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 **Net Core**.NET Core is a newer version of the .NET framework, and it is a general-purpose, cost-free, open-source development platform developed by Microsoft.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 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 Find Control 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

**SET-2**

1. What is c#?
   * C# is a modern, object-oriented programming language developed by Microsoft as part of the .NET framework. It was first released in 2000, and it has become one of the most widely used languages for building Windows applications, web services and more. C# combines the power of C and C++ with the simplicity of languages like Java and VB.
   * Key Features: Object-oriented Programming, Simplicity, Robustness, Security,Portability,High Performance.
2. 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# support automatic garbage collection by Common Language Runtime (CLR) while C does not. C# primarily needs .Net Framework to execute while C is a platform-agnostic language.
3. What is Inheritance?
   * Inheritance is a fundamental concept in object-oriented programming that allows a child’s class to inherit the properties from the superclass. The new class inherits the properties and methods of the existing class and can also add new properties and methods of its own. Inheritance promotes code reuse, simplifies code maintenance, and improves code organization.
   * \* Code Reusability
   * \* Code Maintenance

\* Code Organization

**Single Inheritance:** A derived class that inherits from only one base class.

**Multi-level Inheritance:** A derived class that inherits from a base class and the derived class itself becomes the base class for another derived class.

**Hierarchical Inheritance:** A Base class that serves as a parent class for two or more derived classes.

**Multiple Inheritance:** A derived class that inherits from two or more derived classes (using Interface)

**Hybrid Inheritance:** A Hybrid Inheritance is a mix of two or more inheritance.

1. What is the difference between an Array and Array List in C#?
   * An array is a collection of similar variables clubbed together under one common name. While Array List is a collection of objects that can be indexed individually. With Array List you can access a number of features like dynamic memory allocation, adding, searching, and sorting items in the Array List.
2. **What are extension methods in C#?**
   * Extension methods help to add new methods to the existing once. 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 right’s , 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.
3. Difference Between Ref and Out Keywords in c#.
   * The out is a keyword in c# which is used for the passing the arguments to methods as a reference type. It is generally used when a method returns multiple values. The out parameter does not pass the property.
   * The ref is a keyword in c# which is used for the passing the argument by a reference. Or we can say that if any changes made in this argument in the method will reflect in that variable when the control returns to the calling method. The ref parameter does not pass the property.
4. Difference between Abstract Class and Interface in c#?
   * To declare an abstract class, we use the abstract keyword. An Abstract class is never intended to be instantiated directly. This class must contain at least one abstract method, which is marked by the keyword or modifier abstract in the class definition. The Abstract classes are typically used to define a base class in the class hierarchy.
   * Like a class, Interface can have methods, properties, events, and indexers as its members. But interfaces will contain only the declaration of the members. The implementation of interface’s members will be given by the class who implements the interface implicitly or explicitly.
5. What are the types of classes in C#?
   * There are 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.

**Partial class:** It is 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.

1. What is garbage collection?
   * 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.
2. What is Common Language Runtime?
   * The Common Language Runtime (CLR) is a component of the Microsoft .NET Framework that manages the execution of .NET applications. It is responsible for loading and executing the code written in various .NET programming languages C#, VB.NET,F# and others.
   * When a C-Sharp program is complied, the resulting executable code is in an intermediate language called Common Intermediate Language (CIL) or Microsoft Intermediate Language (MSIL). This code is not machine-specific, and it can run on any platform that has the CLR installed. When the CI code is executed, the CLR compiles it into machine code that can be executed by the processor.
3. What are partial classes?
   * A Partial class is a special feature of c#. It provides a special ability to implement the functionality of a single class into multiple files and all these files are combined into a single class file when the application is compiled. A partial class is created by using a partial keyword. This keyword is also useful to split the functionality of methods, interfaces, or structure into multiple files.
4. What is the difference between String and StringBuilder in c#?
   * The major difference between string and string Builder is that String is that String objects are immutable while StringBuilder creates a mutable string of characters. StringBuilder will make changes to the existing object rather than creating a new object.
   * StringBuilder simplifies the entire process of making changes to the existing 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 pictures which can be evoked using the System. Text namespaces.
5. What is the difference between constant and read only 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.
   * On the other hand, with read-only keyword, you can assign the variable only when it is declared or in a constructor of the same class in which it is declared.
   * Constants are static by default while read-only should have a value assigned when the constructor is declared.
6. **What is Reflection in c#?**
   * Reflection is the process of describing the metadata of types, methods and fields in a code. The namespace System. Reflection enables you to obtain data about the loaded assemblies, the elements within them like classes, methods and value types. Some of the commonly used classes of Syste.Reflection are,
   * 1) Assembly 2) AssemblyName 3) ConstructorInfo 4) MethodInfo 5) ParameterInfo, 5) EventInfo
7. **Difference between the Equality Operator (==) and Equals () Method in c#?**

Equality Operator: It is a reference type which means that if quality 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.

1. What is difference between Early and late binding in c#?
   * **Early Binding**: It recognizes and checks the methods or properties during compile time. In this binding, the compiler already knows about what kind of object it is and what are the methods or properties it holds, here the objects are static objects. The performance of early binding is fast, it is easy to code. It decreases the number of run-time errors.
   * **Late Binding:** The compiler does not know about what kind of object it is and what are the methods or properties it holds, here the objects are dynamic objects. The type of object is decided on the basis of the data it holds on the right-hand side during runtime. The performance of late binding is slower than early binding because it requires lookup at runtime.
2. **What are Properties in C#?**
   * Properties are the special types of class members that provide a flexible mechanism to read, write or compute the value of a private field. Properties function like public data members but are accessors which make easy data access. Properties also support encapsulation, and abstraction through “get”
   * and “set” methods for accessing and modifying them.

### **18) https://www.interviewbit.com/c-sharp-interview-questions?**

ANS: https://www.interviewbit.com/c-sharp-interview-questions/  
  
**19)** **Write a c# program to find the substring from a given string?**  
ANS: https://www.interviewbit.com/c-sharp-interview-questions/  
  
**20)** **Write a program in c# Sharp to find if a given string is palindrome or not?**  
ANS: https://www.interviewbit.com/c-sharp-interview-questions/  
  
**21)** **Write a program in c# sharp to reverse the order of the given words?**

ANS: https://www.interviewbit.com/c-sharp-interview-questions/  
  
**What is the purpose of the using statement in C#?**

**Write a program in c# sharp to reverse a string?**  
ANS: https://www.interviewbit.com/c-sharp-interview-questions/  
  
**23)** **What is the purpose of the using statement in c#?**  
  
The using statement in c# is used to manage resource automatically by ensuring that objects implementing IDisposable are disposed of after use. This helps prevent memory leaks and ensures efficient resource management.

**24) What is the significance of the var keyword in c#?**

In C#, the var keyword allows for implicit typing of variable, meaning the compiler determines the variable’s type at compile-time based on the assigned value.

**Features**:  
1) Implicit Typing  
2) Strongly Typed  
3) Compile-Time Evaluation  
4) Reduces Code Clutter  
  
**25) Explain the purpose of the async and await keywords?**

* Async is used to declare an asynchronous method, and wait is used to asynchronously wait for a task to complete, allowing other tasks to run in the meantime.

**26) Difference between == and Equals methods in c#?**

**\*** == is used for comparing the values of two variables, while the Equals method is used for comparing the contents of objects.

**27) What is a Delegate in c#?**

**\*** A delegate in c# is a type-safe function pointer that holds a reference to a method. It allows methods to be passed as parameters, assigned to variables, and called dynamically at runtime.

28) What is LINQ in c#?  
\* LINQ (Language-Integrated Query) is a query syntax in c# that allows querying collections, databases, XML, and more using a consistent syntax similar to SQL. It makes data manipulation more readable, concise, and type safe.

29) What is Garbage Collection in c#?

\* in c#, Garbage collection (GC) is an automatic memory management feature that frees unused objects from memory, preventing memory leaks and improving performance.

30) What is the use of Singleton Design Pattern in c#?

\* The Singleton Design Pattern ensures that only one instance of a class is created and shared across an application. This pattern is useful for scenarios where having multiple instances could lead to inconsistencies or excessive resource usage.

31) What is the difference between throw and throw ex in c#?

\* Throw is used to rethrow the current exception, preserving the original stack trace, while throw ex discards the original stack trace and starts a new one.

32) How does the try, catch and finally blocks works in c#?

\* try is used to enclose the code that might throw an exception, catch is used to handle the exception, and finally is used to specify a block of code that will be executed, regardless of whether an exception is thrown or not.

33) How does the IEnumerable interface work in c#?

\* The IEnumerable<T> interface in c# is used to iterate over collections using a foreach loop. It is a key part of LINQ and lazy evaluation, making it efficient for working with large data sets.

34) How does the lock statement work in c#?

\* The lock statement is used to synchronize access to a shared resource by acquiring a mutual exclusion lock. It helps avoid race conditions and data corruption in multithreaded applications.

35) What is the role of the params keyword in c#?

\* The params keyword in c# allows you to pass a variable number of arguments to method as an array, without explicitly creating an array in the method call.  
  
36) **Explain the concepts of nullable types in c#?**

\* Nullable types allow variables to be assigned to a value or null. They are useful when dealing with databases or other situations where values may be missing.

37) **How does the yield keyword work in C#?**

\* The yield keyword in c# is used to simplify the creation of iterators. It primarily used in methods that return IEnumerable<t> or IEnumerable<T>,allowing iteration over a collection without creating a full data structure in memory.

**38) Describe the use of the base keyword in c#?**

\* The base keyword is used to access members of the base class from within a derived class.

**39) What is the purpose of this keyword in c#?**

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

**40) Explain the concept of polymorphism in c#?**

\* It allows objects of different classes to be treated as objects of a common base class, providing flexibility, reusability, and extensibility in your code. It allows you to write code that can work with objects of various types, providing a consistent interface to interact with different objects. In c#, polymorphism is primarily achieved through the use of inheritance and interfaces.

**41) Explain the Dispose Method?  
\*** This Dispose method is used to release unmanaged resources explicitly. It’s typically implemented in classes that use unmanaged resources such as file handles, database connections, or network sockets. By calling Dispose, developers can release resources immediately, ensuring timely cleanup and preventing resource leaks.

**42) Explain the Generics?  
\*** Generic is a class which allows the user to define classes and methods with the placeholder. The basic idea behind using Generic is to allow type (Integer, String, etc. and user-defined types) to be a parameter to methods, classes and interfaces. Generic types perform better than normal system types because they reduce the need for boxing, unboxing, and type casting variables or objects.

**43) How does the Nullable<T> struct work in c#?**

**\*** In c#, the compiler does not allow you to assign a null value to a variable. So, c# 2.0 provides a special feature to assign a null value to a variable that is known as the Nullable type. The Nullable type allows you to assign null value to a variable.

**44) What is the purpose of the volatile keyword in c#?  
 \*** The volatile keyword is used to indicate that a field might be modified by multiple threads, preventing compiler optimization that could cause unexpected behavior in a multi-threaded environment.

**45) How does the as keyword work in c#?  
\*** The as operator is used to perform conversion between compatible reference types or Nullable types. This operator returns the object when they are compatible with the given type and return null if the conversion is not possible instead of raising an exception.  **46)** **Explain the role of the static keyword in C#?**

\* The static keyword is used to declare a member that belongs to the type itself rather than to a specific instance of the type.

**47) Difference between value and reference type?** \* Value types are stored on the stack, while reference types are stored on the heap.  
 \* Value types create a copy of the data when they are assigned or passed, while reference types use the same instance of the data when they are assigned or passed.

\* Value types compare the data by value, while reference types compare the data by reference.

\* Value types are immutable, which means that they cannot be changed after they are created, while reference types are mutable, which means that they can be changed after they are created.

\* Value types are used for simple and primitive data, such as numbers, boolens, and structs, while reference types are used for complex and dynamic data, such as objects, arrays and classes.

**48) Describe the concept of inversion of control (IOC) in c#.**\* Inversion of Control is a design principle where the control of the flow of a program is inverted, typically achieved through dependency injection.  
  
**49) What is the purpose of the name of operator in c#?**

**\*** The nameof operator is used to obtain the simple (unqualified) string name of a variable,type,or members.

**50) What is Thread class in c#?**

**\*** The thread class allows the creation of both foreground and background threads.

**\*** It allows the setting and getting of Thread Priorities.

\* It provides properties to check the current state of thread.

\* We can retrieve the reference to the currently executing thread.

\* The Thread class is sealed, meaning it cannot be inherited.  
\* Through the thread class, we can manage thread execution, interruption, and joining.

**51) How does the .NET Framework work?**

**\*** .NET framework-based application that are written in supportive language 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.

**52) Explain CLR (Common Language Runtime).**

\* It is an execution that runs the code and provides services that make the development process easier.

\* CLR is memory management. garbage collections, type safety, exception handling, security, and thread management. It also makes it easier to design the applications and components whose objects interact across languages.

\* The programs written for the .NET Framework are executed by the CLR regardless of programming language. Every .NET Framework version has CLR.

**53) What is FCL (Framework Class Library)?**

\* It has pre-defined methods and properties to implement common and complex functions that can be used by .NET application. It will also provide dates, strings, and numbers.

\* This class library includes APIs for database connection, file reading and writing, drawing,etc.

**54) What is Base Class Library (BCL)?  
\*** The Base Class Library (BCL) has a 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:

**1) User-defined class library:** It includes Assemblies. 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.

**2) Predefined class library:** It contains 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 a system, where using represents a keyword and the system represents a namespace.

**55) Explain 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.

**56) Explain Common Language Specification (CLS)?**

\* Common Language Specification (CLS) is a subset of CTS and defines a set of rules and regulation 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 it own syntax. So when statement belonging to different languages get executed, a common platform will be provided be provided by the CLS to interact and share the information.

**57) 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, 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 program and applications.

**58) What is Just-In-Time (JIT) Compiler in .NET?**

\* Just-In-Time compiler (JIT) is a part of Common Language Runtime (CLR) in .NET which is responsible for managing the execution of .NET program regardless of any .NET programming language. A language-specific compiler converts the source code to the intermediate language. This intermediate language is then converted into the machine code by the Just-In-Time (JIT) compiler. This machine code is specific to the computer environment that the JIT compiler runs on.

**59) 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 instruction provided by MSIL is platform-independent and is generated by the language-specific compiler from the source code. JIT compilers compile the MSIL into machine code based on the requirement.

**60) Is ASP.NET different from ASP?**

**\*** 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 not built-in-support for XML.
* ASP.NET uses the ADO.NET technology to connect and work with databases, ASP uses ADO technology.

**61) What is the order of the events in a page life cycle?**

\* Page\_PreInt \* Page\_PreLoad \* Page\_PreRender

\* Page\_Init \* Page\_Load \* Render

\* Page\_InitComplete \* Page\_LoadComplete

**62) What is catching?**

\* Catching 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.