.NET and C# Interview Questions and Answers

What is .NET?

.NET is a software development platform developed by Microsoft. It includes a large class library named Framework Class Library (FCL) and provides language interoperability across several programming languages.

Is there a full form of .NET?

No, .NET is not an acronym, so it does not have a full form. It is a brand name given by Microsoft.

What is the .NET framework?

The .NET Framework is a software development framework for building and running applications on Windows. It includes the Common Language Runtime (CLR) and the .NET Framework Class Library.

What is CLR (Common Language Runtime)?

CLR is the virtual machine component of the .NET framework that manages the execution of .NET programs. It provides services such as memory management, type safety, exception handling, garbage collection, and security.

What is MSIL code?

Microsoft Intermediate Language (MSIL) is the CPU-independent set of instructions that can be converted to native code. It is the output of compiling .NET programs and is converted to native code by the CLR.

What is an assembly and its types?

An assembly is a compiled code library used for deployment, versioning, and security in .NET. It can be a DLL or an EXE. There are two types: private assemblies and shared assemblies.

What is the difference between .NET Core and the .NET Framework?

.NET Core is a cross-platform, open-source framework for developing applications, while the .NET Framework is a Windows-only version. .NET Core provides better performance and is the foundation for the future of .NET.

What is the latest stable version of .NET?

As of June 2024, the latest stable version of .NET is .NET 7.

What is CTS (Common Type System)?

CTS is a standard that defines how data types are declared, used, and managed in the .NET runtime, ensuring that objects written in different .NET languages can interact with each other.

What is VES (Virtual Execution System)?

VES, part of the CLI, provides an environment for executing managed code. It handles the loading and execution of programs written in intermediate language and provides services such as memory management and security.

Is .NET platform-independent or platform-dependent?

.NET Core is platform-independent, supporting Windows, macOS, and Linux, while the original .NET Framework is platform-dependent and works only on Windows.

Is C# an object-oriented language?

Yes, C# is a fully object-oriented programming language developed by Microsoft as part of its .NET initiative.

What are the similarities and differences between C, C++, and C#?

C is a procedural programming language, C++ is an extension of C with object-oriented features, and C# is a modern, object-oriented language designed for the .NET framework. Unlike C and C++, C# is managed code and provides features like garbage collection.

What are the differences between C# and Java?

C# and Java are both high-level, object-oriented languages with similar syntax. However, C# is designed for the .NET framework and includes features like properties and events, while Java is platform-independent with its JVM and has wider platform support.

How does the C# programming language work?

C# code is compiled into Intermediate Language (IL), which is then converted into native machine code by the CLR. This managed execution process provides services like garbage collection, type safety, and exception handling.

What methods are used for data type conversion in C#?

C# provides several methods for data type conversion, such as Convert.ToInt32, Convert.ToString, Parse, and TryParse. These methods help convert between different data types safely and efficiently.

What is the difference between Convert.ToInt32 and int.Parse methods?

Convert.ToInt32 converts a specified value to a 32-bit integer, handling null and empty values, while int.Parse converts a string to a 32-bit integer but throws an exception if the string is null or not a valid number.

What is garbage collection in C#, and how does it work?

Garbage collection in C# is an automatic memory management feature. It identifies and frees up memory occupied by objects that are no longer in use, ensuring efficient use of memory.

How is memory management handled in C#?

Memory management in C# is primarily handled by the CLR through garbage collection. It automatically allocates and releases memory for objects, reducing the risk of memory leaks and fragmentation.

What is boxing and unboxing in C#?

Boxing is the process of converting a value type to an object type, while unboxing is the reverse process, converting an object type back to a value type. These operations allow value types to be treated as objects.