**What is C#?**

C# is the best language for writing Microsoft .NET applications. C# provides the rapid application development found in Visual Basic with the power of C++. Its syntax is similar to C++ syntax and meets 100% of the requirements of OOPs like the following:

•Abstraction

•Encapsulation

•Polymorphism

•Inheritance

**What is Managed or Unmanaged Code?**

Managed Code

“The code, which is developed in .NET framework is known as managed code. This code is directly executed by CLR with the help of managed code execution. Any language that is written in .NET Framework is managed code”.

Unmanaged Code

The code, which is developed outside .NET framework is known as unmanaged code.

**What is Boxing and Unboxing?**

Boxing and Unboxing both are used for type conversion but have some difference:

Boxing:

Boxing is the process of converting a value type data type to the object or to any interface data type which is implemented by this value type.

Unboxing:

Unboxing is also a process which is used to extract the value type from the object or any implemented interface type.

**What is enum in C#?**

An enum is a value type with a set of related named constants often referred to as an enumerator list. The enum keyword is used to declare an enumeration

**What is the difference between constant and read only in c#?**

Constant (const) and Readonly (readonly) both looks like same as per the uses but they have some differences:

Constant is known as “const” keyword in C# which is also known immutable values which are known at compile time and do not change their values at run time like in any function or constructor for the life of application till the application is running.

Readonly is known as “readonly” keyword in C# which is also known immutable values and are known at compile and run time and do not change their values at run time like in any function for the life of application till the application is running. You can assay their value by constructor when we call constructor with “new” keyword.

**Define Property in C#.net?**

Properties are members that provide a flexible mechanism to read, write or compute the values of private fields, in other words by the property we can access private fields. In other words we can say that a property is a return type function/method with one parameter or without a parameter.

**What is the difference between dispose and finalize methods in c#?**

Finalizer and dispose both are used for same task like to free unmanaged resources but have some differences see.

Finalize:

•Finalize used to free unmanaged resources those are not in use like files, database connections in application domain and more, held by an object before that object is destroyed.

•In the Internal process it is called by Garbage Collector and can’t called manual by user code or any service.

•Finalize belongs to System. Object class.

•Implement it when you have unmanaged resources in your code, and make sure that these resources are freed when the Garbage collection happens.

Dispose:

•Dispose is also used to free unmanaged resources those are not in use like files, database connections in Application domain at any time.

•Dispose explicitly it is called by manual user code.

•If we need to dispose method so must implement that class by IDisposable interface.

•It belongs to IDisposable interface.

•Implement this when you are writing a custom class that will be used by other users.

**What is delegates in C# and uses of delegates?**

C# delegates are same as pointers to functions, in C or C++. A delegate Object is a reference type variable that use to holds the reference to a method.

**What is sealed class in c#?**

Sealed classes are used to restrict the inheritance feature of object oriented programming. Once a class is defined as a sealed class, the class cannot be inherited.

**What are partial classes?**

A partial class is only use to splits the definition of a class in two or more classes in a same source code file or more than one source files. What is Singleton Design Pattern?

1. Ensures a class has only one instance and provides a global point of access to it.

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

3. Most commonly, singletons don't allow any parameters to be specified when creating the instance, since a second request of an instance with a different parameter could be problematic! (If the same instance should be accessed for all requests with the same parameter then the factory pattern is more appropriate.)

4. There are various ways to implement the Singleton Pattern in C#. The following are the common characteristics of a Singleton Pattern.

**What is Serialization?**

Serialization means saving the state of your object to secondary memory, such as a file.

**What is LINQ in C#?**

LINQ stands for Language Integrated Query. LINQ is a data querying methodology which provides querying capabilities to .NET languages with a syntax similar to a SQL query

**What is Reflection in C#.Net?**

Reflection typically is the process of runtime type discovery to inspect metadata, CIL code, late binding and self-generating code.