1. **Net 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](https://www.interviewbit.com/oops-interview-questions/).  
     
   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.
2. **Net Core**.NET Core is a newer version of the .NET framework, and it is a general-purpose, cost-free, open-source development platform developed by Microsoft.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.

3) **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.

1. **Explain about major components of the .NET framework.**

**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 function 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.

1. What is an EXE and a DLL?

**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.  
  
6) **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

* + - * 1. **Explain CLS?**

It’s helping 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 the .NET-compatible languages.

* + - * 1. **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.

9) **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.  
  
10) **Explain the differences between value type and reference type.**

* 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.

11) **What is the difference between managed and unmanaged code?**

| **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. |

12) **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.  
  
13) **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.  
  
14) **Is ASP.NET different from ASP? If yes, explain how?**

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**.**

15) 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

16) **Explain the different types of assembly.**

**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).

17) **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

18) **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.

19) **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

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.

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?**

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.

26) **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