**Dot Net Framework**

.NET framework is a software framework primarily for Microsoft Windows. It includes a large library & provides language interoperability across several programming languages. .NET framework is a platform that provides you environment which you can use to development different kinds of application such as web application, distributed application, web service etc.

.NET framework consists of four major parts:

Application Development Technology - This includes different application technologies such as ASP.NET, WinForms, Web Service etc.

Class Libraries - This includes extensive set of class libraries such as Data Access, Regular Expression, XML, Queuing Support, Directory Service etc.

Base Class Library - This is huge class libraries which include Collections, Thread, Code Generation, IO, Reflection and security.

Common Language Runtime - This is the runtime environment of .NET Framework which is responsible to execute the managed code on machine.

**Portability**

The Dot Net framework allows platform-agnostic & cross-platform implementations for other OS’. The availability of specifications for the CLI, & C# make it possible for third parties to create compatible implementations of the framework & it’s languages on other platforms.

CLR is present for each and every OS and will act a layer above the OS and manages the execution of the .Net applications is understandable by the CLR and the CLR in each OS will handle it.

**MSIL**

When we compile our .Net Program using any .Net compliant language like c#, VB.Net it gets converted to an intermediate code. In case of other languages like java compilation compilation will create an executable binary code. This intermediate code is called MSIL or IL in short. This intermediate code is understandable by CLR.MSIL is OS and hardware independent code. When the program needs to be executed this MSIL is converted into binary executable code, called native code. The presence of IL makes it possible the Cross Language Relationship as all the .Net compliant languages produce the similar standard IL code.

**CLR**

The most important component of .Net framework is the .Net Common Language Runtime(CLR), also called .Net Runtime. It is a framework layer that resides above the Operating System and handles/manages the execution of the .NET applications. Our .Net programs don't directly communicate with the Operating System but through CLR.

**JIT (Just In Time) compilers**

When the MSIL code needs to be executed, JIT compilers comes into picture.CLR invokes JIT compilers to compile the IL code to native executable code (dll or .exe) for the specific machine and the OS.JIT is different from traditional compilers as it compiles the IL code only when it is needed, ie when the function call happens. When the function is called the IL of function’s body is converted to native code, just in time of need. The part of code which is not used in run is not converted to native code. If the same function is called again, CLR uses the same copy of native code without re-compiling.

**GAC**

Each computer where the common language runtime is installed has a machine-wide code cache called the global assembly cache.GAC is a repository of all the .net shared assembly. The assemblies stored in GAC can be shared by all the .NET application.

When there is a need we should share assemblies by installing them into the GAC.There are two ways to deploy an assembly into the GAC;

* Use an installer designed to work with the global assembly cache. This is the preferred option for installing assemblies into the global assembly cache.
* Use a developer tool called the Global Assembly Cache tool (Gacutil.exe), provided by the Windows Software Development Kit (SDK).

**DLL and .EXE**

DLL – Dynamic Link Library. Many dll exists in one application.dll can be shared with other applications.

EXE- is an executable file. Exe can run on its own. Only one exe file exists for one application.

Exe cannot be shared with other application.