes an application easy to develop and maintain over time as it grows in complexity.

Though Model-View-Controller is one of the oldest and most prominent patterns, alternate patterns have emerged over the years. Some popular patterns include MVVM (Model-View-ViewModel), MVP (Model-View-Presenter), MVA (Model-View-Adapter).

8. Explain the role of the various components of the MVC pattern?

Model: Represents all the data and business logic that the user works within a web application. In ASP.NET, the model is represented by C# classes that hold the data and the related logic that operates on that data. The 'Models' directory stores the model classes.

For example, a model class representing a blog post might look like this:

```
// Models/Post.cs
namespace app.Models
{
  public class Post
  {
      public int ID { get; set; }

      public string Title { get; set; }

      public string Body { get; set; }
  }
}
```


View: Represents all the UI logic of the application. In a web application, it represents the HTML that's sent to the user and displayed in the browser.

One important thing to remember is that this HTML is not static or hard-coded. It's generated dynamically by the controller using a model's data. In ASP.NET, the 'Views' directory contains the views in files ending with the .cshtml file extension.

To continue our example of a blog post, a view to render a post might be:

```
// Views/Post.cshtml

<div class="post">
  <div class="title">
      <a href="/posts/@post.ID">@post.Title</a>
  </div>

  <div class=body>
      @Html.Raw(post.Body)
  </div>
</div>
```

Controller: Acts as an interface between Model and View. It processes the business logic and incoming requests, manipulates data using the Model, and interacts with the Views to render the final output.

In ASP.NET, these are C# classes that form the glue between a model and a view. They handle the HTTP request from the browser, then retrieve the model data and pass it to the view to dynamically render a response. The 'Controllers' directory stores the controller classes.

A PostController that builds the view for the post by fetching the Post model will be:

```
// Controllers/PostController
namespace app.Controllers
{
  public class PostsController : BaseController
  {
      public IActionResult Post(int id)
      {
          // Get the post from the database
          Post post = _service.Get(id);

          // Render the post.cshtml view, by providing the post model
          return View(post);
      }
  }
}
```

9. What is the purpose of the .csproj file?

The project file is one of the most important files in our application. It tells .NET how to build the project.

All .NET projects list their dependencies in the .csproj file. If you have worked with JavaScript before, think of it like a package.json file. The difference is, instead of a JSON, this is an XML file.

When you run dotnet restore, it uses the .csproj file to figure out which NuGet packages to download and copy to the project folder (check out the next question to learn more about Nuget).

The .csproj file also contains all the information that .NET tooling needs to build the project. It includes the type of project you are building (console, web, desktop, etc.), the platform this project targets, and any dependencies on other projects or 3rd party libraries.

Here is an example of a .csproj file that lists the Nuget packages and their specific versions.

```
<Project Sdk="Microsoft.NET.Sdk.Web">
<PropertyGroup>
  <TargetFramework>net5.0</TargetFramework>
</PropertyGroup>

<ItemGroup>
  <PackageReference Include="AWSSDK.S3" Version="3.5.6.5" />
  <PackageReference Include="Microsoft.EntityFrameworkCore.Sqlite"
Version="5.0.1" />
```

```
  <PackageReference Include="Microsoft.Extensions.Caching.Memory"
Version="5.0.0" />
  <PackageReference Include="Npgsql" Version="5.0.1.1" />
  <PackageReference Include="Serilog" Version="2.10.0" />
</ItemGroup>
</Project>
```

In the above example,

- The SDK attribute specifies the type of .NET project.
- TargetFramework is the framework this application will run on, .NET 5 in this case.
- The PackageReference element includes the NuGet packages. The Version attribute specifies a version of the package we want.

10. What is NuGet package manager?

Software developers don't write all code from scratch. They rely on libraries of code written by other developers. Any modern development platform must provide a mechanism where developers can download and use existing libraries, often called packages. For example, the JavaScript ecosystem has NPM (Node Package Manager), where developers can find and use libraries written by other JavaScript developers.

NuGet is a package manager for the .NET ecosystem. Microsoft developed it to provide access to thousands of packages written by .NET developers. You can also use it to share the code you wrote with others.

A typical web application developed using ASP.NET relies on many open source NuGet packages to function. For example, Newtonsoft.Json is a very popular package (with 91,528,205 downloads at the time of writing) used to work with JSON data in .NET.

11. What is the purpose of the Program class?

Program.cs class is the entry point of our application. An ASP.NET application starts in the same way as a console application, from a **static void Main()** function.

This class configures the web host that will serve the requests. The host is responsible for application startup and lifetime management, including graceful shutdown.

At a minimum, the host configures a server and a request processing pipeline. The host can also set up logging, configuration, and dependency injection.

12. What is the purpose of the Startup class?

This class handles two important aspects of your application, namely service registration, and middleware pipeline.

Services are C# classes that your application depends on for providing additional functionality, both used by the framework and your application. Examples include logging, database, etc. These services must be registered to be instantiated when your app is running and when it needs them.

The middleware pipeline is the sequence in which your application processes an HTTP request (the next question explains the concept of Middleware in detail).

Startup class contains two methods: ***ConfigureServices()*** and ***Configure()***. As the name suggests, the first method registers all the services that the application needs. The second method configures the middleware pipeline.

13. What is the purpose of the wwwroot folder?

The wwwroot folder contains static files and compiled assets, such as JavaScript, CSS, and images that your web application needs. Wwwroot is the only folder in the entire project that's exposed as-is to the browser.

14. What is the purpose of the appsettings.json file?

Appsettings.json contains all of the application's settings, which allow you to configure your application behavior.

Here is an example of an appsettings.json file.

```
{
 "Logging": {
     "LogLevel": {
          "Default": "Information",
    "Microsoft": "Warning",
    "Microsoft.Hosting.Lifetime": "Information"
      }
  },
"ConnectionStrings": {
    "AppConnection": ""
},
"AWS": {
  "Profile": "local-test-profile",
  "Region": "us-west-2"
},
"AllowedHosts": "*"
}
```


15. What is IIS?

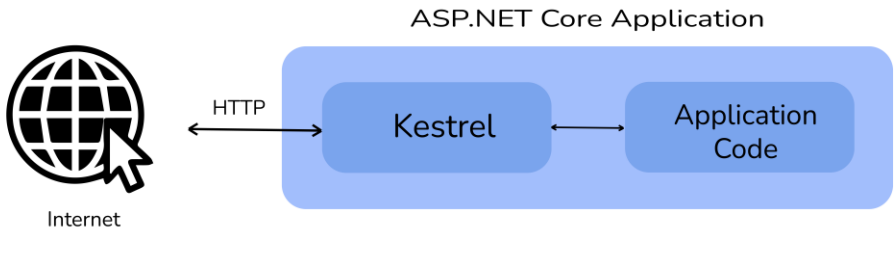
IIS stands for Internet Information Services. It is a powerful web server developed by Microsoft. IIS can also act as a load balancer to distribute incoming HTTP requests to different application servers to allow high reliability and scalability.

It can also act as a reverse proxy, i.e. accept a client's request, forward it to an application server, and return the client's response. A reverse proxy improves the security, reliability, and performance of your application.

A limitation of IIS is that it only runs on Windows. However, it is very configurable. You can configure it to suit your application's specific needs.

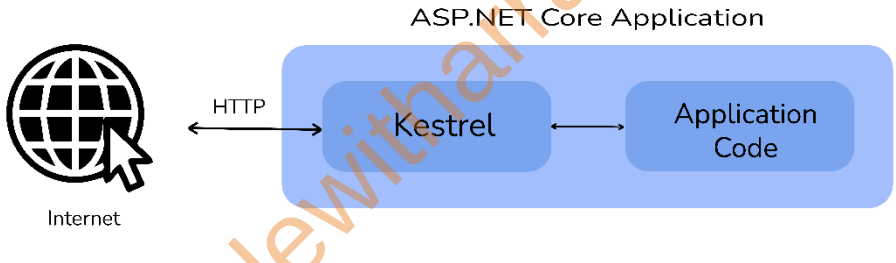
16. What is Kestrel?

Kestrel is an open-source, cross-platform web server designed for ASP.NET Core. Kestrel is included and enabled by default in ASP.NET Core. It is very light-weight when compared with IIS.



Kestrel can be used as a web server processing requests directly from a network, including the Internet.

Though Kestrel can serve an ASP.NET Core application on its own, Microsoft recommends using it along with a reverse proxy such as IIS, Nginx, or Apache, for better performance, security, and reliability.



17. What is the difference between IIS and Kestrel? Why do we need two web servers?

The main difference between IIS and Kestrel is that Kestrel is a cross-platform server. It runs on Windows, Linux, and Mac, whereas IIS only runs on Windows.

Another essential difference between the two is that Kestrel is fully open-source, whereas IIS is closed-source and developed and maintained only by Microsoft.

IIS is very old software and comes with a considerable legacy and bloat. With Kestrel, Microsoft started with high-performance in mind. They developed it from scratch, which allowed them to ignore the legacy/compatibility issues and focus on speed and efficiency.

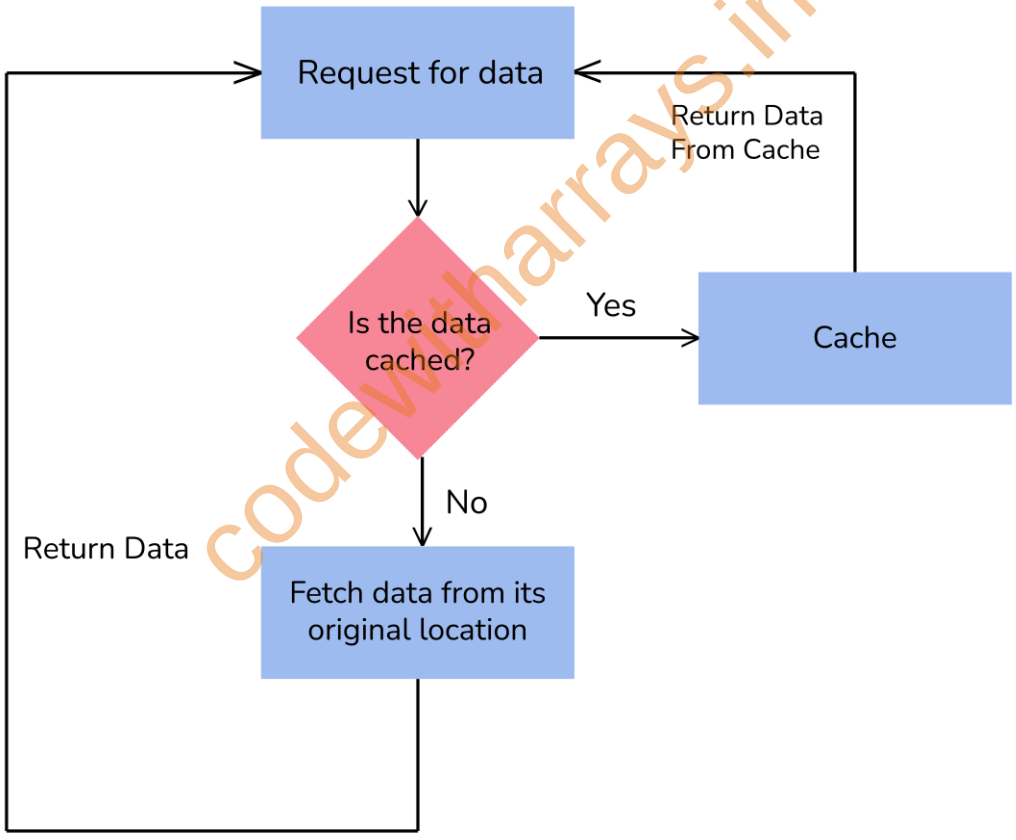
However, Kestrel doesn't provide all the rich functionality of a full-fledged web server such as IIS, Nginx, or Apache. Hence, we typically use it as an application server, with one of the above servers acting as a reverse proxy.

18. What is caching?

Caching is the process of storing data in a temporary storage location that is quicker to access than the original location of the data so that it can be accessed more quickly when the same data is needed next time.

Caching improves the scalability and performance of your application. It does this by reducing the work required to fetch the data. Caching is useful for data that doesn't change frequently and is expensive to create and retrieve.

ASP.NET provides a set of caching features out of the box. You can use the IMemoryCache interface for simple use cases. It represents a cache stored in the web server's memory. ASP.NET also supports distributed caching, which is a cache shared by multiple app servers, with Redis.



InterviewBit

Advanced ASP.NET Interview Questions

19. What is model binding in ASP.NET?

Controllers and views need to work with data that comes from HTTP requests. For example, routes may provide a key that identifies a record, and posted form fields may provide model

properties. The process of converting these string values to .NET objects could be complicated and something that you have to do with each request. Model binding automates and simplifies this process.

The model binding system fetches the data from multiple sources such as form fields, route data, and query strings. It also provides the data to controllers and views in method parameters and properties, converting plain string data to .NET objects and types in the process.

Example:
Let's say you have the following action method on the PostsController class:

```
[HttpGet("posts/{id}")]
public ActionResult<Post> GetById(int id, bool archivedOnly)
```

And the app receives a request with this URL:

```
http://yourapp.com/api/Posts/5?ArchivedOnly=true
```

After the routing selects the action method, model binding executes the following steps.

- Locate the first parameter of GetByID, an integer named id, look through the available sources in the HTTP request and find id = "5" in route data.
- Convert the string "5" into an integer 5.
- Find the next parameter of GetByID, a boolean named archivedOnly.
- Look through the sources and find "ArchivedOnly=true" in the query string. It ignores the case when matching the parameters to the strings.
- Convert the string "true" into boolean true.

Some other examples of attributes include:
1. [FromQuery] - Gets values from the query string.
2. [FromRoute] - Gets values from route data.
3. [FromForm] - Gets values from posted form fields.
4. [FromBody] - Gets values from the request body.
5. [FromHeader] - Gets values from HTTP headers.

20. What is an Action Method?

An action method is a method in a controller class with the following restrictions:

1. It must be public. Private or protected methods are not allowed.
2. It cannot be overloaded.
3. It cannot be a static method.

An action method executes an action in response to an HTTP request.

For example, here is an example of an Index() action method on the PostController. It takes an id as an input and returns an IActionResult, which can be implemented by any result classes (see the following question).

```
public class PostController : Controller
```

```
{
  public IActionResult Index(int id)
  {

  }
}
```

21. What are the different types that implement the IActionResult interface?

ASP.NET Core has many different types of IActionResult:

- ViewResult—Generates an HTML view.
- RedirectResult—Sends a 302 HTTP redirect response to send a user to a specified URL automatically.
- RedirectToRouteResult—Sends a 302 HTTP redirect response to automatically send a user to another page, where the URL is defined using routing.
- FileResult—Returns a file as the response.
- ContentResult—Returns a provided string as the response.
- StatusCodeResult—Sends a raw HTTP status code as the response, optionally with associated response body content.
- NotFoundResult—Sends a raw 404 HTTP status code as the response.

22. What's the HTTPContext object? How can you access it within a Controller?

HTTPContext encapsulates all HTTP-specific information about an individual HTTP request. You can access this object in controllers by using the **ControllerBase.HttpContext** property:

```
public class HomeController : Controller
{
  public IActionResult About()
  {
      var pathBase = HttpContext.Request.PathBase;

      ...

      return View();
  }
}
```


23. What is dependency injection?

Dependency injection is a design pattern that helps to develop loosely coupled code. This pattern is used extensively in ASP.NET.

Dependency injection means **providing** the objects that an object needs (its dependencies) in that object's constructor instead of requiring the object to construct them.

Dependency injection reduces and often eliminates unnecessary dependencies between objects that don't need to know each other. It also helps in testing by mocking or stubbing out the dependencies at runtime.

24. Explain how dependency injection works in ASP.NET Core?

ASP.NET Core injects instances of dependency classes by using the built-in IoC (Inversion-of-Control) container. This container is represented by the IServiceProvider interface that supports constructor injection.

The types (classes) managed by the container are called services. To let the IoC container automatically inject our services, we first need to register them with the IoC container in the Startup class.

ASP.NET Core supports two types of services, namely framework and application services. **Framework** services are a part of ASP.NET Core framework such as ILoggerFactory, IApplicationBuilder, IHostingEnvironment, etc. In contrast, a developer creates the **application services** (custom types or classes) specifically for the application.

25. What is a cookie?

A cookie is a small amount of data that is persisted across requests and even sessions. Cookies store information about the user. The browser stores the cookies on the user's computer. Most browsers store the cookies as key-value pairs.

Write a cookie in ASP.NET Core:

```
Response.Cookies.Append(key, value);
```

Delete a cookie in ASP.NET Core

```
Response.Cookies.Delete(somekey);
```

26. Explain the concept of middleware in ASP.NET Core?

In general, middleware is plumbing software that facilitates communication flow between two components. In a web application framework, middleware provides common services and capabilities to the application outside of the default framework.

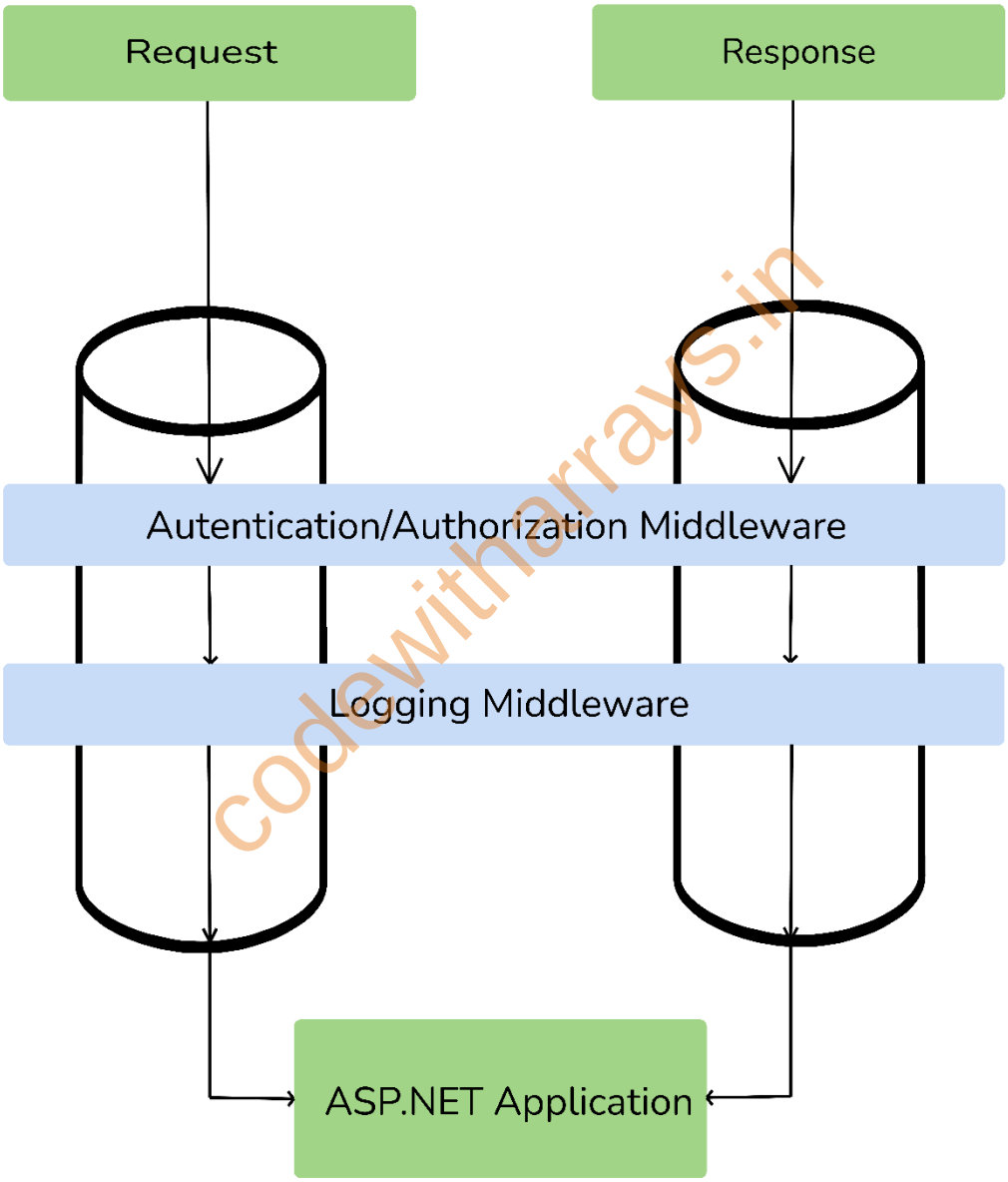
In ASP.NET Core, middleware refers to the C# classes that manipulate an HTTP request when it comes in or an HTTP response when it's on its way out. For example,

Generate an HTTP response for an incoming HTTP request
Intercept and make changes to an incoming HTTP request and pass it on to the next piece of middleware.
Intercept and make changes to an outgoing HTTP response, and pass it on to the next piece of middleware.

One of the most common use cases for middleware is to deal with concerns that affect your entire application. These aspects of your application need to occur with every request, regardless of the specific path in the request or the resource requested. These include things like logging, security, authentication, authorization, etc.

For example, logging middleware logs when a request comes in and passes it to another piece of middleware. Some other common middleware uses include database middleware, error handling middleware, and authentication/authorization middleware.

In each of these examples, the middleware receives a request, modifies it, and then passes it to the next middleware piece in the pipeline. Subsequent middleware uses the details added by the earlier middleware to handle the request in some way. The following diagram illustrates this.



27. What is routing, and how can you define routes in ASP.NET Core?

Routing is the process of mapping an incoming HTTP request to a specific method in the application code. A router maps the incoming requests to the route handler. It takes in a URL as an input and deconstructs it to determine the controller and action method to route the request.

A simple routing pattern, for example, might determine that the **/posts/show** URL maps to the **Show** action method on the **PostsController**.

There are two ways to define routes in an ASP.NET Core MVC application.

- Conventional Routing
- Attribute-Based Routing.

We can use both Conventional Routing and Attribute Routing in an application.

28. Explain how conventional routing works?

As the name suggests, conventional routing uses predefined conventions to match the incoming HTTP request to a controller's action method. It handles the most general cases that a typical web application needs, but you can modify it to suit your specific needs.

For example, the Configure() method in the Startup.cs class contains the following code that sets up the conventional routing.

```
app.UseEndpoints(endpoints =>
{
  endpoints.MapControllerRoute(
      name: "default",
      pattern: "{controller=Home}/{action=Index}/{id?}");
});
```


This code creates a single route named 'default'. The route template pattern '{controller=Home}/{action=Index}/{id?}' is used to match an incoming URL such as **/Posts/Archived/5** to the **Archived(int id)** action method on the PostsController, passing 5 for the id parameter. By default, the router uses the Index method on the HomeController.

29. Explain how attribute-based routing works?

Attribute routing is an alternative routing strategy for conventional routing. It is often used to create REST API endpoints for web services. It uses a set of attributes to map action methods directly to route templates.

Attribute routing directly defines the routes on action methods. We can also use these attributes on the controllers. It enables us to get fine-grained control over what routes map to which actions. With attribute routing, the controller and action names play no part in determining the action method.

For example, we use attributes Blog and Home to map an incoming URL such as **myapp.com/blog/post/3** to the **Show** method on the **PostsController**.

```
[Route("blog")]
public class PostsController : Controller
{
  [HttpGet("post/{id:int}")]
  public IActionResult Show(int id = 0)
  {
      Post post = new Post()
      {
          ID = id
      };

      return View("Show", post);
  }

  [HttpGet("edit/{id:int}")]
  public IActionResult Edit(int id)
  {
      Post postToEdit = _service.Get(id);

      return View("Edit", postToEdit);
  }
}
```

In the above example, the attribute **[Route("blog")]** is placed on the controller, whereas the route **[HttpGet("post/{id:int}")]** is placed on the action method. A controller route applies to all actions in that controller. For example, the second **["edit/{id:int}"]** route matches the url **myapp.com/blog/edit/3**.

In addition to the above route templates, ASP.NET Core provides the following HTTP verb templates.

- [HttpGet]
- [HttpPost]
- [HttpPut]
- [HttpDelete]
- [HttpHead]
- [HttpPatch]

30. What is a RESTful Web Service or a Web API?

Not all web applications return an HTML view as a response to an HTTP request. Sometimes, a client only wants some data from your application, and it wants to handle how that data will be formatted.

For example, let's say your application supports both web and mobile interfaces. Instead of writing two separate projects which return HTML and mobile views, you can write a single application that only returns the specific data that the clients need. Once the clients receive this data, they format it accordingly. The web client renders the HTML using view templates and JavaScript, and the mobile clients generate the appropriate mobile view for its specific platform.

An application might also need to communicate with another application to fetch the data that it needs. For example, when you go to Amazon.com, it communicates with hundreds of other services and applications to retrieve data and renders the final HTML page you see.

Such back-end applications, which provide data, are commonly known as RESTful web services. REST protocol uses verbs like **GET, POST, PUT, DELETE** to communicate between multiple applications. The client and server can be written in different languages and technologies and still work together without knowing about each other, as long as each side knows the format of the data that is getting sent.

ASP.NET Core supports creating RESTful services, also known as web APIs, using C#. A Web API consists of one or more controllers that derive from ControllerBase class.

```
[PostController]
[Route("[controller]")]
public class PostController : ControllerBase
```

An MVC controller derives from the **Controller** class. However, a Web API controller should derive from the ControllerBase class. The reason is that Controller derives from ControllerBase and provides additional support for views, which you don't need for web API requests.

That said, you can use a controller for both rendering views and data. That is, it can act as both an MVC controller and a Web API controller. In this case, you can derive the controller from the Controller class.

31. What is Entity Framework?

Most applications require storing and retrieving data. Usually, we store this data in a database. Working with databases can often be rather complicated. You have to manage database connections, convert data from your application to a format the database can understand, and handle many other subtle issues.

The .NET ecosystem has libraries you can use for this, such as ADO.NET. However, it can still be complicated to manually build SQL queries and convert the data from the database into C# classes back and forth.

EF, which stands for Entity Framework, is a library that provides an object-oriented way to access a database. It acts as an object-relational mapper, communicates with the database, and maps database responses to .NET classes and objects.

Entity Framework (EF) Core is a lightweight, open-source, and cross-platform version of the Entity Framework.

Here are the essential differences between the two:

Cross-platform:

- We can use EF Core in cross-platform apps that target .NET Core.
- EF 6.x targets .NET Framework, so you're limited to Windows.

Performance:

- EF Core is fast and lightweight. It significantly outperforms EF 6.x.

Features:

- EF Core has some features that EF 6.x doesn't have (batching statements, client-side key generation, in-memory database for testing)
- EF 6.x is much more feature-rich than EF Core. EF Core is missing some headline features at the time of writing, such as lazy-loading and full server-side Group By. However, it is under active development, so those features will no doubt appear soon.

Conclusion

32. Conclusion

In this article on ASP.NET interview questions, we learned about the legacy ASP.NET framework and its modern alternative, that is ASP.NET Core. The article explored a broad range of basic and advanced questions that an interviewer would ask in a job interview for a junior/intermediate developer role. We hope it helps for your next job interview!

→Asp.Net Interview MCQs

1.

Which filter executes first in ASP.NET MVC?

- Exception filters
- Authentication filters
- Result filters
- Action filters

2.

What is ActionResult in ASP.NET MVC?

- Interface
- Variable
- Return type of a controller method
- None of the above

3.

In which format data will be returned from XML into the table?

- DataSet
- DataRow
- DataTable
- None of the above

4.

_____ is the default authentication in Internet Information Services(IIS).

- Administrator
- Anonymous
- Standard User
- All of the above

5.

Which Request Processing technique follows the ASP.NET?

- Down-Up
- Top-Down
- Pipeline
- Waterfall

6.

_____ Namespace can be used for ASPX View Engine.

- System.Web.Mvc.WebFormViewEngine
- System.Web.Razor
- Both A & B
- None of the above

7.

Which among the below filters belong to filter types in the ASP.NET MVC application?

- Authorization filters
- Result filters
- Action filters
- All of the above

8.

Which is the default Viewstart Page in ASP.NET MVC?

- _Layout.cshtml
- _ViewStart.cshtml
- _Login.cshtml
- None of the above

9.

What is AuthConfig.cs in ASP.NET MVC?

- Used for configuring user settings
- Used for configuring route settings
- Used for configuring security settings including sites oAuth Login
- None of the above

10.

What is RouteConfig.cs in ASP.NET MVC?

- Used for registering global MVC bundles
- Used for registering MVC filters
- Used for registering MVC config statements and route config
- All of the above

11.

Which filter will be executed at the end in ASP.NET MVC?

- Exception filters
- Response filters
- Action filters
- Authorization filters

12.

Scaffolding is a _____.

- Code generation framework for Web applications in ASP.NET
- Framework for generating the class in ASP.NET
- Interface
- None of the above

13.

Which among the below works on the client-side?

- HiddenField
- ControlState
- ViewState
- All of the above

14.

What is meant by ViewResult()?

- A Concrete class
- An abstract class
- Both a & b
- None of the above

15.

DRY principle in ASP.NET represents _____.

- Don’t Revise Yourself
- Don’t Repeat Yourself
- Don’t Read Yourself
- None of the above

→3.C SHARP C# IMP INERVIEW QUESTIONS:-

→REFERENCE→ INTERVIEW BIT:-

C sharp is an **object-oriented programming** language developed by Microsoft. C# is used with the .NET framework for creating websites, applications, and games. C# has a varied reason for being popular:

Comparatively easier: Starting with C# is termed comparatively easier than other programming languages
Wider use of development: Using C#, you are prone to create web applications or gaming apps. C# has some fantastic features like automatic garbage collection, interfaces, etc. which help build better applications.
Larger target audience: Collaboration with Microsoft provides an edge to the applications created using C# since it would have a wider target audience.

Since C# is such a widely-used programming language, a plethora of big and small organizations base their products using it. So, prepare yourself with basic and advanced level C# questions to ace the interviews.

→3.C# Basic Interview Questions

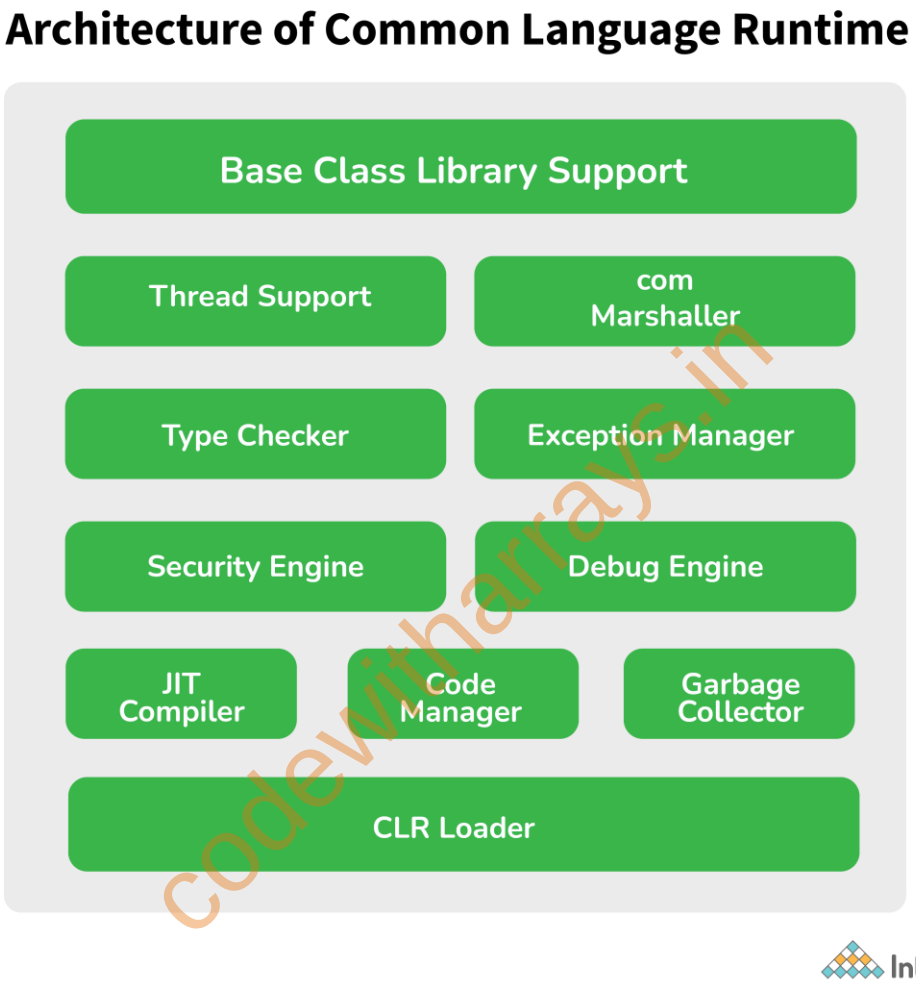
1. How is C# different from C?

You would always know C being the procedural language while C# is a more object-oriented language. The biggest difference is that C# supports automatic garbage collection by

Common Language Runtime (CLR) while C does not. C# primarily needs a **.NET framework** to execute while C is a platform-agnostic language.

2. What is Common Language Runtime (CLR)?

CLR handles program execution for various languages including C#. The architecture of CLR handles memory management, garbage collection, security handling, and looks like:



Architecture of CLR

3. What is garbage collection in C#?

Garbage collection is the process of freeing up memory that is captured by unwanted objects. When you create a class object, automatically some memory space is allocated to the object in the heap memory. Now, after you perform all the actions on the object, the memory space occupied by the object becomes waste. It is necessary to free up memory. Garbage collection happens in three cases:

- If the occupied memory by the objects exceeds the pre-set threshold value.
- If the garbage collection method is called
- If your system has low physical memory

4. What are the types of classes in C#?

Class is an entity that encapsulates all the properties of its objects and instances as a single unit. C# has four types of such classes:

- **Static class:** Static class, defined by the keyword 'static' does not allow inheritance. Therefore, you cannot create an object for a static class.

Sample code:

```
static class classname
{
  //static data members
  //static methods
}
```

- **Partial class:** Partial class, defined by the keyword 'partial' allows its members to partially divide or share source (.cs) files.
- **Abstract class:** Abstract classes are classes that cannot be instantiated where you cannot create objects. Abstract classes work on the OOPS concept of abstraction. Abstraction helps to extract essential details and hide the unessential ones.
- **Sealed class:** Sealed classes are classes that cannot be inherited. Use the keyword sealed to restrict access to users to inherit that class.


```
sealed class InterviewBit
{
   // data members
   // methods
   .
   .
   .
}
```

5. What is a managed and unmanaged code?

Managed code lets you run the code on a managed CLR runtime environment in the .NET framework.
Managed code runs on the managed runtime environment than the operating system itself.
Benefits: Provides various services like a garbage collector, exception handling, etc.

Unmanaged code is when the code doesn't run on CLR, it is an unmanaged code that works outside the .NET framework.
They don't provide services of the high-level languages and therefore, run without them.
Such an example is C++.

6. What is the difference between an abstract class and an interface?

Let's dig into the differences between an abstract class and an interface:

- Abstract classes are classes that cannot be instantiated ie. that cannot create an object. The interface is like an abstract class because all the methods inside the interface are abstract methods.
- Surprisingly, abstract classes can have both abstract and non-abstract methods but all the methods of an interface are abstract methods.
- Since abstract classes can have both abstract and non-abstract methods, we need to use the Abstract keyword to declare abstract methods. But in the interface, there is no such need.

An abstract class has constructors while an interface encompasses none.

Ex.

Abstract class:

```
public abstract class Shape{
public abstract void draw();
}
```

Interface:

```
public interface Paintable{
void paint();
}
```

7. What are the differences between ref and out keywords?

C# ref keywords pass arguments by reference and not value. To use the 'ref' keyword, you need to explicitly mention 'ref'.

```
void Method(ref int refArgument)
{
   refArgument = refArgument + 10;
}
int number = 1;
Method(ref number);
Console.WriteLine(number);
// Output: 11
```


C# out keywords pass arguments within methods and functions.
'out' keyword is used to pass arguments in a method as a reference to return multiple values. Although it is the same as the ref keyword, the ref keyword needs to be initialised before it is passed. Here, The out and ref keywords are useful when we want to return a value in the same variables that are passed as an argument.

```
public static string GetNextFeature(ref int id)
{
   string returnText = "Next-" + id.ToString();
   id += 1;
   return returnText;
}
public static string GetNextFeature(out int id)
{
   id = 1;
   string returnText = "Next-" + id.ToString();
```

```
    return returnText;
}
```

8. What are extension methods in C#?

Extension methods help to add new methods to the existing ones. The methods that are added are static. At times, when you want to add methods to an existing class but don't perceive the right to modify that class or don't hold the rights, you can create a new static class containing the new methods. Once the extended methods are declared, bind this class with the existing one and see the methods will be added to the existing one.

```
// C# program to illustrate the concept
// of the extension methods
using System;

namespace ExtensionMethod {
static class NewMethodClass {

    // Method 4
    public static void M4(this Scaler s)
    {
        Console.WriteLine("Method Name: M4");
    }

    // Method 5
    public static void M5(this Scaler s, string str)
    {
        Console.WriteLine(str);
    }
}

// Now we create a new class in which
// Scaler class access all the five methods
public class IB {

    // Main Method
    public static void Main(string[] args)
    {
        Scaler s = new Scaler();
        s.M1();
        s.M2();
        s.M3();
        s.M4();
        s.M5("Method Name: M5");
    }
}
}
```


Output:

Method Name: M1

Method Name: M2

Method Name: M3

Method Name: M4

Method Name: M5

9. What are Generics in C#?

In C# collections, defining any kind of object is termed okay which compromises C#'s basic rule of type-safety. Therefore, generics were included to type-safe the code by allowing re-use of the data processing algorithms. Generics in C# mean not linked to any specific data type. Generics reduce the load of using boxing, unboxing, and typecasting objects. Generics are always defined inside angular brackets <>. To create a generic class, this syntax is used:

```
GenericList<float> list1 = new GenericList<float>();
GenericList<Features> list2 = new GenericList<Features>();
GenericList<Struct> list3 = new GenericList<Struct>();
```

Here, GenericList<float> is a generic class. In each of these instances of GenericList<T>, every occurrence of T in the class is substituted at run time with the type argument. By substituting the T, we have created three different type-safe using the same class.

10. What is the difference between an Array and ArrayList in C#?

An array is a collection of similar variables clubbed together under one common name. While ArrayList is a collection of objects that can be indexed individually. With ArrayList you can access a number of features like dynamic memory allocation, adding, searching, and sorting items in the ArrayList.

- When declaring an array the size of the items is fixed therefore, the memory allocation is fixed. But with ArrayList, it can be increased or decreased dynamically.
- Array belongs to system.array namespace while ArrayList belongs to the system.collection namespace.
- All items in an array are of the same datatype while all the items in an ArrayList can be of the same or different data types.
- While arrays cannot accept null, ArrayList can accept null values.

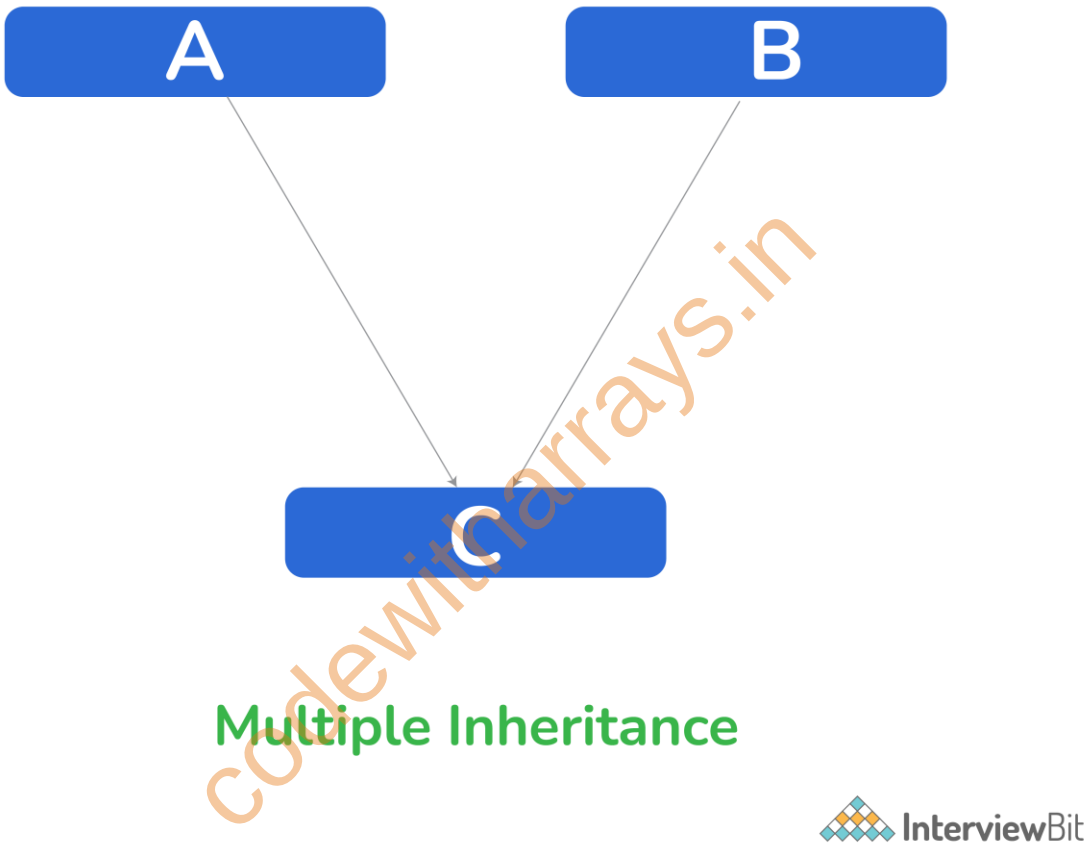
For ex.:

```
// C# program to illustrate the ArrayList
using System;
using System.Collections;

class IB {

   // Main Method
   public static void Main(string[] args)
   {

       // Create a list of strings
       ArrayList al = new ArrayList();
       al.Add("Bruno");
       al.Add("Husky");
       al.Add(10);
       al.Add(10.10);

       // Iterate list element using foreach loop
       foreach(var names in al)
```

```
        {
            Console.WriteLine(names);
        }
    }
}
```

11. What is inheritance? Does C# support multiple inheritance?

Inheritance means acquiring some of the properties from a master class.



Multiple Inheritance in C#

Here, class C can inherit properties from Class A and Class B. Here is an example of inheritance:

```
// C# program to illustrate
// multiple class inheritance
using System;
using System.Collections;

// Parent class 1
class Scaler {

  // Providing the implementation
  // of features() method
  public void features()
  {
```

```
        // Creating ArrayList
        ArrayList My_features= new ArrayList();

        // Adding elements in the
        // My_features ArrayList
        My_features.Add("Abstraction");
        My_features.Add("Encapsulation");
        My_features.Add("Inheritance");

        Console.WriteLine("Features provided by OOPS:");
        foreach(var elements in My_features)
        {
            Console.WriteLine(elements);
        }
    }
}

// Parent class 2
class Scaler2 :Scaler{

    // Providing the implementation
    // of courses() method
    public void languages()
    {

        // Creating ArrayList
        ArrayList My_features = new ArrayList();

        // Adding elements in the
        // My_features ArrayList
        My_features.Add("C++");
        My_features.Add("C#");
        My_features.Add("JScript");


        Console.WriteLine("\nLanguages that use OOPS concepts:");
        foreach(var elements in My_features)
        {
            Console.WriteLine(elements);
        }
    }
}

// Child class
class ScalertoScaler : Scaler2 {
}

public class Scaler1 {

    // Main method
    static public void Main()
    {

        // Creating object of ScalertoScaler class
        ScalertoScaler obj = new ScalertoScaler();
        obj.features();
        obj.languages();
    }
}
```

Also, C# doesn't support multiple inheritances.
Instead, you can use interfaces to inherit the properties using the class name in the signature.

→C# Advanced Interview Questions

12. What is Boxing and Unboxing in C#?

The two functions are used for typecasting the data types:

Boxing: Boxing converts value type (int, char, etc.) to reference type (object) which is an implicit conversion process using object value.

Example:

```
int num = 23; // 23 will assigned to num
Object Obj = num; // Boxing
```

Unboxing: Unboxing converts reference type (object) to value type (int, char, etc.) using an explicit conversion process.

Example:

```
int num = 23;          // value type is int and assigned value 23
Object Obj = num;     // Boxing
int i = (int)Obj;     // Unboxing
```

13. What are Properties in C#?

Properties in C# are public members of a class where they provide the ability to access private members of a class. The basic principle of encapsulation lets you hide some sensitive properties from the users by making the variables private. The private members are not accessible otherwise in a class. Therefore, by using properties in C# you can easily access the private members and set their values.

The values can be easily assigned using get and set methods, also known as accessors. While the get method extracts the value, the set method assigns the value to the variables.

14. What are partial classes in C#?

Partial classes implement the functionality of a single class into multiple files. These multiple files are combined into one during compile time. The partial class can be created using the partial keyword.

```
public partial Clas_name
{
        // code
}
```

You can easily split the functionalities of methods, interfaces, or structures into multiple files. You can even add nested partial classes.

15. What is the difference between late binding and early binding in C#?

Late binding and early binding are examples of one of the primary concepts of OOPS: **Polymorphism.**

For ex: one function calculateBill() will calculate bills of premium customers, basic customers, and semi-premium customers based on their policies differently. The calculation for all the customer objects is done differently using the same function which is called polymorphism.

When an object is assigned to an object variable in C#, the .NET framework performs the binding.

When the binding function happens at compile-time, it is called early binding. It investigates and checks the methods and properties of the static objects. With early binding, the number of run-time errors decreases substantially and it executes pretty quickly.

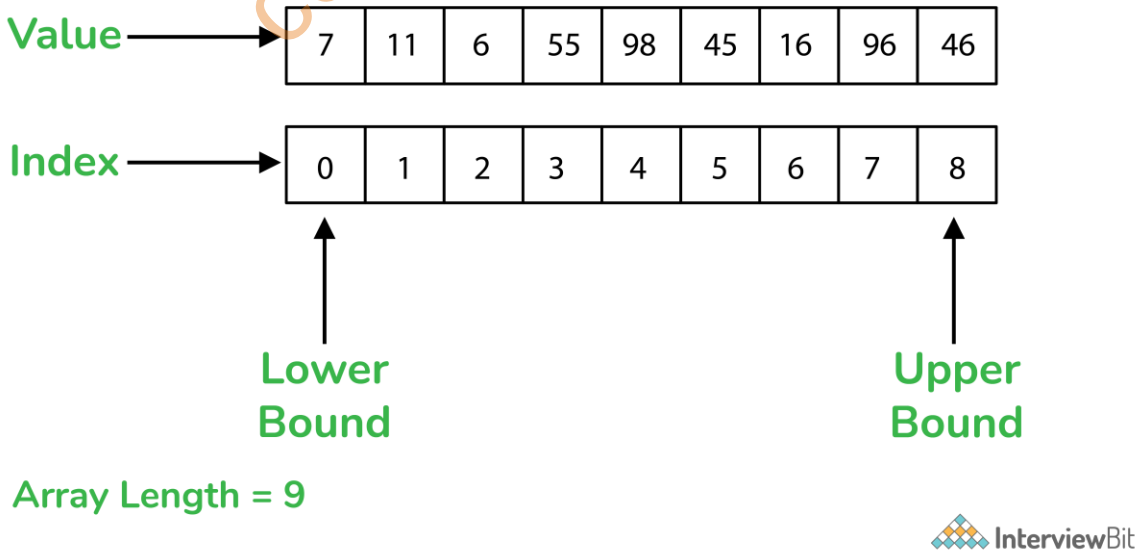
But when the binding happens at runtime, it is called late binding. Late binding happens when the objects are dynamic (decided based on the data they hold) at run-time. It is slower as it looks through during run-time.

16. What are the Arrays in C#?

When a group of similar elements is clubbed together under one name, they are called arrays.

For ex. An array of tea Atea[4]: [green tea, chamomile tea, black tea, lemon tea]. The length of the array defines how many elements are present in the array.

In C#, the memory allocations for the elements of the array happen dynamically. This is how values are stored in an array sequentially.



Arrays in C#

A few pointers for arrays in C#:

- The memory allocation is DYNAMIC.
- Arrays in C# are treated as objects.
- The length of the array is easy to find by detecting the number of members in the array.
- The members in the array are ordered and begin with the index value=0.
- The array types are reference types derived from the base array type.

Syntax: `< Data Type > [ ] < Name_Array >`

17. What are Indexers in C#?

Indexers are called smart arrays that allow access to a member variable. Indexers allow member variables using the features of an array. They are created using the Indexer keyword. Indexers are not static members.

For ex. Here the indexer is defined the same way.

```
<return type> this[<parameter type> index]
{
   get{
       // return the value from the specified index of an internal
collection
   }
   set{
       // set values at the specified index in an internal collection
   }
}
```


18. Difference between the Equality Operator (==) and Equals() Method in C#?

Although both are used to compare two objects by value, still they both are used differently.

For ex.:

```
int x = 10;
int y = 10;
Console.WriteLine( x == y);
Console.WriteLine(x.Equals(y));
Output:
True
True
```

Equality operator (==) is a reference type which means that if equality operator is used, it will return true only if both the references point to the same object.

Equals() method: Equals method is used to compare the values carried by the objects. int x=10, int y=10. If x==y is compared then, the values carried by x and y are compared which is equal and therefore they return true.

Equality operator: Compares by reference

Equals(): Compares by value

19. What are the different ways in which a method can be Overloaded in C#?

Overloading means when a method has the same name but carries different values to use in a different context. Only the main() method cannot be overloaded.

In order to overload methods in C#,

- Change the number of parameters in a method, or
- Change the order of parameters in a method, or
- Use different data types for parameters
  In these ways, you can overload a method multiple times.

For ex.

```
public class Area {
   public double area(double x) {
        double area = x * x;
        return area;
   }
   public double area(double a, double b) {
        double area = a * b;
        return area;
   }
}
```

Here, the method Area is used twice. In the first declaration, one argument is used while in the second one, there were two arguments are used. Using different parameters in the same method, we were able to overload the method area().

20. What is Reflection in C#?

Reflection in C# extracts metadata from the datatypes during runtime.

To add reflection in the .NET framework, simply use System.Refelction namespace in your program to retrieve the type which can be anything from:

- Assembly
- Module
- Enum
- MethodInfo
- ConstructorInfo
- MemberInfo
- ParameterInfo
- Type
- FieldInfo
- EventInfo
- PropertyInfo

21. What is the difference between constant and readonly in C#?

A **const** keyword in C# is used to declare a constant field throughout the program. That means once a variable has been declared **const**, its value cannot be changed throughout the

program.

In C#, a constant is a number, string, null reference, or boolean values.

For ex:

```
class IB {

   // Constant fields
   public const int xvar = 20;
   public const string str = "InterviewBit";

   // Main method
   static public void Main()
   {

       // Display the value of Constant fields
       Console.WriteLine("The value of xvar: {0}", xvar);
       Console.WriteLine("The value of str: {0}", str);
   }
}
Output:
The value of xvar is 20.
The value of string is Interview Bit
```

On the other hand, with readonly keyword, you can assign the variable only when it is declared or in a constructor of the same class in which it is declared.

Ex:

```
public readonly int xvar1;
   public readonly int yvar2;

   // Values of the readonly
   // variables are assigned
   // Using constructor
   public IB(int b, int c)
   {

       xvar1 = b;
       yvar2 = c;
       Console.WriteLine("The value of xvar1 {0}, "+
                       "and yvar2 {1}", xvar1, yvar2);
   }

   // Main method
   static public void Main()
   {
     IB obj1 = new IB(50, 60);
   }
}

Output:
The value of xvar1 is 50, and yvar2 is 60
```

Constants are static by default while readonly should have a value assigned when the constructor is declared.

Constants can be declared within functions while readonly modifiers can be used with reference types.

22. What is the difference between String and StringBuilder in C#?

The major difference between String and StringBuilder is that String objects are immutable while StringBuilder creates a mutable string of characters. StringBuilder will make the changes to the existing object rather than creating a new object.

StringBuilder simplifies the entire process of making changes to the existing string object. Since the String class is immutable, it is costlier to create a new object every time we need to make a change. So, the StringBuilder class comes into picture which can be evoked using the System.Text namespace.

In case, a string object will not change throughout the entire program, then use String class or else StringBuilder.

For ex:

```
string s = string.Empty;
for (i = 0; i < 1000; i++)
  {
    s += i.ToString() + " ";
  }
```

Here, you'll need to create 2001 objects out of which 2000 will be of no use.

The same can be applied using StringBuilder:

```
StringBuilder sb = new StringBuilder();
for (i = 0; i < 1000; i++)
 {
   sb.Append(i); sb.Append(' ');
 }
```


By using StringBuilder here, you also de-stress the memory allocator.

→C# Coding Problems

23. Write a program in C# Sharp to reverse a string?

```
internal static void ReverseString(string str)
{

  char[] charArray = str.ToCharArray();
  for (int i = 0, j = str.Length - 1; i < j; i++, j--)
  {
      charArray[i] = str[j];
      charArray[j] = str[i];
  }
  string reversedstring = new string(charArray);
  Console.WriteLine(reversedstring);
```

24. Write a program in C# Sharp to reverse the order of the given words?

```
internal static void ReverseWordOrder(string str)
{
  int i;
  StringBuilder reverseSentence = new StringBuilder();
   int Start = str.Length - 1;
  int End = str.Length - 1;
   while (Start > 0)
  {
      if (str[Start] == ' ')
      {
          i = Start + 1;
          while (i <= End)
          {
              reverseSentence.Append(str[i]);
              i++;
          }
          reverseSentence.Append(' ');
          End = Start - 1;
      }
      Start--;
  }
   for (i = 0; i <= End; i++)
  {
      reverseSentence.Append(str[i]);
  }
  Console.WriteLine(reverseSentence.ToString());
}
```

25. Write a program in C# Sharp to find if a given string is palindrome or not?

```
internal static void chkPalindrome(string str)
{
   bool flag = false;
   for (int i = 0, j = str.Length - 1; i < str.Length / 2; i++, j--)
   {
       if (str[i] != str[j])
       {
           flag = false;
           break;
       }
       else
           flag = true;
   }
   if (flag)
   {
       Console.WriteLine("Palindrome");
   }
   else
       Console.WriteLine("Not Palindrome");
```

Output:

Input: Key Output: Not Palindrome
Input: step on no pets Output: Palindrome

26. Write a C# program to find the substring from a given string.

```
internal static void findallsubstring(string str)
{
   for (int i = 0; i < str.Length; ++i)
```

```
    {
        StringBuilder subString = new StringBuilder(str.Length - i);
        for (int j = i; j < str.Length; ++j)
        {
            subString.Append(str[j]);
            Console.Write(subString + " ");
        }
    }
}
```

27. Write a C# program to find if a positive integer is prime or not?

```
 static void Main(string[] args)
{
    if (FindPrime(47))
    {
        Console.WriteLine("Prime");
    }
    else
    {
        Console.WriteLine("Not Prime");
    }
    Console.ReadLine();
}
internal static bool FindPrime(int number)
{
    if (number == 1) return false;
    if (number == 2) return true;
    if (number % 2 == 0) return false;
     var squareRoot = (int)Math.Floor(Math.Sqrt(number));
     for (int i = 3; i <= squareRoot; i += 2)
    {
        if (number % i == 0) return false;
    }
     return true;
}
```


→C# MCQ'S

1.

What is the need for 'Conversion of data type' in C#?

- To store a value of one data type into a variable of another data type
- To get desired data
- To prevent situations of runtime error during change or conversion of data type
- None of the mentioned

2.

What are the types of 'Data Conversion' in C#?

- Implicit Conversion

- Explicit Conversion
- Implicit Conversion and Explicit Conversion
- None of the mentioned

3.

What is the subset of ‘int’ datatype?

- long, ulong, ushort
- long, ulong, uint
- long, float, double
- long, float, ushort

4.

What will be the output of the following C# code?

```
static void Main(string[] args)
{
    int a, b, c, x;
    a = 90;
    b = 15;
    c = 3;
    x = a - b / 3 + c * 2 - 1;
    Console.WriteLine(x);
    Console.ReadLine();
}
```

- 90
- 92
- 89
- 88

5.

The correct way of incrementing the operators are:

- d =+ 1
- c += 1
- b ++ 1

- ++ a ++

6.

Which reference modifier is used to define reference variables?

- &
- ref
- #
- $

7.

What will be the output of the following C# code?

```
static void Main(string[] args)
{
   int []a = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
   func(ref a);
   Console.ReadLine();
}
static void func(ref int[] x)
{
   Console.Write(" numbers are : ");
   for (int i = 0; i < x.Length; i++)
   {
       if (x[i] % 2 == 0)
       {
           x[i] = x[i] + 1;
           Console.Write(x[i] +" ");
       }
   }
}
```

- numbers are : 3 5 7 9 11
- numbers are : 2 4 6 8 10
- numbers are : 2 3 4 5 6
- none of the mentioned

8.

What will be the output of the following code?

```
static void Main(string[] args)
  {
      int x = 4 ,b = 2;
      x -= b/= x * b;
```

```
        Console.WriteLine(x + " " + b);
        Console.ReadLine();
    }
```

- 4 2
- 0 4
- 2 2
- 4 0

9.

What will be the output of the following C# expression?

```
int  a+= (float) b/= (long)c
```

- float
- int
- long
- none of the mentioned

10.

Which of the following modifiers is used when an abstract method is redefined by a derived class?

- Override
- Overloads
- Base
- Virtual

11.

What is the process of defining a method in terms of itself, that is a method that calls itself?

- Polymorphism
- Abstraction
- Encapsulation

Recursion

12.

What is the output of this code?

```
class maths
{
   public int fun(int k, int y)
   {
       return k + y;
   }
   public int fun1(int t, float z)
   {
       return (t+(int)z);
   }
}
class Program
{
   static void Main(string[] args)
   {
       maths obj = new maths();
       int i;
       int b = 90;
       int c = 100;
       int d = 12;
       float l = 14.78f;
       i = obj.fun(b, c);
       Console.WriteLine(i);
       int j = (obj.fun1(d,  l));
       Console.WriteLine(j);
       Console.ReadLine();
   }
}
```

190, 26.78f

0, 26.78f

190, 26

190, 0

→4.ADO.NET INTERVIEW QUESTIONS:-

→REFERENCE:- INTERVIEW BIT:-

Introduction to ADO.NET

ADO.NET is a technology used for data and is provided by the Microsoft .NET Framework. It is a part of the .NET framework that supports the communication between relational and non-relational systems with the help of a set of software components. It supports disconnected architecture using which programmers are allowed to access data and data services from a database without depending on the data source.

ADO.NET is comprised of a group of built-in classes that are useful for establishing the database connection, for gaining access to XML, relational data, and application data, and for retrieval of a result. It can be used in various programming languages such as Visual Basic.NET, Visual C++, etc., that are supported by the .NET framework.

Advantages of ADO.NET

ADO.NET has various advantages which can be categorized into the following categories:

- **Interoperability:** It provides the ability to communicate across heterogeneous environments, once the connection has been established between them.
- **Scalability:** It provides the ability to serve an increasing number of clients without reducing the performance of the system. So we can say that ADO.NET is highly scalable because it is flexible enough to be easily expanded when there is a requirement for the same.
- **Productivity:** It provides the ability to rapidly develop robust applications for data access using rich and extensible component object models provided by the ADO.NET.
- **Performance:** An improvement over earlier ADO.NET versions because of the disconnected data model. It can establish connections quickly to fetch data without any delay.

Scope of ADO.NET

ADO.NET being one of the products of Microsoft, it is good enough to position itself strongly in the market. ADO.NET has massive community support, so it is definitely having a large scope ahead. You could learn ADO.NET along with hands-on experience on the .Net frame

work in order to have a good scope. Any full-stack developer who has a better grip over both front-end and back-end technology can precisely learn ADO.NET.

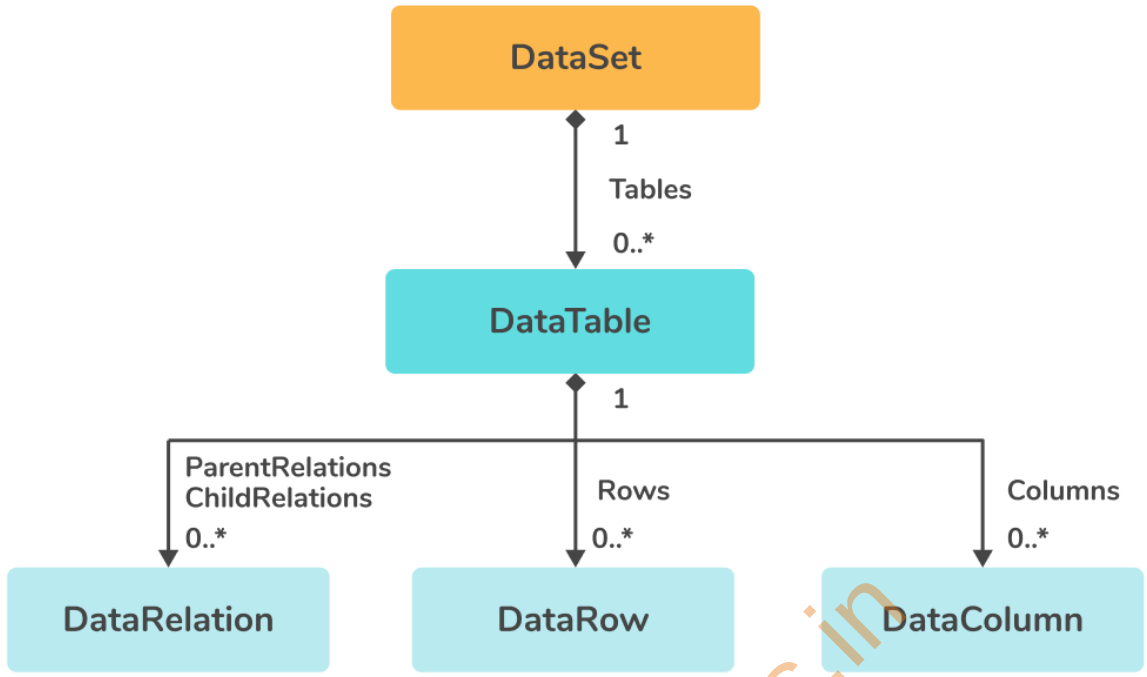
→4.ADO.NET Interview Questions

1. What is ADO.NET?

- ADO.NET stands for ActiveX Data Object, it is a part of the .NET Framework by Microsoft. ADO.NET framework provides a set of classes that are used to handle data communication with data sources such as XML files and databases (such as SQL, Oracle, MySQL, MS Access, etc.).
- ADO.NET can separate mechanisms for data connectivity, data access, and data manipulation.
- It has introduced the disconnected architecture, in which data can be stored in a DataSet. ADO.NET has providers for database connection, commands for execution, and result retrieval.
- The ADO.NET classes are stored in the DLL named `System.Data.dll`.
- Various applications like ASP.NET applications, console applications, windows applications, etc., will use ADO.NET for database connection, command execution, and retrieval of data.

2. What is DataSet in ADO.NET?

- The DataSet is a collection of database tables(row and column format) that contain the data. It is helpful for fetching the data without any need for Data Source interaction, that is why it is called a disconnected data access method.
- It is an in-memory data store that can contain multiple tables at the same time. DataRelation objects can be used to relate these tables.
- For creating a DataSet object, ADO.NET provides a DataSet class that consists of constructors and methods to carry out data-related operations.
- It can be used with various data sources, with XML data, or to manage the application's local data. The DataSet will include related tables, data constraints, and relationships among the tables.



InterviewBit

Structure Of A DataSet

3. Give the differences between ADO and ADO.NET.

ADO	ADO.NET
It is Component Object Modelling(COM) based.	It is Common Language Runtime(CLR) based.
It works in connected mode to access the data store.	It does require an active connection, works in disconnected mode to access the data store.
It uses the RecordSet object to access and store data from the data sources.	It uses a DataSet object to access and store data from the data sources.
It provides a feature of locking.	It does not provide a feature of locking.
Data is stored in binary form.	Data is stored in XML.
It does not support XML integration.	It supports XML integration.
Using a single connection instance, it is not possible to send multiple transactions.	Using a single connection instance you can send multiple transactions.
We can create only client-side cursors.	Both client-side and server-side cursors can be created.
It supports sequential row access in a RecordSet.	Non-sequential data access is supported in DataSet by using a collection-based hierarchy.

ADO	ADO.NET
It will make use of SQL JOINs and UNIONs for combining data from multiple tables. It is not possible to fetch records from multiple tables independently.	It will make use of DataRelational objects to combine data from multiple tables without the help of JOINs and UNIONs. Therefore records from multiple tables are maintained independently.

4. What is a DataAdapter in ADO.NET?

- A DataAdapter is used to access data from a data source by functioning as a bridge between DataSet and a data source. DataAdapter class includes an SQL command set and a database connection. It is helpful to fill the DataSet and resolve changes to the data source.
- The DataAdapter will make use of the Connection object that belongs to the .NET Framework data provider for connecting with a data source. Along with that, it will also use Command objects to retrieve data from the data source as well as to resolve changes to the data source.
- DataAdapter properties that permit the user to control the database are the Select command, Update command, Insert command, and Delete command.
- Example code for the usage of DataAdapter:

```
using System;
using System.Data.SqlClient;
using System.Data;
namespace DataAdapterExample
{
    public partial class DataAdapterDemo : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            using (SqlConnection conn = new SqlConnection("data source=.;
database=items; integrated security=SSPI"))
            {
                SqlDataAdapter da = new SqlDataAdapter("Select * from
items", conn);
                DataSet s = new DataSet();
                da.Fill(s);
                GridView1.DataSource = s;
                GridView1.DataBind();
            }
        }
    }
}
```


Here, DataAdapter will receive the data from the items table and fill the DataSet, which will be later used to display the information retrieved from the items database.

5. Explain the difference between ADO.NET and ASP.NET.

ADO.NET(ActiveX Data Objects)	ASP.NET(Active Server Pages)
ADO.NET is a Library within the .NET framework.	ASP.NET is a Framework.
It is a technology useful for accessing data from databases.	It is a technology useful for the creation of dynamic web pages.

ADO.NET(ActiveX Data Objects)	ASP.NET(Active Server Pages)
Here, data can be converted into XML format.	Here, We can write our code into VB.Net, C#, ASP.Net, etc.
It is used to develop reliable and scalable database applications with high performance for client-server applications.	It is used to create dynamic web pages, web applications, websites, and web services.

6. Explain about DataSet types in ADO.NET.

DataSet can be said as a collection of database tables(row and column format) that holds the data. There are two types of DataSet in ADO.NET. They are:

1. **Typed DataSet:** A typed DataSet is derived from the DataSet base class and can be created by selecting the DataSet option provided by Visual Studio. It will be created as an XML schema(.xsd file) that contains DataSet structure information such as rows, columns, and tables. Data from the database is moved into a dataset and from the dataset to another component in the XML format.
2. **Untyped DataSet:** Untyped DataSet does not have an associated XML schema with it. Users are supposed to add columns, tables, and other elements to it. Properties can be set during design time or can add them during run time.

Example program for the usage of DataSet:

```
using System;
using System.Data.SqlClient;
using System.Data;
namespace DataSetDemo
{
    public partial class DataSetExample : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            using (SqlConnection conn = new SqlConnection("data source=.;
database=employee; integrated security=SSPI"))
            {
                SqlDataAdapter da = new SqlDataAdapter("Select * from
employee", conn);
                DataSet d = new DataSet();
                da.Fill(d);
                GridView1.DataSource = d;
                GridView1.DataBind();
            }
        }
    }
}
```

Here, DataSet will be filled by DataAdapter that receives data from the employee table. This DataSet will be used to display the information received from the employee database.

7. Explain the difference between DataTable and DataSet.

DataTable	DataSet
DataTable consists of a single database table that is placed within a memory.	DataSet consists of a collection of multiple database tables which is placed within a memory.
It has a row and column collection.	It has a database table collection.
It allows fetching only a single TableRow at a time.	It allows fetching multiple TableRows at a time.
It is a single database table, so there will not be any relation with other tables.	It represents a collection of DataTable objects, so there might be a relation between them to obtain a particular result.
In this, DataSource objects are not serialized.	In this, DataSource objects are serialized.
UniqueConstraint and ForeignKeyConstraint objects are not available enforcing data integrity.	UniqueConstraint and ForeignKeyConstraint objects are available for enforcing data integrity.

8. What are the different namespaces available in ADO.NET?

Various namespaces available under ADO.NET is given below:

1. **`System.Data`:** It contains the definition for rows, columns, relations, views, tables, constraints, and databases.
2. **`System.Data.SqlClient`:** It is a collection of classes that are helpful in connecting to a Microsoft SQL Server database such as SqlConnection, SqlCommand, SqlDataAdapter, etc.
3. **`System.Data.Odbc`:** It consists of classes that are required for connecting with most Odbc Drivers. These classes include OdbcConnection, OdbcCommand.
4. **`System.Data.OracleClient`:** It has classes required for connection with an Oracle database, OracleConnection, OracleCommand.

9. What is object pooling?

Object pooling is a repository of the objects in memory that can be reused later without creating them. This object pooling reduces the burden of creating objects when it is required. Whenever there is a requirement of an object, the object pool manager will process the request and serve accordingly. It is designed for optimizing the use of limited resources so that the demands of client requests will be fulfilled.

10. Differentiate DataSet and DataReader.

DataSet	DataReader
DataSet provides read/write access to data, so we can update the data.	DataReader provides read-only access to data, so we can't update the data.

DataSet	DataReader
It has a disconnected architecture, which means the data obtained from the database can be accessed even after the database connection was closed.	It has a connected architecture, which means to access the data retrieved from the database, the connection must be opened.
It supports various database tables from different databases.	It supports only a single table from a single database.
It provides slower access to data due to overhead.	It provides faster access to data.
Both forward and backward scanning of data is possible.	Only forward scanning of data is possible.

11. What are the different execute() methods available in ADO.NET?

Different execute() methods supported by SqlCommandObject in ADO.NET is given below:

- **`ExecuteScalar()`:** This method returns only a single value from the first row and first column of the ResultSet after the execution of the query. Even if ResultSet is having more than one row or column, all those rows and columns will be ignored. If the ResultSet is empty, it will return NULL.
- **`ExecuteNonQuery()`**: This method returns the number of rows affected by the execution of a query. This method is not useful to return the ResultSet.
- **`ExecuteReader()`**: This method returns an object of DataReader which is a read-only and forward-only ResultSet. It needs a live connection with the Data Source. We cannot directly instantiate the DataReader object. A valid DataReader object can be created with the help of the ExecuteReader() method.
- **`ExecuteXmlReader()`**: This method builds an object of the XmlReader class and will return the ResultSet in the form of an XML document. This method is made available in SQL Server 2000 or later.

12. What is a transaction in ADO.NET? Explain the types of transactions available in ADO.NET.

In ADO.NET, transactions are used when you want to bind several tasks together and execute them in the form of a single unit. The transaction provides data consistency by ensuring either all of the database operations will be succeeded or all of them will be failed. For example, consider an application that performs two tasks. First, it updates an item_order table with order information. Second, it updates an item_inventory table that holds inventory information, where a number of items ordered will be debited. If any one of the tasks fails, then both updates must be rolled back.

Two types of transactions supported by ADO.NET are as follows:

- **Local Transaction:**
  - A local transaction is a single-phase transaction that is directly handled by the database. Every .NET Framework data provider has its own Transaction object for bringing out local transactions.
  - For example, if we want to perform a transaction using SQL Server database, we import a `System.Data.SqlClient` namespace. Similarly, to perform an Oracle transaction, import the `System.Data.OracleClient` namespace. A

`DbTransaction` class will be used for writing code that is independent of the provider and that requires transactions.

- **Distributed Transaction:**
  - A distributed transaction is coordinated by a transaction monitor and will make use of fail-safe mechanisms like two-phase commit for transaction resolution. This transaction will affect multiple resources.
  - If the user can make use of a distributed transaction, if he wants to do a transaction across multiple data servers such as Oracle, SQL Server, etc.
  - If you want a distributed transaction to commit, all participants must guarantee that data modification made will be permanent. Changes must remain unchanged even if the system crash or other unforeseen events occur. Even if a single participant will make this guarantee fail, then the entire transaction will fail, and updates made to data within the transaction scope are rolled back.

13. Explain the difference between OLEDB (Object Linking and Embedding DataBase) and ODBC (Open DataBase Connectivity).

OLEDB	ODBC
An API(Application Programming Interface) that allows accessing data from different sources in a uniform manner.	It is an API for accessing DBMS (DataBase Management System).
It supports both relational and non-relational databases.	It supports only relational databases.
It is procedural-based.	It is component-based.
It is easier to deploy.	It is difficult to deploy.
It gives a higher performance on loading and extracting the data.	It performs less compared to OLE DB on loading and extraction of data.
`OleDbConnection = New OleDbConnection(connetionString)` is used to make connection with OLE DB data source.	`resource odbc_connect(string datasource , string username , string password , [int cursor_type ])` is used to make a connection to an ODBC data source. On success, this function will return a connection resource handle that is helpful in accessing the database using subsequent commands.

14. What is data binding in ADO.NET?

- Data binding in ADO.NET is the process through which user interface (UI) controls of a client application are configured to update or fetch data from data sources like a database or XML document. Using data binding, the user will be able to bind values to the particular control.
- There are two types of data binding based on the type of binding offered:

1. **Simple data binding:** It is the process of binding the control with only one value in the dataset. The controls such as label, text box will be made bound to the control using the control properties.
2. **Complex data binding:** It is the method of binding the component with the Database. The controls can be a Dropdown list, GridView, or combo box. One or more than one value can be displayed from the dataset using the complex data binding.

15. What is Connection pooling?

The task of grouping database connections in the cache memory is to make them available whenever there is a requirement of connection. Opening a new database connection every time is a time-consuming process. Connection pooling allows you to reuse existing and active database connections, whenever there is a need, and thus increases the application performance.
By setting the pooling property into true or false in the connection string, we can enable or disable the connection pooling in the application. It is enabled by default in every application.

16. What is DataTable in ADO.NET?

DataTable in ADO.NET represents a single table in a DataSet that has in-memory relational data. The data within DataTable is local to the .NET framework-based application to which it belongs but can be populated using a DataAdapter from different data sources such as Microsoft SQL Server. The DataTable class belongs to the `System.Data` namespace within the library of .NET Framework.

DataTable can be represented in `.aspx.cs` code as given below:

```
protected void DataTableExample()
{
    SqlConnection conn = new SqlConnection("Write the database connection
string");
    conn.Open();
    SqlCommand cd = new SqlCommand("Write the query or procedure", conn);
    SqlDataAdapter d = new SqlDataAdapter(cd);
    DataTable dt = new DataTable();
    d.Fill(dt);
    grid.DataSource = dt;
    grid.DataBind();
}
```

The SQL connection and SQL command object will be created. We pass the SQL query to the object of the SQL command class. A new data table object will be created by using the DataTable class and it is filled with data using a data adapter.

17. Name some of the properties and methods provided by the DataReader in ADO.NET?

Some of the properties provided by the DataReader are as follows:

- **Depth:** It represents the depth of nesting for a row.
- **FieldCount:** It gives the total column count in a row.
- **Item:** It obtains the column value in a native format.

- **RecordsAffected:** It gives the number of transaction affected rows.
- **IsClosed:** It represents whether a data reader is closed.
- **VisibleFieldCount:** It is used to obtain the number of unhidden fields in the SqlDataReader.

Some of the methods provided by the DataReader are as follows:

- **Read():** This method reads a record from the SQL Server database.
- **Close():** It closes a SqlDataReader object.
- **NextResult():** It moves the data reader to the next result during the time of batch transactions.
- **Getxxx():** Various types of Getxxx() methods such as GetBoolean(Int32), GetChar(Int32), GetFloat(Int32), GetDouble(Int32), etc., are provided by the DataReader. These methods will read a value of a particular data type from a column. For example, GetFloat() will return a column value as a Float and GetChar as a character.

18. What are the conditions for connection pooling?

The conditions for connection pooling are:

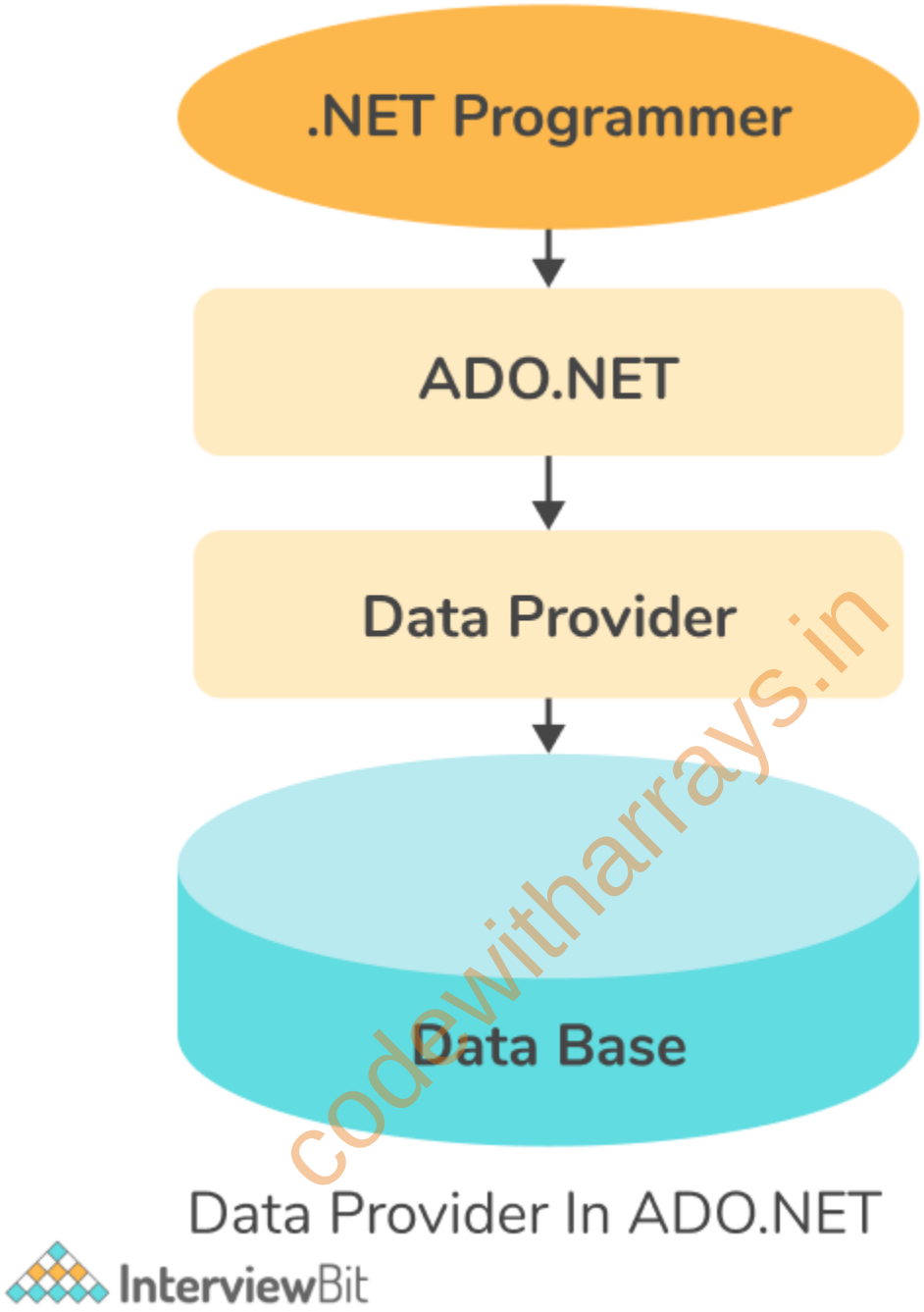
- There must be several processes with the same parameters and security settings so that they can share the same connection.
- The connection string should be identical.

19. What are the data providers in ADO.NET?

Data providers are used to transferring the data between the client application and the data store. It encapsulates the database-specific details. Data providers are helpful for database connection, data retrieval, storing the data in a dataset, reading the retrieved data, and updating the database.

The data providers that comes along with the ADO.NET Framework are:

- **OLE DB:** The OLEDB provider is available under `System.Data.OleDb` namespace. This provider can be used to access Microsoft Access, DB2/400, SyBase, and SQL Server 6.5 and earlier.
- **ODBC:** The ODBC provider is available under `System.Data.Odbc` namespace. This provider is used when there will not be any newer provider is available.
- **SQL Server:** The Microsoft SQL Server provider is available under `System.Data.SqlClient` namespace. Classes available under this provider will provide the same functionality as the generic OLEDB provider.



20. Why Stored Procedure is used in ADO.NET?

The reasons for using Stored Procedures in ADO.NET are given below:

- For improved performance
- For security reasons
- Easier to use and maintain
- Lesser Network Traffic
- Execution time is less

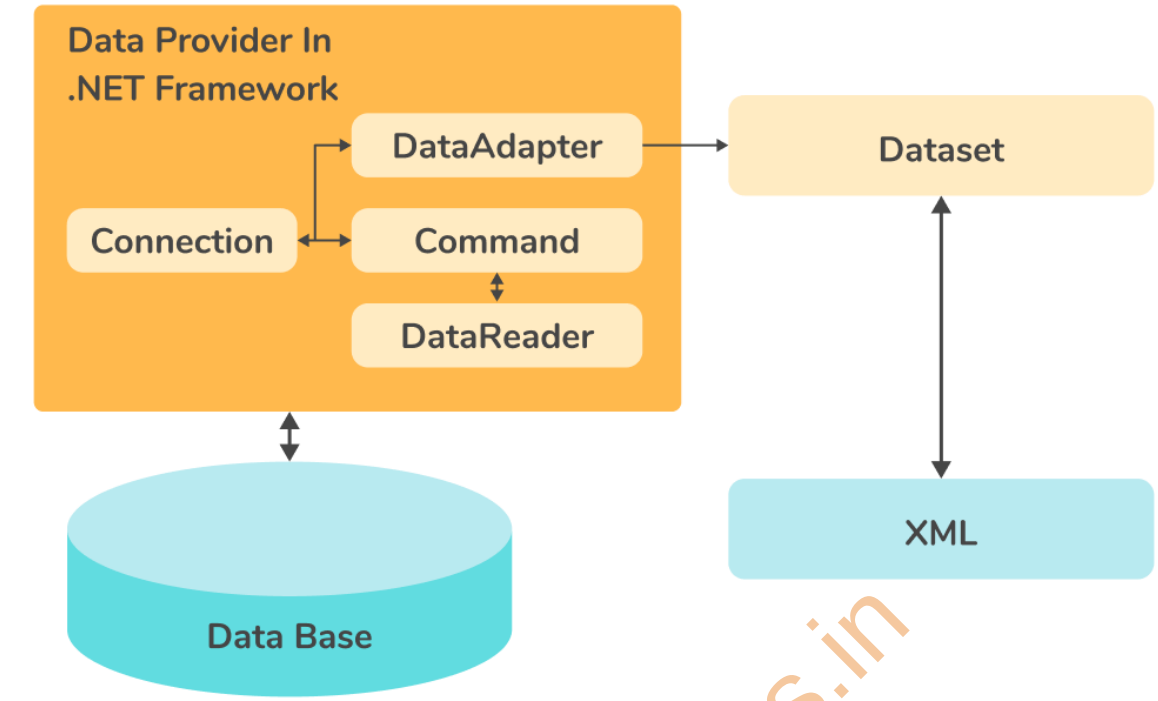
ADO.NET Interview Questions for Experienced

21. Explain ADO.NET Architecture.

ADO.NET is based on an Object Model where data residing in the database is accessed using a data provider. It is a technology of data access given by the Microsoft .Net Framework, which helps to communicate between relational and non-relational systems using a common group of components.

The components of ADO.NET architecture are:

- **Data Provider:** It provides data to all the applications that perform the database updates. The application can access data through the DataSet or DataReader object. A data provider is a having group of components such as Command, Connection, DataReader, and DataAdapter objects. Command and Connection objects are the necessary components irrespective of the operations like Insert, Delete, Select, and Update.
- **Connection:** The connection object is needed to connect with the database such as SQL Server, MySQL, Oracle, etc. To create a connection object, you must know about where the database is located(Ex: IP address or machine name, etc.) and the security credentials(Ex: user name and password-based authentication or windows authentication).
- **Command:** The command object is the component where you will write the SQL queries. Then by using the command object, execute the queries over the connection. By using the command object and SQL queries, you will be able to fetch the data or send the data to the database.
- **DataReader:** DataReader is a connected read-only RecordSet that is helpful in reading the records in the forward-only mode.
- **DataAdapter:** The DataAdapter acts as a bridge between the dataset and command object. It receives the data from the command object and puts it into the data set.
- **DataSet:** The DataSet is a disconnected RecordSet that can be browsed in both forward and backward directions. We can also update the data using the dataset. DataSet is filled by using DataAdapter.
- **DataView Class:** A DataView allows you to create various views of data from DataTable, which can be used for data-binding applications. Using this, you can display the table with different order of sorting or you can filter the data based on a filter expression or by row state, etc.
- **XML:** It is possible to create an XML representation of a dataset. In the dataset's XML representation, data is represented in XML format and the database schema is represented in XML Schema Definition(XSD) language.

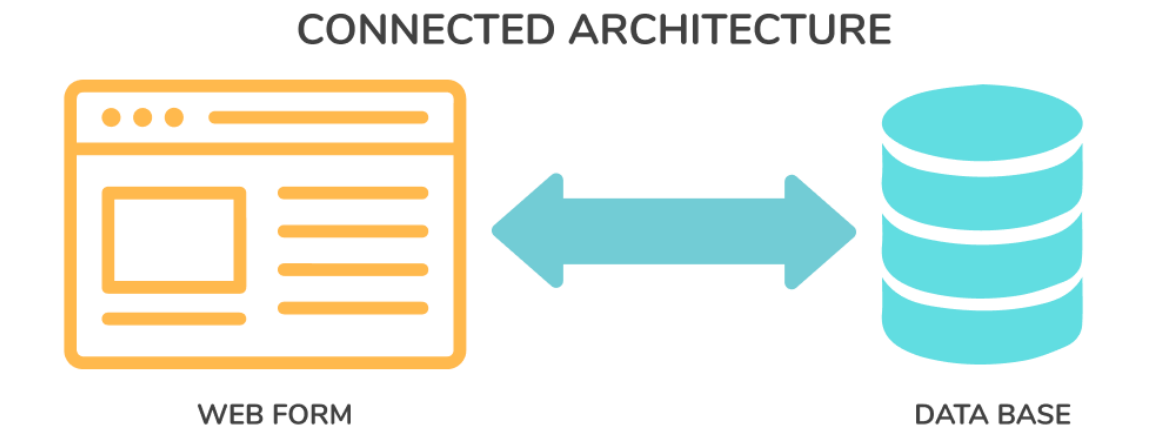


InterviewBit

22. Briefly explain connected and disconnected architecture of ADO.NET.

Connected Architecture:

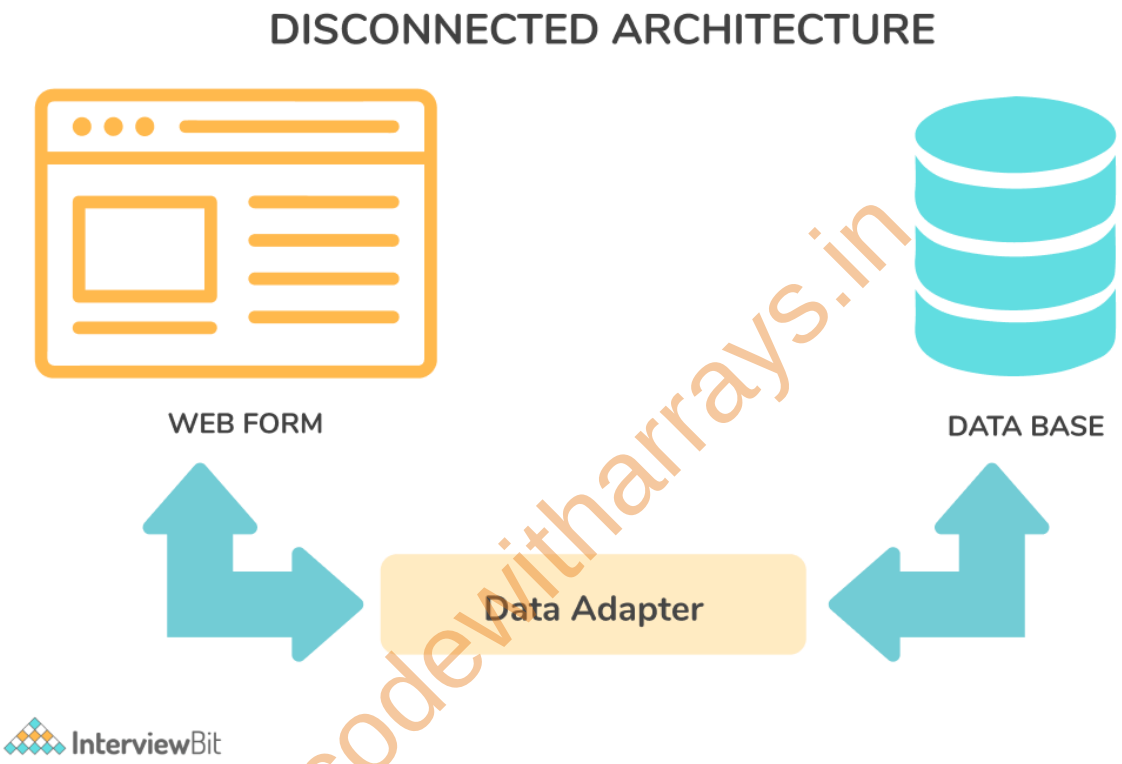
- In connected architecture, the connection must be kept open for accessing the data retrieved from the database. Connected architecture is based on Connection, DataReader, Command, and Transaction classes.
- You constantly visit the database for any CRUD (Create, Read, Update, and Delete) operation you want to do. This will create high traffic to the database, but this is usually faster as you are doing only smaller transactions.
- DataReader can be said as a Connected Architecture as it holds the connection open until it fetches all the rows one by one.



InterviewBit

Disconnected Architecture:

- In disconnected architecture, even if the database connection is closed, data retrieved from the database can be accessed. Disconnected architecture is based on classes connection, CommandBuilder, DataAdapter, DataSet, and DataView.
- Here, we retrieve and store a recordset from the database so that you can perform many CRUD (Create, Read, Update, and Delete) operations on the data within memory, it will be re-synchronized when you reconnect with the database.
- DataSet is a Disconnected Architecture because all records are brought at once and holding the database connection alive is not necessary.



23. Explain about ExecuteScalar() in ADO.NET.

- A single value from the first row and first column of the ResultSet will be returned by ExecuteScalar() method on query execution.
- If the ResultSet is having multiple rows or columns, all those rows and columns will be ignored except the first row and first column. If the ResultSet is empty, this function will return NULL.
- The best situation to use ExecuteScalar() method is when we are using functions such as COUNT(), SUM(), etc., as it uses only a few resources compared to the ExecuteReader() method.
- Example:

```
public void ExecuteScalarExample()
{
    SqlConnection con = new SqlConnection();
    con.ConnectionString =
ConfigurationManager.ConnectionStrings["conString"].ConnectionString;
    try
    {
```

```
        SqlCommand cd = new SqlCommand();
        cd.Connection = con;
        cd.CommandText = "SELECT SUM(SALARY) FROM EMPLOYEE";
        cd.CommandType = CommandType.Text;
        con.Open();
        Int32 SalaryTotal = Convert.ToInt32(cd.ExecuteScalar());
        MessageBox.Show("Total Salary of the employee is : " +
SalaryTotal.ToString());
        cd.Dispose();
        con.Dispose();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Here, we create an object of the class SqlConnection and SqlCommand. We pass SQL Statement to the object of SqlCommand class, which returns a single value. When ExecuteScalar() function gets executed, a single value will be returned, i.e, the total salary of employees. This value will be displayed using a message box.

24. Explain about ADO.NET objects.

There are seven main objects in ADO.NET. They are:

1. **DataSet:** It is available under both `System.Data.ADO` and the `System.Data.SQL` namespaces. DataSet is a database cache built-in memory for using it in disconnected operations. It holds the complete collection of tables, constraints, and relationships.
2. **SQLDataSetCommand:** It represents a stored procedure or a database query that can be used to populate the DataSet object. It corresponds to the ADO's Command object-provided functionalities.
3. **SQLCommand:** It represents a stored procedure or a T-SQL statement that will be executed by SQL Server. It corresponds to another set of functionalities provided by the ADO's Command object.
4. **SQLParameter:** It can be used to pass parameters to the object of SQLCommand or SQLDataSetCommand class. When you are passing a parameter for SQLCommand using SQLParameter, SQLParameter will represent a parameter that can be used by T-SQL statement or stored procedure. Whenever a parameter has been passed for SQLDataSetCommand using SQLParameter, SQLParameter will represent a column from a result set.
5. **SQLConnection:** It represents an open connection to the data source like SQL Server. This object is similar to the standard Connection object in ADO.
6. **SQLDataReader:** It reads a forward-only stream of data from a SQL Server database. It works with an open database connection.
7. **SQLError:** It collects runtime warnings and error conditions related information that will be encountered by an ADO.NET application. It corresponds to ADO's Error object.

25. What are the different authentication techniques used to connect with MS SQL Server?

Before performing any task in the database, SQL Server will authenticate. Two types of authentication techniques are:

- **Windows Authentication:** This default authentication is provided only through Windows domain accounts. This SQL Server security model is strongly integrated with Windows, so it is also referred to as integrated security. Particular Windows users and group accounts are allowed to login into SQL Server. Windows users who are already been authenticated or logged onto Windows do not have to provide additional credentials.
  The below-given `SqlConnection.ConnectionString` specifies Windows authentication without any need of providing a user name or password by the user.

```
C#
"Server=MSSQL1;Database=Institute;Integrated Security=true;
```

- **SQL Server and Windows Authentication Mode(Mixed-mode):** Authentication will be provided with the help of the Windows and SQL Server Authentication combination. User name and password pair will be maintained within SQL Server. In order to use this mixed-mode authentication, you need to create SQL Server logins that are stored in SQL Server. After that, you can supply the user name and password to SQL Server at run time.

The below-given ConnectionString specifies Mixed mode authentication:

```
C#
"Persist Security Info=False;User ID=Harsh;Password=xyz@123;Initial
Catalog=Institute;Server=MySqlServer"
```

26. What is Response.Expires and Response.ExpiresAbsolute property?

- `Response.Expires` property is specific to the minutes that a particular page stays in the cache for the specific time from the time it has been requested. For example, if `Response.Expires` value is set to 5 minutes, then the page is instructed to be in cache for 5 minutes from the time it has been requested.
- `Response.ExpiresAbsolute` property helps to provide the proper time at which a specific page cache has been expired. For example, `Response.ExpiresAbsolute` provides information like 14 March 15:40:15. This time tells about when the page was in cache.

27. How to load multiple tables into a dataset?

```
DataSet ds=new DataSet();
SqlConnection con=new SqlConnection("connection_string");
SqlDataAdapter da=new SqlDataAdapter("select * from Employee1",con);
da.Fill(ds.Tables.Add());
da=new SqlDataAdapter("select * from Employee2",con);
da.Fill(ds.Tables.Add());
```

After tables have been added into a DataSet, the below-given code tells about how to make use of the DataSet tables. If you decide to use the first table in a dataset or to copy the table data into a data table, then follow the below-given code:

```
DataTable dt=new DataTable();
dt=ds.Tables[0];
```

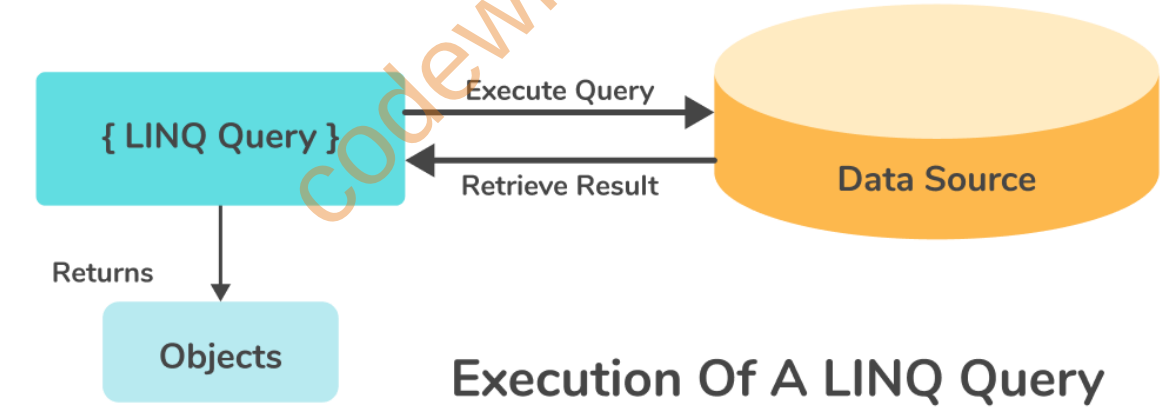
The above code can be used to add the required number of tables in a dataset. This ensures connection-less access to data. As the dataset is filled with multiple tables, every time we want to query the data the database connection is not required. It also makes sure about the reusability of data.

28. What is the difference between connected and disconnected architecture in ADO.NET?

Connected architecture	Disconnected architecture
It is connection-oriented.	It is not connection-oriented.
DataReader is a connected architecture.	DataSet is a disconnected architecture.
High speed and performance are given by connected methods.	Disconnected methods are low in speed and performance.
Data persistence is not possible using DataReader.	Data persistence is possible using DataSet.
It carries the single table data.	It carries data from multiple tables.
We can't update the data as it is read-only.	Here we can update the data.

29. What is LINQ?

- LINQ(Language Integrated Query) is a structured query syntax that helps the programmers and testers to retrieve data from various data sources such as Collections, XML Docs, ADO.NET DataSet, web service, MS SQL Server, etc.
- It is integrated with C# or VB.NET and it eliminates the mismatch between different programming languages and databases. It provides a single querying interface for various data source types.
- An object will be returned as a result of LINQ query execution. It will allow you to use an object-oriented approach on the result set and there is no need to worry about the transformation of different result formats into objects.



30. How can you identify whether any changes are made to the DataSet object since the time it was last loaded?

The DataSet object has two methods to track down the changes:

- **GetChanges():** It returns the DataSet object that has been changed since it was loaded or since the execution of the AcceptChanges() method.
- **HasChanges():** It indicates if any modifications were made since from the time the DataSet object was loaded or after a method call to the AcceptChanges() was made.

Use the RejectChanges() method, if you want to reverse the entire changes since from the time the DataSet object was loaded.

31. What is the difference between Dataset.Clone() and DataSet.Copy() methods?

- The method **Clone()** copies only the DataSet structure. The copied structure will have all the constraints, relations, as well as DataTable schemas used by the DataSet. It does not copy the data stored in the DataSet.
- The **Copy()** method copies the DataSet structure along with the data in the DataSet. The original data will not be affected.

32. Which methods are provided to add or remove rows from the DataTable object?

The collection of rows for the DataTable object has been defined by the DataRowCollection class. DataRowCollection class has the method NewRow() for adding a new DataRow to DataTable. This method creates a new row that implements the similar schema that is applied to the DataTable.

The methods provided by the DataRowCollection object are given below:

- **Add()**- It adds a newly created row into DataRowCollection.
- **Remove()**- It deletes the object DataRow from DataRowCollection.
- **RemoveAt()**- It deletes a row for which location is marked by an index number.

33. How to make SQL Server connection in ADO.NET?

Consider the below example where a connection to the SQL Server has been established. An employee database will be used to connect. The C# code will be:

```
using (SqlConnection con = new SqlConnection(connectionString))
{
  con.Open();
}
```


Using block will be useful in closing the connection automatically. It is not required to explicitly call the close() method, because using block will do this implicitly when the code exits the block.

```
// ConnectionExample.cs

using System;
using System.Data.SqlClient;
namespace ConsoleApplicationExample
{
    class ConnectionExample
    {
        static void Main(string[] args)
        {
            new Program().ConnectingMethod();
        }
        public void ConnectingMethod()
        {
            using (
```

```
                    // Creating Connection
                    SqlConnection conn = new SqlConnection("data source=.;
database=employee; integrated security=SSPI")
                 )
            {
                conn.Open();
                Console.WriteLine("Connection Has Been Successfully
Established.");
            }
        }
    }
}
```

Output:

```
Connection Has Been Successfully Established.
Press any key to continue...
```

On execution, if the connection has been established, a message will be displayed on an output window.

If the connection is not created with the help of using a block, a connection must be closed explicitly.

34. What is serialization? Write an example program to serialize a DataSet.

Serialization is the method of converting an object into a byte stream which can be stored as well as transmitted over the network. The advantage of serialization is that data can be transmitted in a cross-platform environment across the network and also it can be saved in a storage medium like persistent or non-persistent.

The code for serializing a DataSet is:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Xml.Serialization;
using System.IO;
public partial class Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection("Data
Source=data_source_name;Initial Catalog=employee;Integrated
Security=True");  //Create connection object
        SqlDataAdapter da = new SqlDataAdapter("select * from emp", conn);
//DataAdapter creation
        DataSet s = new DataSet();
        da.Fill(s);
        FileStream fObj = new FileStream("C:\\demo.xml", FileMode.Create);
// Create a XML file
        XmlSerializer sObj = new XmlSerializer(typeof(DataSet));
        sObj.Serialize(fObj, s);  //Serialization of a DataSet
        fObj.Close();
    }
}
```

In the above given example, the database name is employee and, the table name is emp. The data in a DataSet will be serialized and stored in a demo.xml file by using Serialize() method.

35. Give an example code to fill the GridView by using the object of DataTable during runtime.

```
using System;
using System.Data;

public partial class Default : System.Web.UI.Page
{
     protected void Page_Load(object sender, EventArgs e)
     {
          GridView gridView1=new GridView();       //Create GridView object
          DataTable t = new DataTable("Employee"); // Create the table
object
          DataColumn c = new DataColumn();        //Creating table column
          DataRow r;                              //Instance of row
          c.ColumnName = "EmpID";                 //Heading of the coloumn
          c.DataType = Type.GetType("System.Int32"); //Set the data type of
EmpID as an Integer
          t.Columns.Add(c);                      //Adding a column to data
table
          c = new DataColumn();
          c.ColumnName = "EmpName";
          c.DataType = Type.GetType("System.String"); //Set the type of
EmpName as String
          t.Columns.Add(c);
          for (int i = 0; i < 5; i++)          //This code will create 5
rows
          {
               r = t.NewRow();
               //Add Column values
               r["EmpID"] = i;
               r["EmpName"] = "Employee " + i;
               t.Rows.Add(r);
          }
          gridView1.DataSource = t;  //Set gridView1 Datasource as
DataTable t
          gridView1.DataBind();         //Bind Datasource to gridview
     }
}
```


Output:

EmpID	EmpName
0	Employee 0
1	Employee 1
2	Employee 2
3	Employee 3
4	Employee 4

Conclusion

ADO.NET is a brilliant technology that was developed by Microsoft on the framework of .NET. The primary role and responsibility of ADO.NET technology is to setup a bridge between backend language and your database. A good experience of this technology will be of great use from a development point of view.

ADO.NET technology will definitely help in your career growth as it has quite a good scope. Also learning this interesting technology will always be great fun. ADO.NET along with the knowledge of databases will definitely be exceptional from a growth perspective.

→4.ADO.NET MCQ

1.

__ class acts as a bridge between a byte stream and a character stream?

- InputStreamReader
- OutputStreamReader
- ByteStreamReader
- None of the above

2.

Which ADO.NET object reads data from the data store in forward-only mode?

- DataAdapter object
- OutputStreamReader object
- DataReader object
- SqlDataReader object

3.

Which one of the following is retrieved by the data adapter in disconnected access?

- ResultSet
- DataSet

- Both ResultSet and DataSet
- None of the above

4.

What does ADO.NET stand for?

- Access Data Object for .NET framework
- Address Data Object
- Active Dataset Object
- ActiveX Data Object for .NET framework

5.

Which object of ADO.NET has the best performance?

- DataSet object
- DataProvider object
- DataReader object
- DataAdapter object

6.

Which object is very faster in obtaining data from the database?

- SqlDataReader object
- DataReader object
- DataSet object
- ResultSet object

7.

We can create and process in-memory databases using _____.

- Record
- Table

- View
- DataSet

8.

Is ADO.NET a framework?

- No, it is a database
- No, it is a library within .NET framework
- Yes
- None of the above

9.

DataTable is _____.

- Member of a System.Data
- Member of a DataSet
- Member of a System.Data.SqlClient
- Member of a DataAdapter

10.

The DataReader object is considered as an alternative to _____.

- DataAdapter
- DataSet
- Both a and b
- None of the above

11.

Which of the following is not an ADO.NET DataAdapter object?

- QueryDataAdapter
- OleDbDataAdapter

- SQLDataAdapter
- All of the above

12.

__ database is designed for the ADO.NET SqlConnection object.

- Microsoft SQL Server
- Access
- MySQL
- Oracle

13.

What represents the name of the database in a connection string?

- Data source
- Database
- Initial catalog
- Catalog initial

14.

Which method is used to sort the data in the ADO.NET?

- Sort() method of GridViewControl
- Sort() method of ListViewControl
- Sort() method of DataViewControl
- ToSort() method of GridViewControl

15.

_____ class of ADO.NET provides a disconnected environment.

- Command
- DataReader

- Data
- DataSet

16.

How will you execute a stored procedure in the database?

- Using executeProcedure() method on a Statement object
- Using execute() method on a StoredProcedure object
- Using run() method on a ProcedureCommand object
- Using execute() method on a CallableStatement object

→5.WEB API INTERVIEW QUESTIONS:-

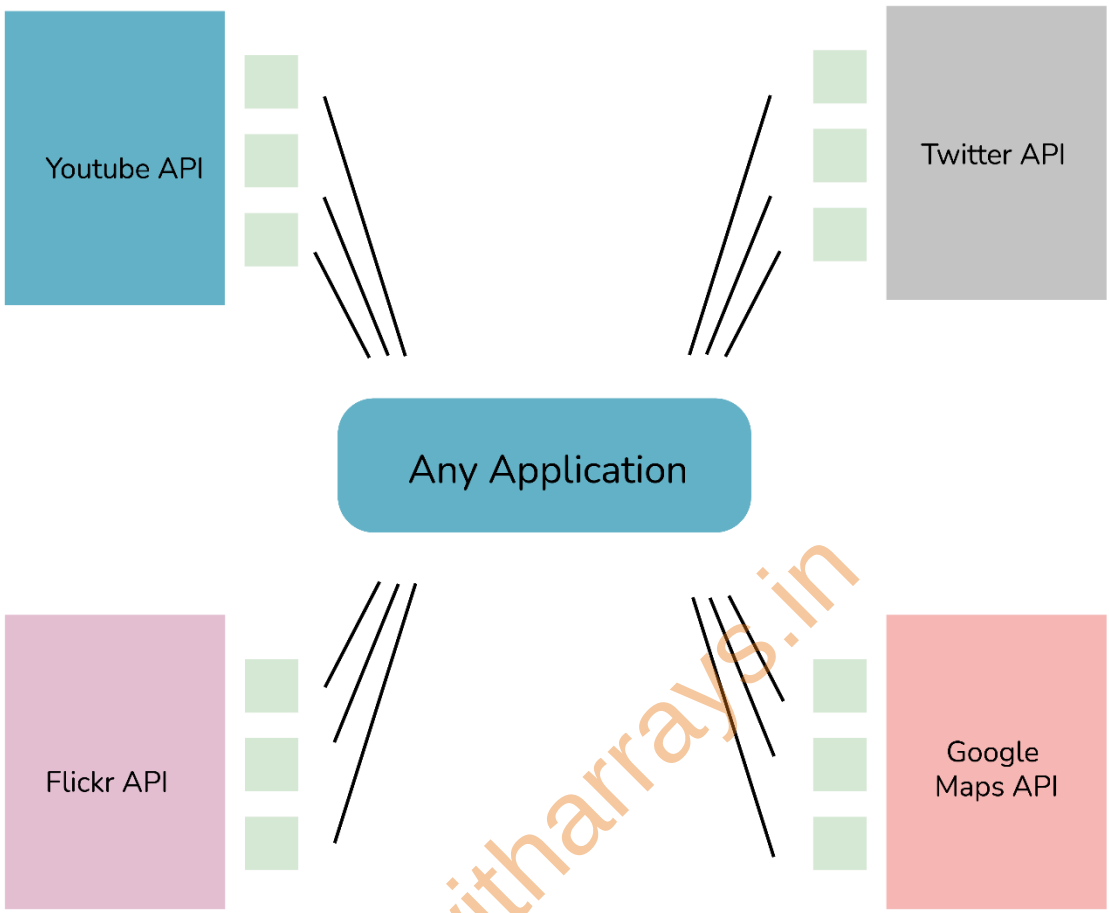
→REFERENCE→ INTERVIEW BIT:-

→5Web API Basic Interview Questions

1. Why is the Web API important?

Web API is generally considered as a service that basically provides us information or data from the server. It is very important because of the following reasons:

- It is used to provide an interface for websites and client applications to have access to data.
- It can also be used to access data from the database and save data back to the database.
- It supports different text formats such as XML, JSON, etc.
- It is suitable or compatible with any type of browser and any type of device like mobile, desktop, web, etc.
- It uses low bandwidth such as XML or JSON data, etc., and is therefore considered good for devices that have limited bandwidth such as smartphones, etc.
- From a business point of view, web API is more applicable for UI/UX, increases interest in the company's product and services, increases website traffic.

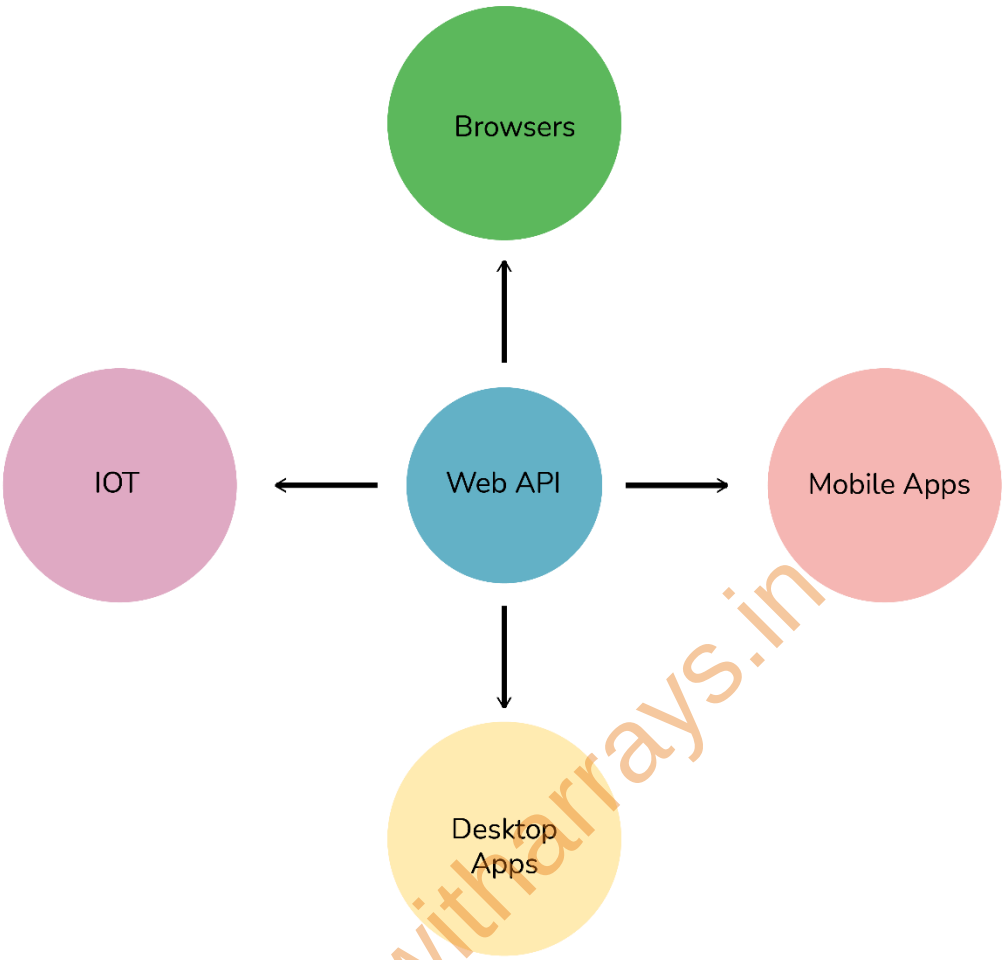


API Providding Services to Application

InterviewBit

2. What is Web API and why we use it ?

Web API (Application Programming Interface), as the name suggests, is an API that can be accessed over the Web using the HTTP protocol. It is basically considered the best platform for revealing or uncovering data and services to various different services. It is a tool that can be used to push data to a server and can be accessed by server code. It can be built or developed using various technologies like java, ASP.NET, etc.



Web API Uses:

- It contains additional layers that simply standardize communications and provide different options on how to format input and output.
- It can be used with ASP.NET MVC and different types of web applications such as ASP.NET WebForms.
- If one wants to create resource-oriented services, then Web API services are considered the best.
- It also helps to develop REST-ful services and SOAP-based services.

3. What are the main return types supported in Web API?

It does not have any specific data type. It can return data of any type depending upon the business requirement. There are many HTTP methods like GET, POST, PUT, etc., which can return data in different formats depending upon the use case.

4. What is the difference between Web API and WCF?

WCF (Windows Communication Foundation): It is a framework used for developing SOAP (Service-oriented applications). This framework is used for developing, configuring, and deploying, or implementing network-distributed services.

Web API: It is an application programming interface for both web browsers and web servers. Browser API simply extends or increases the functionality of web browsers whereas Server API simply extends or increases the functionality of web server.

Web API	WCF
It is used to develop both SOAP-based services and RESTful services.	It is used to deploy only SOAP-based services.
It supports various MVC features such as routing, model binding, etc.	It does not support any MVC features.
It only supports HTTP protocol.	It supports various protocols such as HTTP, UDP, custom transport.
It is considered best for developing RESTFUL services.	It supports only limited RESTFUL services.
It is good when one wants to expose an expensive range of clients such as iPhones, browsers, mobile phones, tablets, etc.	It is good for creating services that uses expedite transport channels such as TCP, UDP, Named pipes, etc.
It offers support for UTF-8 encoding format.	It offers TEXT, Binary encoding support, MTOM (Message Transmission Optimization Mechanism), etc.

5. Why to choose Web API over WCF?

Web API is considered the best choice over WCF because of the following reasons:

- Web API uses all features of HTTP such as URIs, request/response headers, caching, versioning, various content formats, etc.
- One does not have to define or explain any extra config setting for different devices in Web API.
- Web API uses different text formats including XML because of which it is faster and more preferred for lightweight services.
- Web API also supports MVC features whereas WCF does not support MVC features.
- Web API provides more flexibility as compared to WCF.
- Web API uses standard security like token authentication, basic authentication, etc., to provide secure service whereas WCF uses WS-I standard to provide secure service.

6. What is different between REST API and RESTful API?

REST (Representation State Transfer) **API**: It is basically an architectural style that makes productive use of existing technology and protocols of the web. It is a set of rules that developers need to follow when they develop their API or services that are scalable. It is used with HTTP protocol using its verbs such as GET, DELETE, POST, PUT.
RESTful API: It is simply referred to as web services executing such as architecture.

REST API	RESTful API
REST is an architectural pattern used for creating web services.	RESTful API is used to implement that pattern.
The data format of REST is based on HTTP.	The data format of RESTful is based on JSON, HTTP, and Text.
Working of URL is based on request and response.	Working of RESTful is based on REST applications.
It is more user-friendly and highly adaptable to all business enterprises and IT.	It is too flexible.
It is required to develop APIs that allow interaction among clients and servers.	It simply follows REST infrastructure that provides interoperability among different systems on the whole network.

7. What are the advantages of using Rest in Web API?

REST is very important and beneficial in Web API because of the following reasons:

- It allows less data transfer between client and server.
- It is easy to use and lightweight.
- It provides more flexibility.
- It also handles and controls various types of calls, returning various data formats.
- It is considered best for using it in mobile apps because it makes less data transfer between client and server.
- It uses simple HTTP calls for inter-machine communication rather than using more complex options like CORBA, COM+, SOAP, or RPC.

8. What is REST and SOAP? What is different between them?

REST (Representational State Transfer): It is a new and improved form of web service. It describes the architectural style of networked systems. It does not require greater bandwidth when requests are sent to the server. It just includes JSON message. For example:

```
{"city":"Mumbai","state":"Maharashtra"}
```

SOAP (Simple Object Access Protocol): It is a simple and lightweight protocol that is generally used for exchanging structured and typed information on the Web. It works mostly

with HTTP and RPC (Remote Procedure Call). This protocol is mainly used for B2B applications one can define a data contract with it. SOAP messages are heavier in content and therefore use greater bandwidth.

For example:

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope"
SOAP-ENV:encodingStyle=" http://www.w3.org/2001/12/soap-encoding">
<soap:Body>
<Demo.guru99WebService xmlns="http://tempuri.org/">
<EmployeeID>int</EmployeeID>
</Demo.guru99WebService>
</soap:Body>
</SOAP-ENV:Envelope>
```

REST	SOAP
It is basically an architectural pattern.	It is basically a messaging protocol.
It usually works with various text formats such as plain text, HTML, JSON, XML, etc.	It only works with XML formats.
It is totally stateless.	It has some specifications for both stateless and stateful implementation.
Its performance is faster as compared to SOAP.	Its performance is slower as compared to REST.
It uses XML and JSON to send and receive data.	It uses WSDL (Web Service Description Language) for communication among consumers or users and providers.
REST has to resend transfer whenever it determines any errors.	SOAP includes built-in error handling for communications errors using WS-ReliableMessaging specification.
It calls services using the URL path.	It calls services by calling RPC (Remote Procedure Call) method.

9. What is Web API 2.0?

It is basically an enhanced and modified feature of Web API. This new version supports various new features as given below:

- New Routing Attribute
- Secure ASP.NET Web API using OAuth 2.0
- Support for Cross-Origin requests using CORS
- IHttpActionResult return type
- Support for `$`expand, `$`select in OData Service

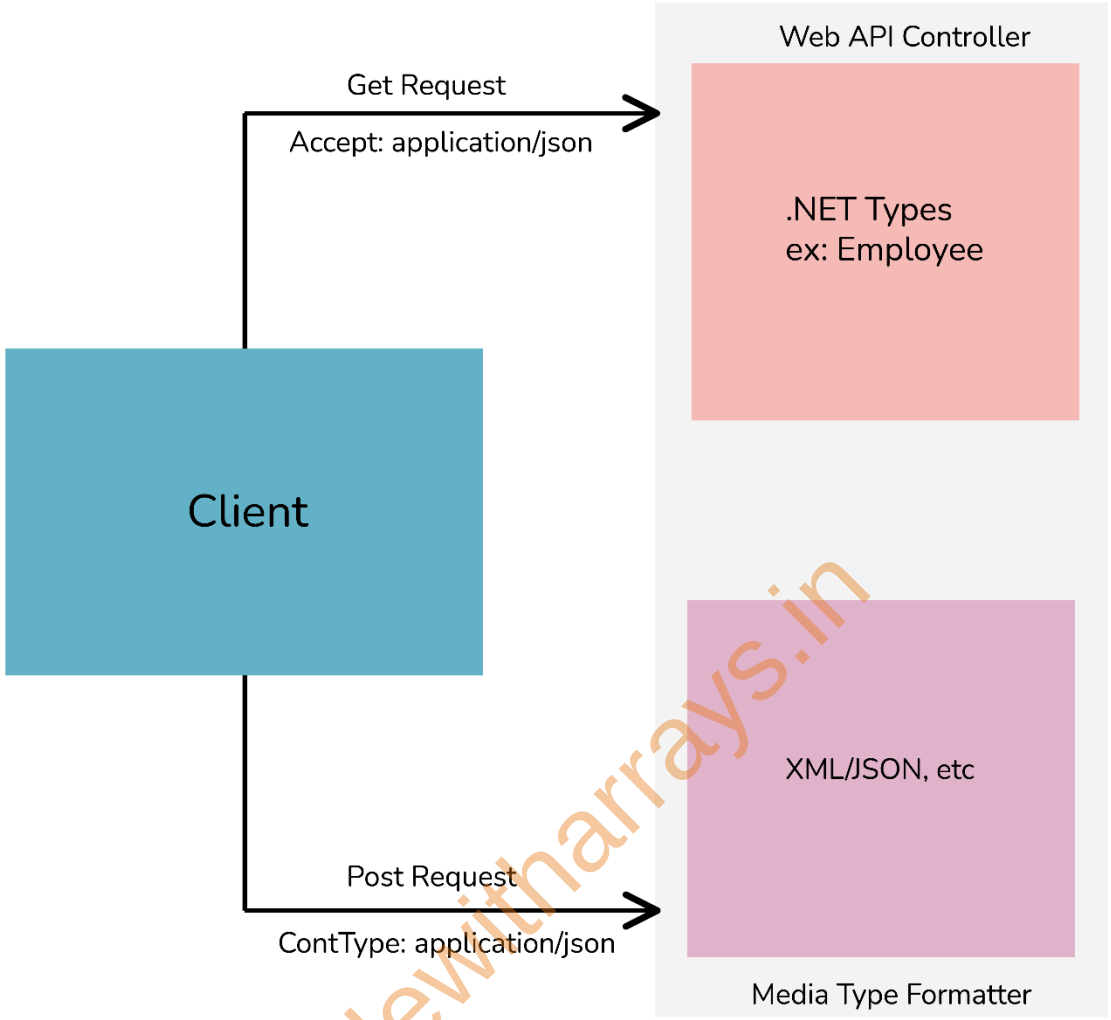
Because of all the new features of Web API 2.0, it is considered an optimal choice and suitable development model that makes it easier to develop RESTful services interfaces to

different clients running on various platforms. It also supports configuring routes in the Web API method or controller level.

10. Explain media type formatters.

In web API, media type formatters are classes that are responsible for serialization data. Here, serialization generally means a process of translating data into a format that can be transmitted and reconstructed later. Because of serializing request/response data, Web API can understand request data format in a better way and send data in a format that the client expects. It simply specifies data that is being transferred among client and server in HTTP response or request.

Media Type Formatter Class	MIME Type	Description
JsonMediaTypeFormatter	application/json, text/json	Handles JSON format
XmlMediaTypeFormatter	application/xml, text/json	Handles XML format
FormUrlEncodedMediaTypeFormatter	application/x-www-form-urlencoded	Handles HTM form URL-encoded data
JQueryMvcFormUrlEncodedFormatter	application/x-www-form-urlencoded	Handles model-bound HTML form URL-encoded data



11. Web API supports which protocol?

Web API generally supports only HTTP protocol.

12. Which of the following Open-source libraries is used by WEB API for JSON serialization?

Json.NET library is generally used by Web API for JSON serialization.

13. What is XML and JSON?

XML (Extensible Markup Language):

- It is especially designed to store and transport data.
- It is similar to HTML but is more flexible than HTML because it allows users to create their own custom tags.

- It is used for representing structured information such as documents, data, configuration, etc.

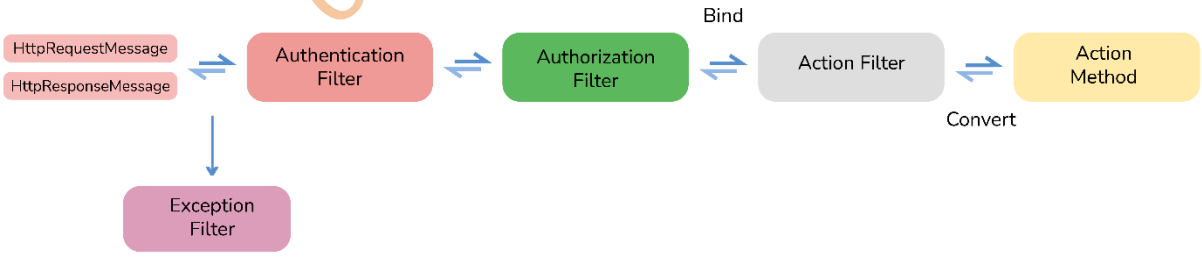
JSON (JavaScript Object Notation):

- It is a lightweight format designed to store and transport data.
- It is easier to understand and is a standard text-based format used for representing structured data based on JavaScript object syntax.
- It is faster and easier to use.

14. What are Web API filters?

Filters are basically used to add extra logic at different levels of Web API framework request processing. Different types of Web API filters are available as given below:

- **Authentication Filter:** It handles authentication and authenticates HTTP requests. It also helps to authenticate user detail. It checks the identity of the user.
- **Authorization Filter:** It handles authorization. It runs before controller action. This filter is used to check whether or not a user is authenticated. If the user is not authenticated, then it returns an HTTP status code 401 without invoking the action.
- **AuthorizeAttribute** is a built-in authorization filter provided by Web API.
- **Action Filter**: It is attributing that one can apply to controller action or entire controller. It is used to add extra logic before or after controller action executes. It is simply a way to add extra functionality to Web API services.
- **Exception Filter**: It is used to handle exceptions that are unhandled in Web API. It is used whenever controller actions throw an unhandled exception that is not HttpResponseException. It will implement an "IExceptionFilter" interface.
- **Override Filter**: It is used to exclude specific action methods or controllers from the global filter or controller level filter. It is simply used to modify the behavior of other filters for individual action methods.



ASP.NET Web API Filter Pipeline

InterviewBit

15. Who can consume Web API?

A large range of clients such as browsers, mobile devices, iPhone, etc., include or consume web API. It is also good for using along native applications that require web services but not SOAP support. It can also be consumed by any client that supports HTTP verbs such as GET, DELETE, POST, PUT.

16. How to handle errors in Web API?

Web API generally provides greater flexibility in terms of handling errors. Exception handling is a technique that is used to handle run-time errors in application code. One can use HttpResponseException, HttpError, Exception filters, register exception filters, Exception handlers to handle errors. Exception filter can be used to identify unhandled exceptions on actions or controllers, exception handlers can be used to identify any type of unhandled exception application-wide, and HttpResponseException can be used when there is the possibility of an exception.

17. How to register an exception filter globally?

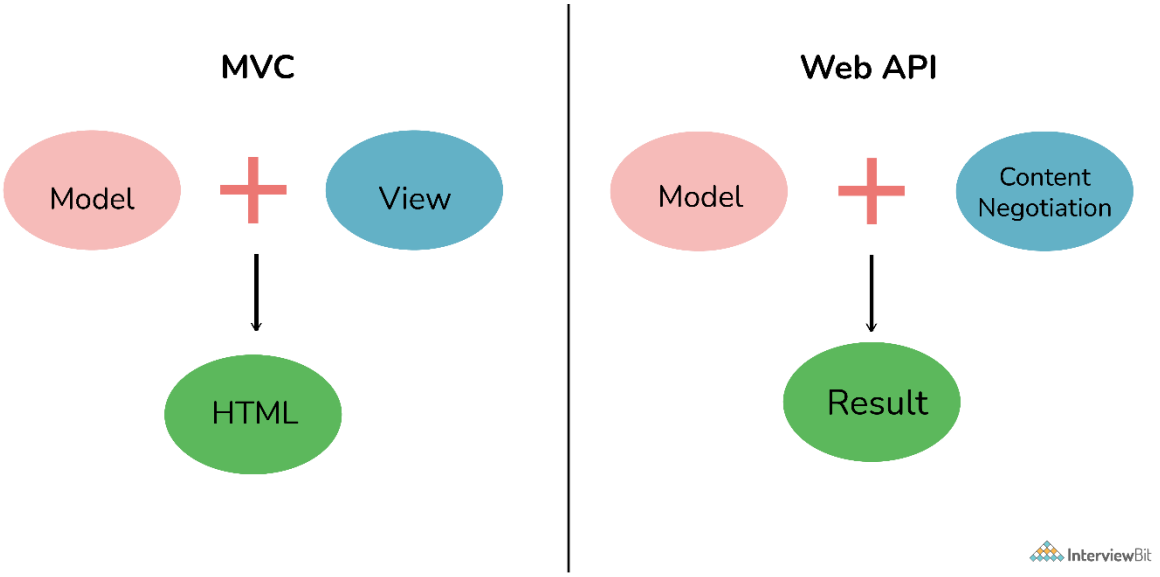
One can register exception filter globally using following code:

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

18. What is MVC? Write difference between MVC and Web API?

MVC (Model, View, and Controller) is basically an application design model that comprises three interconnect parts I.e., model, view, and controller. It allows coders to factor out different components of the application and update them more easily. It is mostly used for developing model user interfaces. Its main purpose is to display patterns in structure for keeping display and data separate to enable both of them to change without affecting others.

MVC	Web API
It can be used to build Web applications that reply as both data and views.	It is used to build HTTP services that reply only as data.
It returns data in JSON format by using JSONResult.	It returns data in different formats such as JSON, XML, etc.
It supports content negotiation, self-hosting.	It does not support content negotiation, self-hosting.
It is not able to build REST-full services.	It is very helpful in creating REST-full services.
It returns a view (HTML).	It returns REST responses.



ASP.NET Web API Interview Questions

19. What is ASP.NET Web API?

ASP stands for Active server pages. ASP.NET is an updated version of legacy ASP. It is a framework that is used for developing HTTP services to provide responses to client requests. It can be accessed in different applications on different platforms. It is provided by Microsoft open-source technology for developing and consuming HTTP-based services on top of .NET Framework. It is very easy to build HTTP services using ASP.NET Web API. These services can be used by different clients as given below:

- Desktop Applications
- Mobile Applications
- IOTs
- Browsers

20. What are the advantages of using ASP.NET Web API?

Some of the core advantages of using ASP.NET Web API are given below:

- It provides the best platform for developing RESTful applications on .NET Framework.
- It works the same way that HTTP works with help of HTTP verbs such as GET, POST, PUT, DELETE for all crud operations.
- It provides enough flexibility in Web API creation.
- It completely supports routing.
- It also supports model binding, validation, Odata (Open Data Protocol) that allows creation and consumption of RESTful APIs.
- It has the ability to develop custom help and test pages with help of ApiExplorer.
- One can develop non-SOAP-based services such as plain XML, JSON strings, etc.
- It also increases the TDD (Test Data-Driven) approach in the development of RESTful services.

21. What are new features used in ASP.NET Web API 2.0

ASP.NET Web API includes a number of new exciting features as given below:

- Attribute Routing
- CORS (Cross-Origin Resource Sharing)
- OWIN (Open Web Interface for .NET) self-hosting
- IHttpActionResult
- Web API OData

22. What is the use of HttpResponseMessage?

It is used to set response values such as header and status control. It simply allows us to work with HTTP protocol. It represents HTTP response messages that encapsulate data and status code.

```
// GetEmployee action
public HttpResponseMessage GetEmployee(int id)
{
     Employee emp = EmployeeContext.Employees.Where(e => e.Id ==
id).FirstOrDefault();
     if (emp != null)
      {
               return Request.CreateResponse<Employee>(HttpStatusCode.OK,
emp);
      }     else
      {
               return Request.CreateErrorResponse(HttpStatusCode.NotFound,
"Employee Not Found");
      }
}
```


23. What is the difference between ApiController and Controller?

ApiController: It is used to return data that is arranged in series and then sent to the client.

```
public class TweetsController : ApiController
{
         // GET: /Api/Tweets/
        public List<Tweet> Get()
      {
          return Twitter.GetTweets();
       }
}
```

Controller: It is used to provide normal views.

```
public class TweetsController : Controller
{
        // GET: /Tweets/  [HttpGet]  public ActionResult Index()
       {
           return Json(Twitter.GetTweets(), JsonRequestBehavior.AllowGet);
       }
}
```

24. What do you mean by Caching and What are its types?

Caching is basically a technique or process of storing data somewhere or in the cache for future requests. The cache is a temporary storage area. Caching keeps all frequently or recently accessed files or data in the cache memory and accesses them from the cache itself rather than actual address of data or files. The cache interface simply improves the storage mechanism for request/response object pairs that are being cached.

Advantages of Caching:

- It is considered the best solution to ensure that data is served where it is needed to be served that too at a high level of efficiency which is best for both client and server.
- It delivers web objects faster to the end-user.
- It reduces load time on the website server.
- It leads to faster execution of any process.
- It decreases network costs.

Types of Caching:
There are basically three types of caching as given below:

- Page Caching
- Data Caching
- Fragment Caching

25. WCF is replaced by ASP.NET Web API. True/False?

WCF: It is a framework used for developing SOAP (Service Oriented Applications Protocols). It also supports various transport protocols as given above.
ASP.NET Web API: It is a framework used for developing non-SOAP-based services. It is limited to HTTP-based services.

No, it's not true that ASP.NET Web API has replaced WCF. WCF was generally developed to develop SOAP-based services. ASP.NET Web API is a new way to develop non-SOAP-based services such as XML, JSON, etc. WCF is still considered a better choice if one has their service using HTTP as the transport and they want to move to some other transport like TCP, NetTCP, MSMQ, etc. WCF also allows one-way communication or duplex communication.

26. What are the main return types supported in ASP. Net Web API?

It supports the following return types:

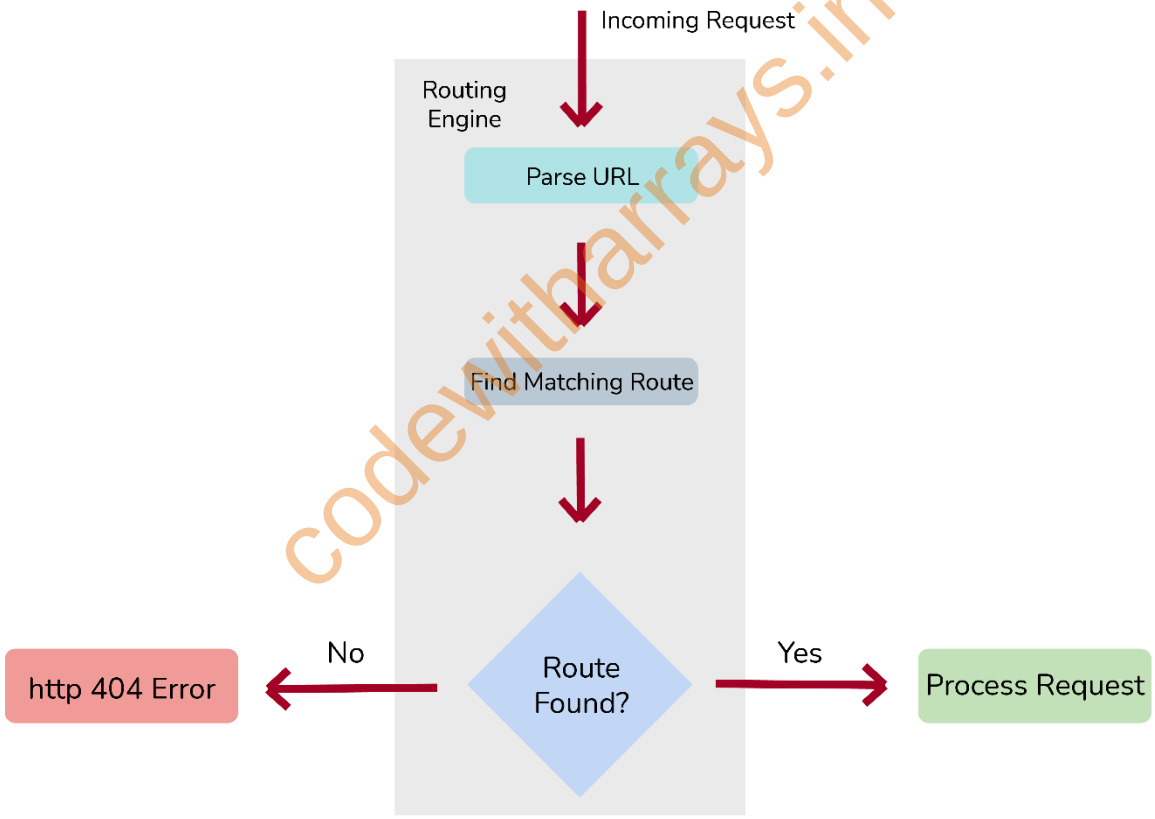
- HttpResponseMessage
- IHttpActionResult
- Void
- Other types such as string, int, etc.

27. What is ASP.NET Web API routing?

Routing is the most important part of ASP.NET Web API. Routing is a way how Web API matches a URI to an action. It is basically a process that decides which action and controller should be called. The controller is basically a class that handles all HTTP requests. All public methods of controllers are basically known as action methods or just actions. Whenever a Web API framework receives any type of request, it routes that request to action.

There are basically two ways to implement routing in Web API as given below:
Convention-based routing: Web API supports convention-based routing. In this type of routing, Web API uses route templates to select which controller and action method to execute.

Attribute-based routing: Web API 2 generally supports a new type of routing known as attribute routing. As the name suggests, it uses attributes to define routes. It is the ability to add routes to the route table via attributes.



How Routing Works?

InterviewBit

28. HOW to secure ASP.NET Web API?

Web API has become key to programming web-based interactions. It can be accessed by anyone who knows the URL. Therefore, they have become targets for hackers. One needs to

secure Web API by controlling Web API and by deciding who can and who cannot have access to Web API. There are basically two ways or techniques that make our Web API more secure.

Authentication: It is a process that helps to identify and check users by their credentials such as password, username, etc. To have access to the web API, firstly user credentials are needed to be passed in the request header. If user credentials are not passed into the request header, then the server returns 401 status code (unauthorized). The best authentication to be used is OAuth 2.0.

Authorization: It is a process that helps to decide whether or not a user has access to perform an action. Authorization filters are used to implement authorization.

29. What are Exception filters in ASP.NET Web API?

Exception filter is generally used to handle all unhandled exceptions that are generated in web API. It implements IExceptionFilters interface. It is the easiest and most flexible to implement. This filter is executed whenever the controller method throws any unhandled exception at any stage that is not an HttpResponseExecption exception.

30. Which .NET framework supports ASP.NET Web API?

.NET Framework 4.0 generally supports the first version of ASP.NET Web API. After that, .NET Framework 4.5 supports the latest version of web API i.e., ASP.NET Web API 2.

31. What is HttpConfiguration in Web API?

It is considered as the main class that includes different properties with help of which one can override the default behavior of Web API.

Some properties are given below:

- **DependencyResolver:** It sets or gets a dependency resolver for dependency injection.
- **Services:** It gets web API services.
- **ParameterBindingRules:** It gets a collection of rules for how parameters should be bound.
- **MessageHandlers:** It sets or gets message handlers.
- **Formatters:** It sets or gets media-type formatters.

32. Can we return View from ASP.NET Web API method?

No, we cannot return the view from the ASP.NET Web API method. ASP.NET web API develops HTTP services that provide raw data or information. ApiController in ASP.NET MVC application only renders data that is serialized and sent to the client. One can use a controller to provide normal views.

33. What is content negotiation in ASP.Net Web API?

Content negotiation is basically a process of selecting the best representation from multiple representations that are available for a given response. It simply allows one to choose rather

than negotiate content that one wants to get in response. It is performed at the server-side. In simple words, it chooses the best media type for matters to return a response to an incoming request.

34. Difference between HTTP GET vs HTTP Post?

HTTP (HyperText Transfer Protocol) simply manages request-response between client and server. It works as a request-response protocol between client and server.

HTTP GET: This method is used to get information or data from a respective server at a specified URL.

Example:
GET/RegisterStudent.asp?user=value1&pass=value2

HTTP POST: This method is used to send data or information to respective servers.

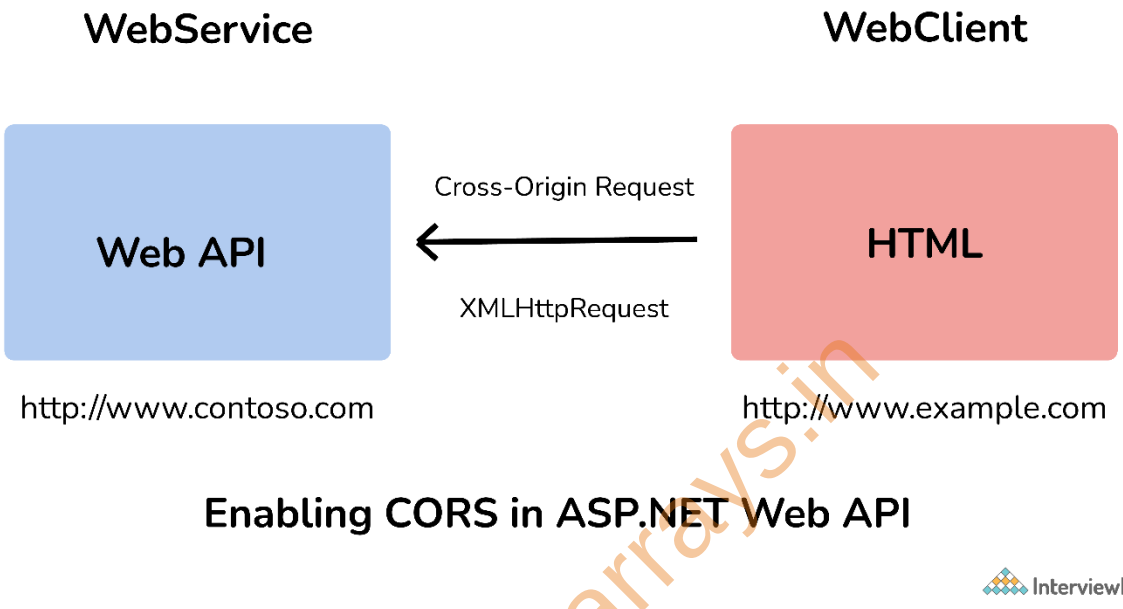
Example:
POST/RegisterStudent.asp HTTP/1.1
Host: www.guru99.com
user=value1&pass=value2

HTTP GET	HTTP POST
Its parameters are included in the URL.	Its parameters are included in the body.
This method is used to request data from specified resources and has no other effect.	This method is used to send data to a server to create or update resources.
It carries a request parameter appended in the URL string.	It carries request parameters in the message body that make it a more secure way of sending data or information from the client to the server.
Request method using GET is cacheable.	Request method using POST is not cacheable.
GET requests are less safe than POST.	Post request is safer than GET.
There is a restriction on data type in GET method and only ASCII characters are allowed.	There are no restrictions on data type in this method and binary data is also allowed.
Data is visible to everyone in the URL.	Data is not displayed in the URL. It is present in the payload.

35. What is CORS in Web API?

CORS (Cross-Origin Resource Sharing) is basically a mechanism that allows one to make requests from one website to another website in a browser that is normally not allowed by

another policy called SOP (Same Origin Policy). It supports secure cross-origin requests and data transfers among clients or browsers and servers. Here, cross-origin request means requests coming from different origins. CORS simply resolves the same-origin restriction for JavaScript. One can enable CORS for web API using the respective web API package or OWIN middleware.



Enabling CORS in ASP.NET Web API

InterviewBit

36. Name method that validates all controls on page?

Page.Validate()

37. What parameters can be passed in the URL of API?

Context keys, documents keys, or anything that initiates API to hit the exact end-point are few parameters that one can pass in the URL to define the complete end-point.

38. What is the use of DelegatingHandler?

DelegatingHandler is used to develop a custom Server-Side HTTP Message Handler in ASP.NET Web API. It is used to represent Message Handlers before routing in Web API.

39. Web API uses which library for JSON serialization?

Json.NET library is used by Web API for JSON serialization.

40. Explain method to handle error using HttpError in Web API?

CreateErrorResponse is an extension method that can be used in Web API controller methods to return error codes and error messages. It creates an HttpError object and then wraps it inside an HttpResponseMessage object.

41. How to unit test Web API?

Using Web API tools like Fiddler, we can perform unit testing in Web API. Fiddler is basically a free debugging proxy for any browser that can be used to compose and execute various HTTP requests to Web API and check HTTP response. It is simply used for testing restful web services. It allows one to inspect and check both incoming and outgoing data to monitor and modify requests and responses before the browser receives them. Below is given some setting that is needed to be done fiddler:

Fiddler – Compose Tab -> Enter Request Headers -> Enter Request Body and then execute.

Conclusion

42. Conclusion

Web API is an extensible framework that serves us information from the server and is used for developing ASP.NET. It is totally based on Http and is easy to define, expose and consume in a REST-ful way. It is considered an ideal platform for developing RESTful applications on .NET framework. From the above Web API interview questions and answers, you can learn more about Web API, its importance, its benefits, etc.

→5.Web Services MCQ

1.

Name the component of HTTP response that contains metadata for the HTTP Response message as key-value pairs?

- Status/Response
- Http Version
- Response Header
- Response Body

2.

Which tool is used for automating and generating API?

- OpenAPI and Swagger
- Jersey API and CFX
- Axis and Restlet
- JerseyAPI and Swagger

3.

What is a model in Web API?

- Type of Data
- Shape of Data
- Html Content
- Collection of Data

4.

ActionResult() is _

- Concrete Class
- Abstract Class
- Both A and B
- None of the above

5.

ViewResult() is _

- Abstract Class
- Concrete Class
- Both A and B
- None of the above

6.

Name object that is not a component of ASP.NET Web API

- File Access
- AdRotator
- LinkCounter
- Counter

7.

Which of the following file extensions is used for ASP.NET Web API?

- .ASPX
- .ASP
- .Web
- None the of above

8.

Web forms are inherited from which of the following base classes?

- Page Class
- Session Class
- Master Page
- None the of above

9.

Full Form of WSDL:

- Web Server Description Language
- Web Server Descriptor Language
- Web Services Description Language
- Web Server Descriptor Language

10.

Web API supports which protocol?

- UDP
- TCP
- HTTP

None of the above

→6.VB.NET INTERVIEW QUESTIONS:-

→REFERENCE → INTERVIEW BIT:-

What is VB.NET?

Visual Basic or VB.NET can be termed as an **Object-Oriented programming** language developed on Microsoft's .NET Framework. Windows apps, Web apps, and Web services are all developed with VB.NET. Vb.Net is an extension of the ancient Visual Basic language; however,VB.NET is not backwards compatible with VB6, and all the code created in the previous version will not compile in VB.NET. Object-oriented ideas are fully supported in VB.NET. Mono is an open-source alternative to the .NET framework. It can also run VB.NET programs, not just on Windows but also on Linux and Mac OSX.

Features of VB.NET:

Some of the features of VB.NET are as follows:

- All the elements in VB.NET are objects, including primitive types, for instance, Integer, Short, Long, String, Boolean, and so on) as well as user-defined types, assemblies, and events. The basic class Object is inherited by all objects. Microsoft's .NET framework is responsible for the development of VB.NET. It has complete access to each of the .Net Framework's libraries.
- Applications built with the .Net framework are cross-platform. The framework was created in a manner such that it may be used with each of the programming or scripting languages listed below: Visual Basic, C++, JavaScript, COBOL, and many more.
- Every one of these languages can be integrated with each other and access the framework. The .Net framework is made up of a massive library of code that is being applied by client languages such as VB.Net. Object-oriented techniques are used in these languages.

→6.VB.Net Interview Questions for Freshers

1. State some advantages of using VB.NET.

Some benefits of using VB.NET are as follows:

- VB.NET is a modern, general-purpose language.
- It is an object-oriented programming language.
- For a newbie, VB.NET is quite simple to learn.
- It is a structure-oriented programming language.
- VB.NET can be built on a number of different platforms.
- Conditional Compilation is supported by VB.NET.
- VB.NET has a Standard Library and Automatic Garbage Collection.
- It has Property and Event support.
- VB.NET aids in the management of delegates and events.
- Generics, Indexers, and Simple Multithreading are all supported.

2. What do you understand by metadata and namespace? Name the namespace that can be used to access data? What do you understand about JIT?

- **Metadata:** Metadata is defined as "data on the body of the data" and can be found in library catalogues. In practice, metadata's usage comes in evaluating database data, although it can also be utilised for other purposes.
- **Namespace:** In the.NET language, namespaces are nothing but an organised manner of representing classes, structures, and interfaces. All .NET Languages have access to namespaces, which are hierarchically organised indexes of a class library.
- **JIT:** It is an acronym for Just in Time compiler, which is a component of the runtime execution environment. JIT is divided into three categories:
    - **Normal JIT:** Normal JITs compile called functions or procedures at runtime and compile them for the first time when they are invoked.
    - **Pre-JIT:** Pre JITs compile an application before it is deployed.
    - **Econo-JIT:** Econo JITs compile called functions and procedures at runtime.

The **System.Data** namespace is used to access and manage data from the required data source. This namespace deals only with the data from the specified database.

3. State some key differences between VB.NET and Visual Basic.

Comparison Parameter	VB.NET	VB
Platform Dependence	VB.NET is platform-independent.	VB is platform dependent.
Backwards Compatibility	VB.NET is not backwards compatible.	VB is backwards compatible.
Compiled or Interpreted	VB.NET is a compiled language.	VB is an interpreted language.
Exception Handling	In VB.NET, we can perform Exception Handling using the 'Try….Catch' statements.	In VB, we can perform Exception Handling using the 'On Error…..Goto' statements.
Development of Multithreaded Applications	It is possible to develop multi-threaded applications using VB.NET.	It is not possible to develop multi-threaded applications using VB.

You can download a PDF version of Vb Net Interview Questions.

Download PDF

---

4. State some key differences between C# and VB.NET.

Comparison Parameter	C#	VB.NET
Optional Parameters	Optional Parameters will be allowed in C#.	VB.NET does not allow optional parameters.
Case Sensitivity	C# is not case sensitive.	VB.NET is case sensitive.
Error Handling	C# supports structured error handling as well as unstructured error handling.	VB.NET supports only unstructured error handling.
Release of Unmanaged Resources	In C#, unmanaged resources cannot be released.	In VB.NET, the "Using" keyword can be used for releasing unmanaged resources.

5. What do you understand about assembly in VB.NET? What do you understand about Strong Name with reference to a .NET assembly?

Assemblies are a component of .NET applications and are considered to be the main part of all .NET apps. They can be either a DLL (Dynamic Link Library) or an executable file. The two forms of assemblies are as follows:

- **Private** - Private assemblies are those that are only utilised by one application and are kept in the application's directory.
- **Public** - Public assemblies, also known as shared assemblies, are saved in the Global Assembly Cache (GAC), which can be accessed by several programs.

Strong Name is nothing but a feature of .Net whose usage lies in uniquely identifying shared assemblies. A strong name is a solution to the problem of several objects with the same name being created, and it may be assigned using Sn.exe.

6. State your understanding about Option Strict and Option Explicit.

In general, .Net allows any data type to be implicitly converted.

- **Option Strict** is used to prevent the leakage of data during the conversions of data type, and it guarantees the compile-time notification of certain forms of conversions.
- **Option Explicit** is a keyword in a file that is used to exclusively declare each and every variable using declare keywords such as Dim, Private, Public, and Protected. An error occurs at compile time in the event of an undeclared variable name persisting.

7. What is the base class of VB.NET?

The **System.object** class is the base class of VB.NET.

8. In the .Net Framework, what is the INTERNAL keyword?

An access specifier that will be displayed in a specific assembly, that is, in a DLL (Dynamic Link Library) file, is the INTERNAL keyword. This is visible throughout the assembly as a unique binary component.

9. Explain the usage of the NEW keyword with an example.

The constructor is used along with the NEW keyword. Its usage can be as that of a modifier or an operator. The NEW keyword hides inherited members from base class members when used as a modifier. It produces objects for invoking constructors when used as an operator. The NEW keyword creates a new object instance, provides a constructor constraint on a type argument, or designates another function as a class constructor. A New clause can be used in either a declaration or an assignment statement. When you run the statement, it calls the proper constructor of the specified class, passing any parameters you have given it.

An example of the usage of the NEW keyword is given below:

```
Dim teacher1 As New Teacher()
```

10. What is ReDim statement?

The ReDim statement is used to resize or enlarge a dynamic array that has already been formally declared with empty parenthesis using a Private, Public, or Dim declaration (without dimension subscripts). To alter the number of elements and dimensions in an array, use the ReDim statement repeatedly. However, unless the array is included in a Variant, one can't define an array of one data type and then use ReDim to transform it to another data type. If the array is stored in a Variant, the type of the items can be changed with an As type clause, unless the Preserve keyword is used, in which case no data type changes are allowed.

An example of the usage of the ReDim keyword is given below:

```
Dim DemoArray() As Integer ' Declaring a dynamic array.
Redim DemoArray(10) ' Allocating 10 elements.
For J = 1 To 10 ' Looping for 10 times.
DemoArray(J) = J ' Initializing the DemoArray.
Next J
```

The ReDim statement is used in the above example to allocate and reallocate storage space for dynamic array variables in this example. The Option Base is assumed to be 1. The ReDim keyword is only applicable to arrays and is used to modify the size of one or more dimensions of an array that has already been declared. When needed, Redim can free up or add elements to an array.

11. Explain Jagged arrays in VB.NET.

An array of arrays array named scores of Integers is demonstrated in the code below. is called a Jagged array. Declaring a jagge:

```
Dim marks As Integer()() = New Integer(10)(){}
```

The example code snippet below shows how to use a jagged array:

```
Module demoArray
  Sub Main()
     ' making one jagged array of 3 arrays of integers
     Dim arr As Integer()() = New Integer(2)() {}
     arr(0) = New Integer() {5, 6}
     arr(1) = New Integer() {15, 10}
     arr(2) = New Integer() {32, 60}
     Dim a, b As Integer
     ' printing the value of every array element

     For a = 0 To 3
        For b = 0 To 1
           Console.WriteLine("arr[{0},{1}] = {2}", a, b, arr(a)(b))
        Next b
     Next a
     Console.ReadKey()
  End Sub
End Module
```

The output of the above code snippet will be as follows:

```
arr[0][0]: 5
arr[0][1]: 6
arr[1][0]: 15
arr[1][1]: 10
arr[2][0]: 32
arr[2][1]: 60
```


12. What are class access modifiers?

Keywords that are used to specify the declared accessibility of a member or a type is known as class access modifiers. The various types of class access modifiers are as follows:

- Public
- Private
- Protected
- Protected Internal
- Internal

13. What do you understand about globalization in the .NET framework?

Globalization is essentially the process of adapting your application to different cultures. You must recognise that your application's users may speak a different language than you and that their language, in addition to being a different language, follows different rules. Some languages, for example, should be displayed right to left, whereas others should be displayed left to right. You must now consider not just the various languages, but also the various

currencies, date and time settings, and the various colours associated with various civilizations. With globalisation, you can make your software accessible to people from all over the world, regardless of their language.

14. What do you understand about an assembly manifest in the .NET framework?

An assembly manifest is a text file in the.NET Framework that contains metadata about the code in a CLI assembly. It describes the components' relationships and dependencies, as well as the assembly's versioning information, scope information, and security permissions. The Manifest file type can be saved as a PE file type. As a Manifest, you can save the Assembly Name, Version, Culture, and key token.

15. Define Nested Classes and Enumerators in VB.NET.

- The classes that can be declared within another class's scope are known as **nested classes**. These classes are included in the contained class's scope and are available inside that class's scope.
- A value type containing a set of constants given to the set of the list is known as an **Enumerator** or **Enum**. When more than one number needs to be defined, enumeration is utilised.

VB.Net Interview Questions for Experienced

16. Define Garbage Collection in VB.NET. State your understanding of the Dispose() and Finalize() methods.

Garbage collection in VB.NET, often known as automatic memory management, is a technique for discarding dynamically allocated memory automatically. Garbage collection is handled by a garbage collector, who will recycle memory if it is determined that it will be used in the future.

- The garbage collector invokes the Finalize() method, which aids in the cleanup of unmanaged resources. Other resources, such as window handles and database connections, are managed through the iDisposable interface.
- To explicitly release unused resources, the Dispose() method is handled by the IDisposable interface. Even if other references to the object are alive, Dispose() method can be called.

17. Differentiate between the following: Dataset Vs DataReader?

Dataset	DataReader
Datasets are capable of holding multiple tables coming from a single data source. A dataset can even store relationships between those tables.	DataReader is having read-only access to the data. Also, DataReader has been set to be forward only.
Dataset is disconnected architecture.	DataReader is connected architecture.

Dataset	DataReader
Dataset is capable of persisting content.	DataReader is not capable of persisting contents because of the fact that it is read-only.

18. What is delegates in VB.NET?

A delegate is a type of object that can be used to refer to a method. When you assign a delegate to a method, it functions precisely like that method. Delegates are objects that serve as placeholders for methods. Because they are comparable to function pointers used in other programming languages, they are also referred to as type-safe function pointers. Visual Basic delegates, unlike function pointers, are a reference type based on the class System.Delegate.

19. What do you understand about Authentication and Authorization? What are the different types of authentication?

The process of getting credentials from users and validating their authenticity is known as authentication. The process of granting access to authenticated resources is known as authorization. The authorization follows Authentication and authorization can be done in various ways, for instance, users can be authorised to have various kinds of access to the application such as READ ONLY access, Write Access, Delete Access, etc. Both application architects and developers are concerned about authentication. Applications that store sensitive data must be safeguarded against hostile assaults and competitors attempting to steal data or intellectual property. When building a security model for your application, keep in mind the business needs for authentication as well as the impact that a security model choice can have on performance, scalability, and deployment.

Authentication can be divided into various categories:

- Windows Authentication
- Passport Authentication
- Form Authentication
- Anonymous Authentication
- Basic Authentication
- Digest Authentication
- Port Authentication
- Certificate Authentication
- Using Cookies

20. What are the different sorts of garbage collector generations?

In the garbage collector, there are three generations:

- **Generation 0:** It represents a brand new object that has never been marked for collection.
- **Generation 1:** It identifies an object that has been designated as a collection but has not yet been removed.
- **Generation 2:** This indicates an object that has been swept by the Garbage Collector more than once.

21. What is Global Assembly Cache (GLC)? What is it used for?

The Global Assembly Cache (GLC) is used to store shared .NET assemblies. The following criteria make use of it:

- If there are any unique security needs for .Net assemblies.
- If a .Net application has to share data with other programs.

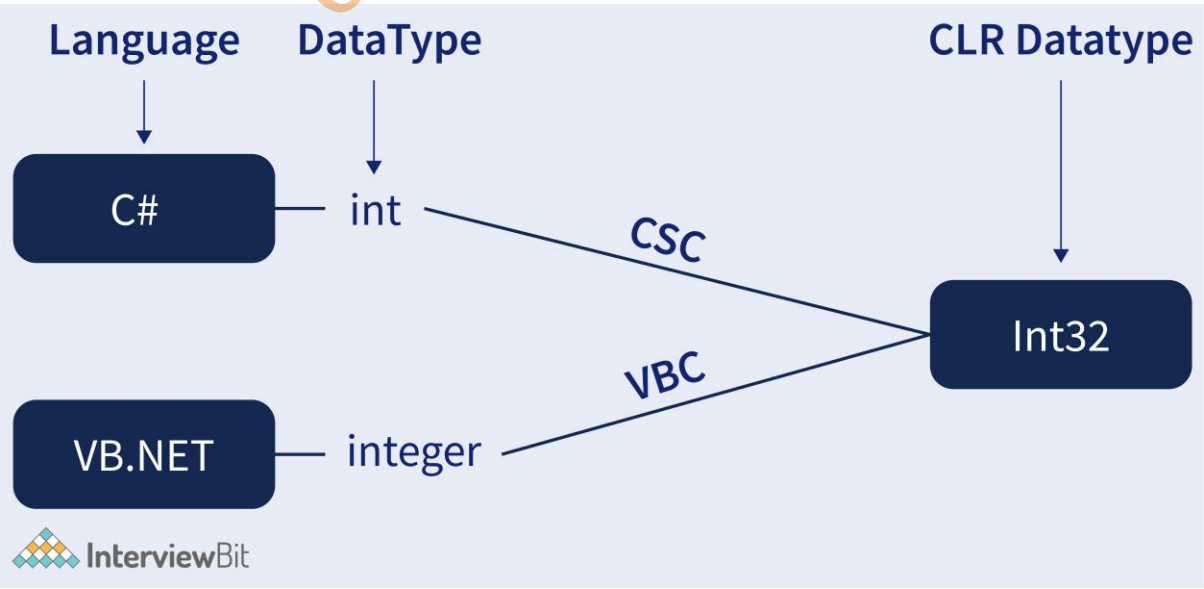
22. What is Common Language Runtime or CLR?

CLR stands for Common Language Runtime, and it is the brain behind the .NET framework. It is the responsibility of the runtime to ensure that the program's code is executed correctly. The following is taken care of by CLR:

- Verification of Garbage Collection Codes.
- IL (Intermediate Language) for Code Access Security.

23. What are Common Type System(CTS) and Common Language Specification(CLS)?

Common Type System (CTS): The data types that managed code can use are described by the Common Type System (CTS). In the runtime, CTS describes how these types are declared, utilised, and managed. Cross-language integration, type safety, and high-performance code execution are all made easier using it. You can utilise the CTS rules to create your own classes and values. Alternatively, we can comprehend something like this that the data type is dealt with by CTS. So we have a number of languages, each of which has its own data type. While one language's data type may be incomprehensible to other languages, the .NET Framework language can comprehend all data types. Integer data types are available in both C# and VB.NET. As a result, a variable specified as an int in C# and Integer in VB.NET utilises the same structure Int32 from CTS after compilation as shown in the figure below. All of the structures and classes in CTS are shared by all.NET languages, and their purpose is to support language NET's independence. As a result, it is known as CTS.



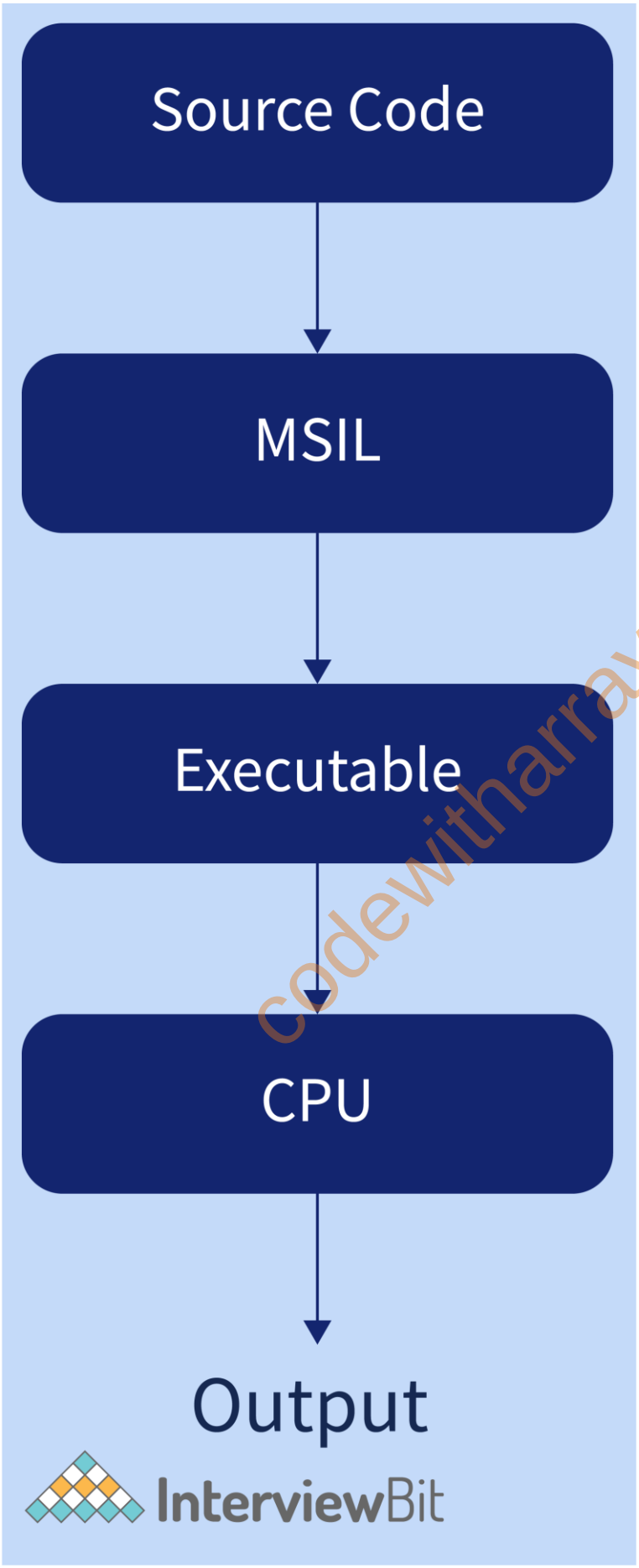
Common Language Specification(CLS): CLS is a subset of CTS and stands for Common Language Specification. It specifies a set of rules and constraints that must be followed by any language that runs on the .NET framework. CLS Compliant languages are those that follow this set of rules. CLS facilitates cross-language integration, or interoperability, in basic terms. As an example, if we are talking about C# and VB.NET, every statement in C# must end with a semicolon. Although it is often known as a statement Terminator, that is, each statement in VB.NET should not conclude with a semicolon (;). An explanation of the mentioned example is that while the syntax rules change from language to language, CLR can understand all of them because, in .NET, each language is transformed into MSIL (Microsoft Intermediate Language) code after compilation, and MSIL code is CLR's language specification.

24. What do you understand by Managed Code in VB.NET?

Managed Code is a type of code that is created to use the functionalities of a controlled runtime environment, such as the CLR (Common Language Runtime) in the .NET Framework. Instead of being directly performed by the operating system, it is always implemented by the controlled runtime environment. The controlled runtime environment provides a lot of features, for instance, garbage collection, handling of exceptions, type checking, bounds checking, and other services to programs without requiring the intervention of the user. It also allocated memory to the code, type safety, and other features. Applications built-in languages such as Java, VB.Net, and many others are always aimed at runtime environment services to manage execution, and programs coded in these languages are referred to as managed code.

In the scenario of the .NET Framework, the compilation of the managed code is always done by the compiler in MSIL (Microsoft Intermediate Language) before creating an executable. The CLR's Just In Time (JIT) Compiler translates the intermediary language into native code that is particular to the architecture lying beneath when the programmer runs the executable. This procedure is running in a managed runtime execution environment, and the environment is in charge of the code's functionality. The managed code is executed as indicated in the image below, with the source code coded in any .NET Framework language.

Because managed code is compiled into an intermediate language, the JIT compiler turns this intermediate language into architecture-specific instructions, providing platform independence.



25. State the advantages of Managed Code in VB.NET. What is the primary disadvantage of managed code?

Some of the **advantages** of Managed Code in VB.NET are as follows:

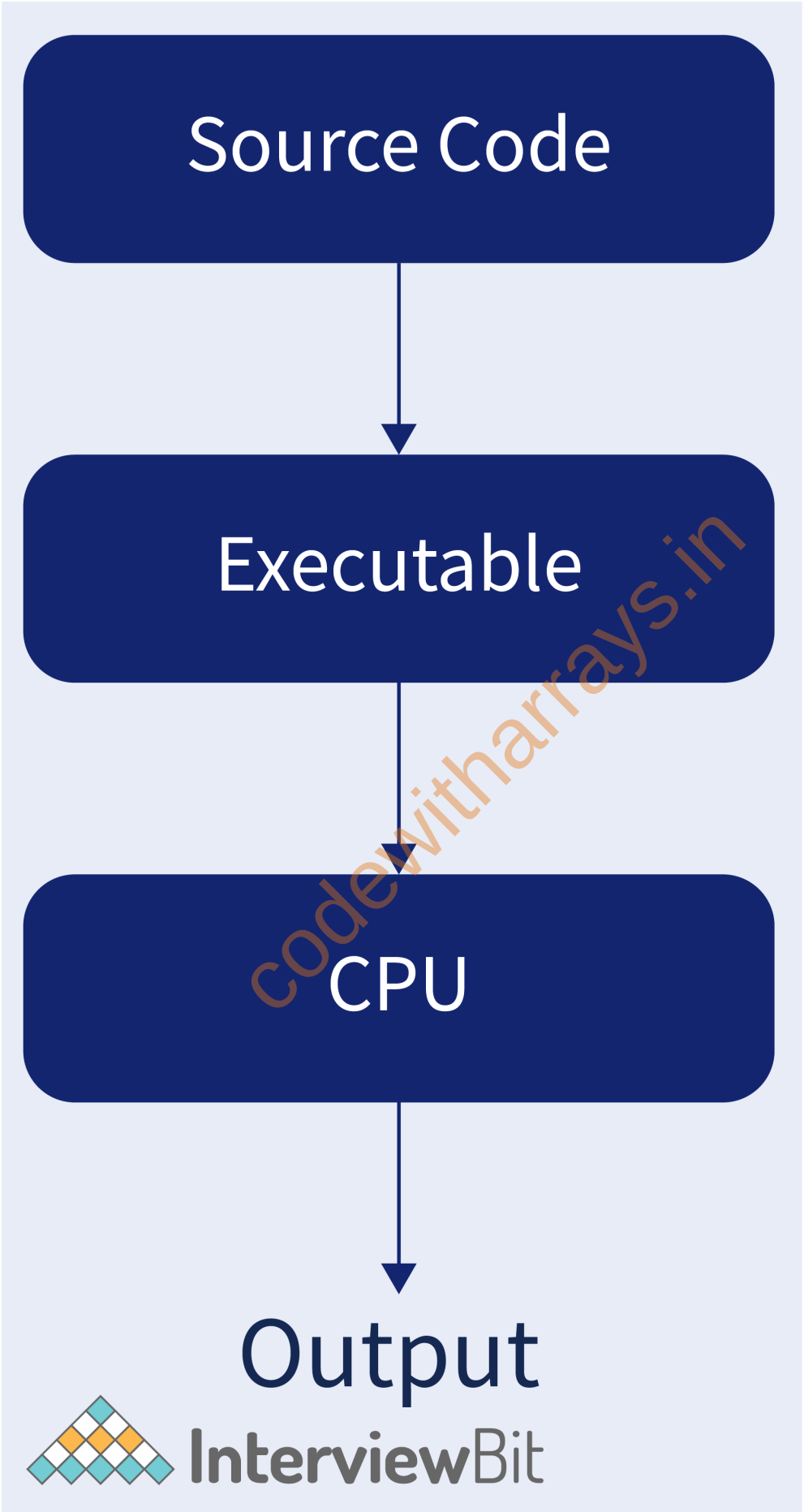
- It implements trash collection on its own.
- It improves the application's security by automatically checking memory spaces for preventing buffer overflow when using the runtime environment.
- It also checks references, which determines whether or not the reference points to a genuine object and whether or not they are duplicates.
- It also includes type checking at runtime and dynamic type checking.

The **fundamental drawback** of managed languages is that we can't allot memory on our own, and we can't access the architecture of the Central Processing Unit or CPU at a basic level. C#, Visual Basic, F#, and other high-level languages that may be executed on top of .NET are examples of managed code. You do not receive machine code when you compile code written in those languages with their respective compilers. The runtime then compiles and executes the Intermediate Language code. The only exception to this rule is C ++, which may generate native, unmanaged binaries for Windows. Runtime engines include Java, Visual Basic, and the.NET Common Language Runtime (CLR). A DBMS may also have a runtime engine for the programming language it supports.

26. What is Unmanaged Code in VB.NET ? State the benefits and drawbacks of unmanaged code.

Unmanaged code is code that the operating system directly executes. It has always focused on the architecture of the processor and is dependent on the architecture of the computer. On compilation, this code tends to choose a particular architecture and execute on that platform. Loosely speaking, if you wish to run similar codes on a different architecture, you must recompile it in accordance with that architecture. The compilation is always done to the native code, which is architecture-specific.

The developer manages to allot space, security, type safety, and other aspects of unmanaged programming. As a result, various memory-related issues arise, such as space overflow, memory leakage, override of pointers, and so on. Unmanaged code executable files are usually binary images, x86 code that is loaded directly into memory. Unmanaged code is used in applications created in VB 6.0, C, C++, and other languages. The following graphic depicts the running of an unmanaged code:



Some of the benefits of Unmanaged Code are as follows:

- Unmanaged code gives the programmer low-level access.
- It can also allow us to access the hardware directly.
- It enables the user to skip certain of the managed code framework's parameters and restrictions.

Some of the drawbacks of Unmanaged Code are as follows:

- The application is not secure as a result of it.
- Memory-related difficulties, such as memory space overflow, arise as a result of access to memory allocation.
- The programmer is also in charge of handling errors and exceptions.
- It is not concerned with rubbish collection.

27. Can the .tostring method handle NULL values?

No, the .tostring method cannot handle NULL values. Instead of it, we can use Convert .tostring method to handle NULL values.

28. Define application domains. How can we create application domains?

- A lightweight process controlled by the .NET runtime can be thought of as an application domain. Hosts such as windows shell, asp.net, and many more build application domains.
- For example, when you execute a .net application from the command line, the host is the shell. For each application, the shell establishes a new application domain.

29. Explain garbage collection in VB.NET? State the advantages of garbage collection.

The garbage collector in VB.NET regulates memory allocation and release for your application. When you create a new object, the managed heap is used to allocate memory for it via the common language runtime. The runtime continues to allocate space for new objects as long as address space in the managed heap is available. Memory, on the other hand, is finite. In order to liberate some memory, the garbage collector must eventually complete a collection. Based on the allocations being made, the garbage collector's optimising engine calculates the optimal moment to do a collection. When the garbage collector does a collection, it searches the managed heap for objects that are no longer in use by the application and executes the appropriate procedures to free up their memory. The garbage collector (GC) is an automatic memory manager in the Common Language Runtime (CLR). The garbage collector is in charge of managing an application's memory allocation and release. This means that you do not have to create code to handle memory management duties if you are dealing with managed programs. Automatic memory management can solve problems like forgetting to free an item, resulting in a memory leak, or trying to access memory for an object that has already been freed.

A few advantages of garbage collection are as follows:

- It eliminates the need for developers to manually release memory.
- Objects are efficiently allocated on the controlled heap.

- Objects that are no longer in use are reclaimed, their memory is cleared, and the memory is made available for future allocations. Managed objects come with clean content out of the box, so their builders don't have to worry about filling in every data field.
- Provides memory safety by ensuring that an object cannot use memory assigned to another object for its own purposes.

Conclusion:

In this article, we aim to make our readers aware of what VB.NET is and what are the various features offered by this framework. It is an extremely important topic from the point of view of a technical interview. We have also discussed various questions on VB.NET with difficulty ranging from easy to hard as asked in any VB.NET interview. We hope that our readers get a good insight on VB.NET using this article and ace their interview.

→6.VB.NET MCQ

1.

Visual Basic was developed by:

- Symantec
- Sybase
- Ashton Tate
- Microsoft

2.

In Visual Basic, which of the following extensions is used to indicate the project file?

- .vvb
- .vbp
- .vb
- .cls

3.

An undeclared data type takes which Data types by default in Visual Basic?

- Object
- Int

- Char
- String

4.

Which of the following converts the expression to Bool data type in VB.NET?

- CChar(expression)
- CBool(expression)
- CByte(expression)
- CDate(expression)

5.

Which access modifier specifies that an attribute at the start of a source file applies to the entire assembly?

- Assembly
- Auto
- Ansi
- Async

6.

Which of the following access modifiers states that an argument is supplied in such a way that the called procedure or property cannot affect the value of a variable in the calling code that is underlying the argument?

- ByRef
- Default
- ByVal
- Friend

7.

When a procedure is called, which of the following access modifiers specifies that a procedure parameter might be omitted?

- Out
- Overload
- Optional
- NonOverridables

8.

Which of the following access modifiers indicates that a class or structure declaration is only a partial definition?

- Partial
- Overrides
- Overridable
- ParamArray

9.

Which of the following access modifiers tells Visual Basic to convert all strings to Unicode values, regardless of the name of the external procedure?

- Shared
- Shadows
- Unicode
- Static

10.

The operator symbol, operands, and code that define an operator procedure on a class or structure are declared in which of the following statements?

- Declare
- Property
- Sub
- Operator

THANK -YOU:-

https://www.youtube.com/@codewitharrays

https://www.instagram.com/codewitharrays/

https://t.me/codewitharrays Group Link: https://t.me/cceesept2023

+91 8007592194 +91 9284926333

codewitharrays@gmail.com

https://codewitharrays.in/project