What is Managed or Unmanaged Code?

**Managed Code**

“Managed code is the code that is developed using the .NET framework and its supported programming languages such as C# or VB.NET. Managed code is directly executed by the Common Language Runtime (CLR or Runtime) and its lifecycle including object creation, memory allocation, and object disposal is managed by the Runtime. Any language that is written in .NET Framework is managed code".

**Unmanaged Code**

The code that is developed outside of the .NET framework is known as unmanaged code.

## What is Boxing and Unboxing in C#?

The process of converting from a value type to a reference type is called boxing. Boxing is an implicit conversion. Here is an example of boxing in C#.

The process of converting from a reference type to a value type is called unboxing. Here is an example of unboxing in C#.

What is the difference between a struct and a class in C#?

**Struct**

* The struct is a value type in C# and it inherits from System.Value Type.
* Struct is usually used for smaller amounts of data.
* Struct can’t be inherited from other types.
* A structure can't be abstract.
* No need to create an object with a new keyword.
* Do not have permission to create any default constructor.

**Class**

* The class is a reference type in C# and it inherits from the System.Object Type.
* Classes are usually used for large amounts of data.
* Classes can be inherited from other classes.
* A class can be an abstract type.
* We can create a default constructor.

## What is the difference between Interface and Abstract Class in C#?

* A class can implement any number of interfaces but a subclass can at most use only one abstract class.
* An abstract class can have non-abstract methods (concrete methods) while in case of interface, all the methods have to be abstract.
* An abstract class can declare or use any variables while an interface is not allowed to do so.

## What is the difference between constant and readonly in C#?

Const is nothing but "constant", a variable of which the value is constant but at compile time. It's mandatory to assign a value to it. By default, a const is static and we cannot change the value of a const variable throughout the entire program.

Readonly is the keyword whose value we can change during runtime or we can assign it at run time but only through the non-static constructor.

## 

## Can “this” be used within a static method?

We can't use 'this' in a static method because the keyword 'this' returns a reference to the current instance of the class containing it.

What are extension methods in C#?

Extension methods enable you to add methods to existing types without creating a new derived type, recompiling, or otherwise modifying the original type.

What is the difference between String and StringBuilder in C#?

**String**

* It’s an immutable object that holds a string value.
* Performance-wise, string is slow because it creates a new instance to override or change the previous value.
* String belongs to the System namespace.

**StringBuilder**

* StringBuilder is a mutable object.
* Performance-wise StringBuilder is very fast because it will use the same instance of StringBuilder object to perform any operation like inserting a value in the existing string.
* StringBuilder belongs to System.Text.Stringbuilder namespace.

**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 that is used to extract the value type from the object or any implemented interface type

**"is" operator**

In C# language, we use the "is" operator to check the object type. If two objects are of the same type, it returns true, else it returns false.

**"as" operator**

The "as" operator behaves in a similar way as the "is" operator. The only difference is it returns the object if both are compatible with that type. Else it returns a nu

