**.Net Framework Part 1**

**1. What is mean by .Net Framework?**

The .NET framework is a collection of all the tools and utilities required to execute the .NET managed applications on a particular platform.

**2. What is mean by CLR?**

Common Language Runtime is the core component of .Net framework. It is similar to the Java Virtual Machine or JVM in Java, which handles the execution of code and provides useful services for the implementation of the program. It provides a number of services, including the following

* management (loading and execution)
* Application memory isolation
* Verification of type safety
* Conversion of IL to native code
* Access to metadata (enhanced type information)
* Managing memory for managed objects
* Enforcement of code access security
* Exception handling, including cross-language exceptions
* Interoperation between managed code, COM objects, and pre-existing DLLs (unmanaged code and data)
* Automation of object layout
* Support for developer services (profiling, debugging, and so on)

**3. What is difference between managed and unmanaged code?**

The managed code is always executed by a managed runtime execution environment like CLR for .Net. Metadata information of the code will be exchanged with runtime, so runtime environment can guarantee what the code is going to do and provide the necessary security checks before executing any piece of code

Code that is directly executed by the Operating System is known as un-managed code. Example applications written in VB 6.0, C++, C, etc are unmanaged code that typically targets the processor architecture and is always dependent on the computer architecture. In unmanaged code the memory allocation, type safety, security, etc needs to be taken care of by the developer.

**4. What is mean by MSIL?**

MSIL or IL stands for Microsoft Intermediate Language; if you compile managed code, the compiler translates your source code into Microsoft intermediate language. MSIL is platform independent language which can be converted to native code while installing software or at runtime by using Just-in compiler.

**5. What is mean by CTS?**

Common type system defines how types are declared, used, and managed in the runtime, and is also an important part of the runtime's support for cross-language integration. CTS is responsible for defining types that can be used across the .Net Languages. CTS Provides the data types, values, object types. This helps developers to develop applications in different languages.

For example, an integer variable in C# is written as int, whereas in VB.Net it is written as integer. Therefore in .Net Framework you have single class called System.Int32 to interpret these variables.

**6. What is mean by CLS?**

Common Language Specification is the subset of CTS; it is specification that defines the set rules and guidelines that all supporting language should follow. It integrate in such a way that programs written in any language can interoperate with one another.