**What is the .NET framework?**

The .NET framework supports an object-oriented approach that is used for building applications on windows. It supports various languages like C#, VB, Cobol, Perl, .NET, etc. It has a wide variety of tools and functionalities like class, library and APIs that are used to build, deploy and run web services and different applications.

**What are the different components of .NET?**

Following are the components of .NET

* Common Language run-time
* Application Domain
* Common Type System
* .NET Class Library
* .NET Framework
* Profiling

**What do you know about CTS?**

* CTS stands for Common Type System. It follows certain rules according to which a data type should be declared and used in the program code. CTS also describes the data types that are going to be used in the application. We can even make our own classes and functions following the rules in the CTS, it helps in calling the data type declared in one program language by other programming languages.

**What is CLR?**

CLR stands for common language run-time, it is an important component of the .NET framework. We can use CLR as a building block of various applications and provides a secure execution environment for applications.

Whenever an application written in C# is compiled, the code is converted into an intermediate language. After this, the code is targeted to CLR which then performs several operations like memory management, security checks, loading assemblies, and thread management.

**Explain CLS.**

Common language specification helps the developers to use the components that are inter-language compatible with certain rules that come with CLS. It then helps in reusing the code in other .NET compatible languages.

**What do you know about JIT?**

JIT is a compiler which stands for Just In Time. It is used to convert the intermediate code into the native language. During the execution, the intermediate code is converted into the native language.

**What is the difference between Response.Redirect and Server.Transfer?**

Response.Redirect basically redirects the user’s browser to another page or site. The history of the user’s browser is updated to reflect the new address as well. It also performs a trip back to the client where the client’s browser is redirected to the new page.

Whereas, Server.Transfer transfers from one page to the other without making any round-trip back to the client’s browser. The history does not get updated in the case of Server.Transfer.

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

|  |  |
| --- | --- |
| **Managed code** | **Unmanaged code** |
| Managed code is managed by **CLR** | Any code that is not managed by **CLR** |
| .NET framework is necessary to execute managed code | Independent of .NET framework |
| **CLR**manages memory management through garbage collection | Own runtime environment for compilation and execution |
| Eg : Code written in AS.NET Language | Eg : Code written with C++ Language |

**What do you know about boxing and unboxing?**

|  |  |
| --- | --- |
| **Boxing** | **Unboxing** |
| Implicit | Explicit |
| Converting a value type to the type object | Extracting the value type from the object |
| eg : obj myObject = i; | eg : i = (int)myObject; |

**What is BCL?**

* BCL is a base class library of classes, interfaces and value types
* It is the foundation of .NET framework applications, components, and controls
* Encapsulates a huge number of common functions and make them easily available for the developers
* It provides functionality like threading, input/output, security, diagnostics, resources, globalization, etc.
* Also serves the purpose of interaction between user and runtime
* It also provides namespaces that are used very frequently. for eg: system, system.Activities, etc.