

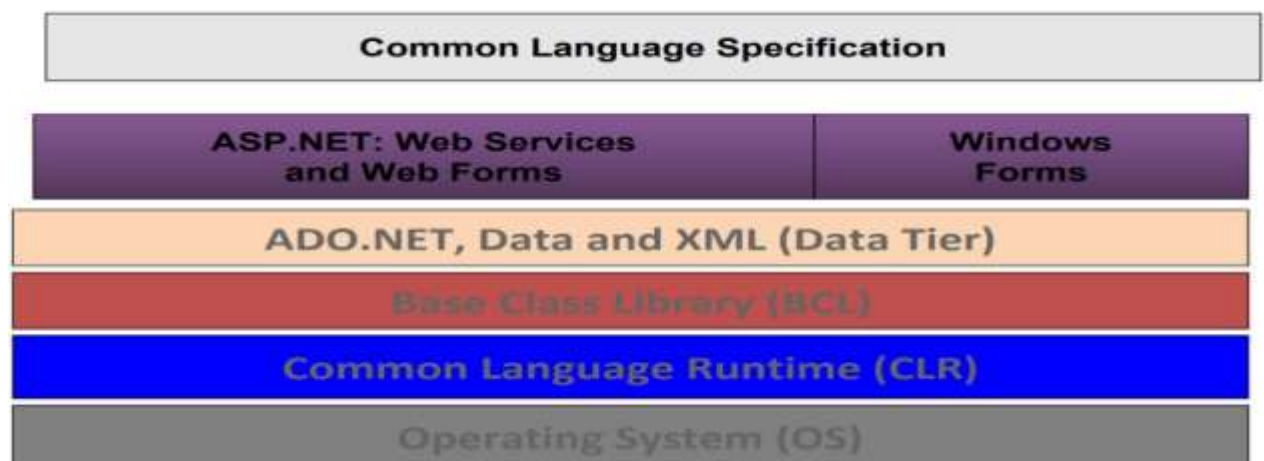
Enrolment number: 22032211123

Name: Parmar Karan

Subject: GUI Programming

1) Explain CLR with its parts.

Since there are numerous different languages, .NET has specified those commonalities in something called the Common Language Specification (CLS). CLS defines a set of features that are needed by many common applications.



Common Language Specification (CLS) is a set of rules and guidelines that govern the way code is written in different .NET languages to ensure interoperability. It's important to understand that CLS is not specific to Visual C# or WinForms, but rather it applies to all languages that target the .NET Framework.

Here are some key aspects of the CLS within the .NET Framework:

Data Types: CLS-compliant languages must support a common set of data types defined in the Common Type System (CTS). This includes basic types like integers, floating-point numbers, characters, and more.

Accessibility: CLS-compliant languages must support the accessibility levels defined in the .NET Framework, such as public, private, protected, and internal. This ensures that code written in one language can access members written in another language if they are properly exposed.

Error Handling: CLS-compliant languages must support structured exception handling mechanisms, such as try-catch-finally blocks. This ensures consistent error handling across different languages.

2) What is .NET? Explain its features.

- .Net Framework is a software development platform developed by Microsoft for building and running Windows applications. The .Net framework consists of developer tools, programming languages, and libraries to build desktop and web applications. It is also used to build websites, web services, and games.
- The .Net framework was meant to create applications, which would run on the Windows Platform. The first version of the .Net framework was released in the year 2002.
- The Microsoft .Net framework can be used to create both – Form-based and Web-based applications. Web services can also be developed using the .Net framework.
- The framework also supports various programming languages such as Visual Basic, C# , C++, and F#. So developers can choose and select the language to develop the required application. C# and visual basic are the main languages used in .Net framework.

=> **features**

- **Interoperability:**

“Interoperability is the property that enables different programming languages to interact within the same system”.

The .Net framework provides a lot of backward support. Suppose if you had an application built on an older version of the .Net framework, say 2.0. And if you tried to run the same application on a machine which had the higher version of the .Net framework, say 3.5. The application would still work. This is because with every release, Microsoft ensures that older framework versions gel well with the latest version.

- **Common Language Runtime engine (CLR):**

The “Common Language Infrastructure” or CLI is a platform in .Net architecture on which the .Net programs are executed.

The Common Language Runtime (CLR) is programming that manages the execution of programs written in any of several supported languages, allowing them to share common object-oriented classes written in any of the languages. It is a part of Microsoft's .NET Framework.

- **Language independence:**

.NET is language independent. This means that, as a developer, you can develop in one of the many languages that target .NET implementations, such as C#, F#, and Visual Basic. You can access the types and members of class libraries developed for .NET

- **Base Class Library:**

.NET Base Class Library is the sub part of the Framework that provides library support to Common Language Runtime to work properly. It includes the System namespace and core types of the .NET framework.

- **Simplified deployment:**

When you're ready to publish your app to a web server or other centralized location so that users can install it, you can choose from several deployment methods. Some of these are provided with Visual Studio.

- **Security:**

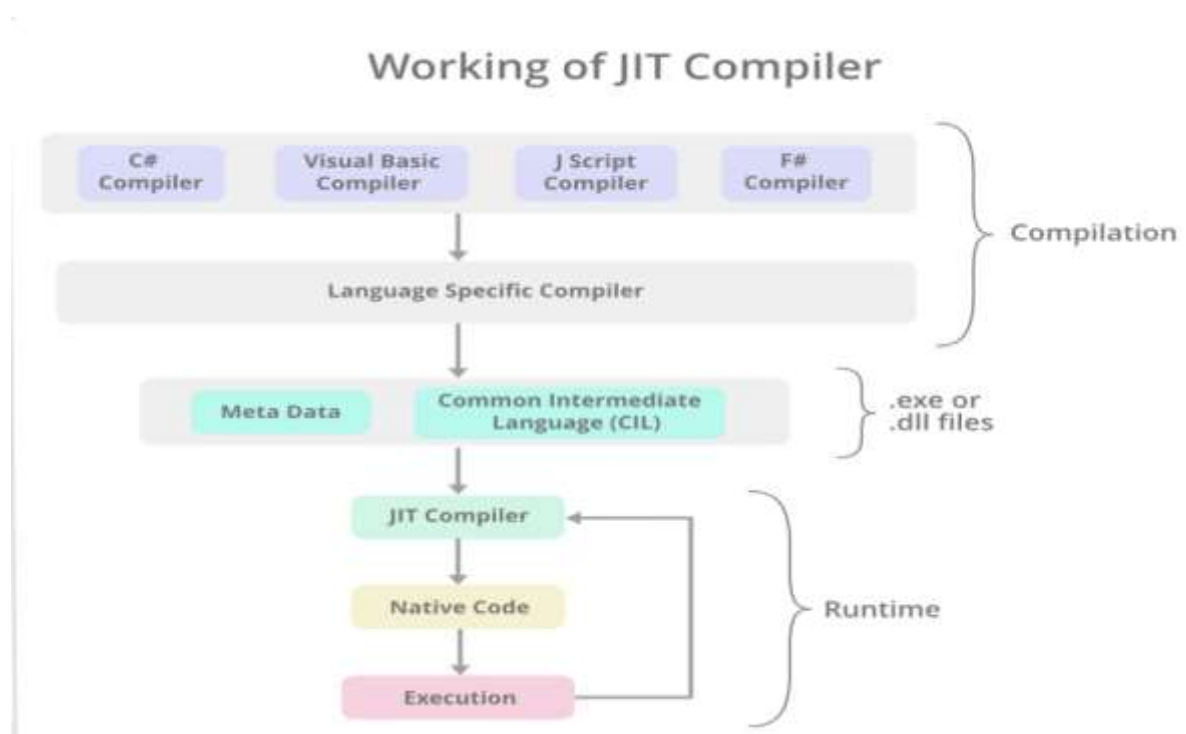
The common language runtime and .NET provide many useful classes and services that enable developers to write secure code, use cryptography, and implement role-based security.

- **Portability:**

portability is the possibility to use the same software in different environments. It applies to the software that is available for two or more different platforms or can be recompiled for them.

3) Define JIT.

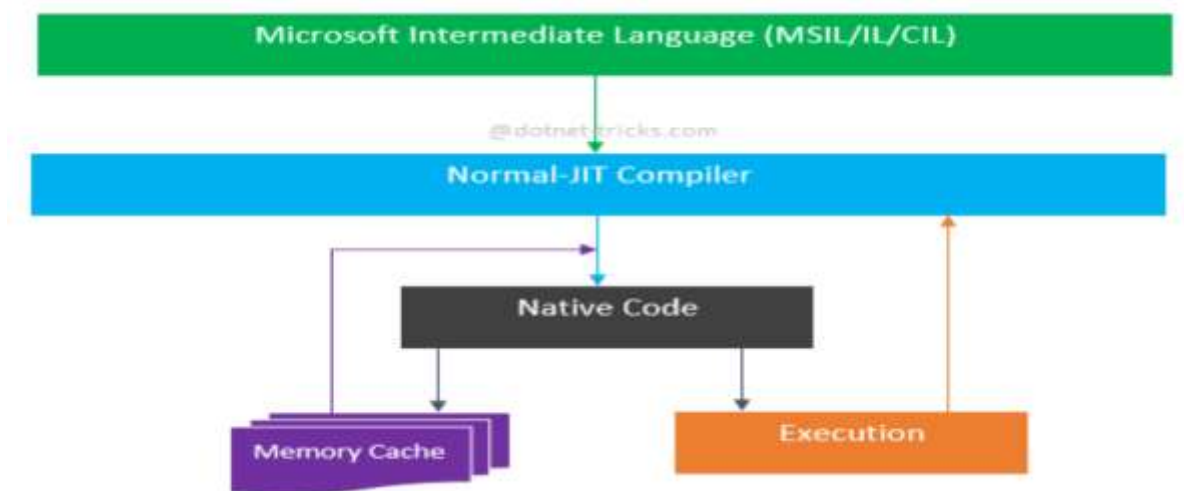
- JIT stands for just-in-time compiler. It converts the MSIL code to CPU native code as it is needed during code execution. It is called just-in-time since it converts the MSIL code to CPU native code; when it is required within code execution otherwise it will not do nothing with that MSIL code.
- JIT compilers (as their name suggests) use MSIL code, which is independent of the machine, operating system, and CPU. Several JIT compilers exist, each targeting a different architecture, and the appropriate one will be used to create the native code required.



- **JIT are of three types:**

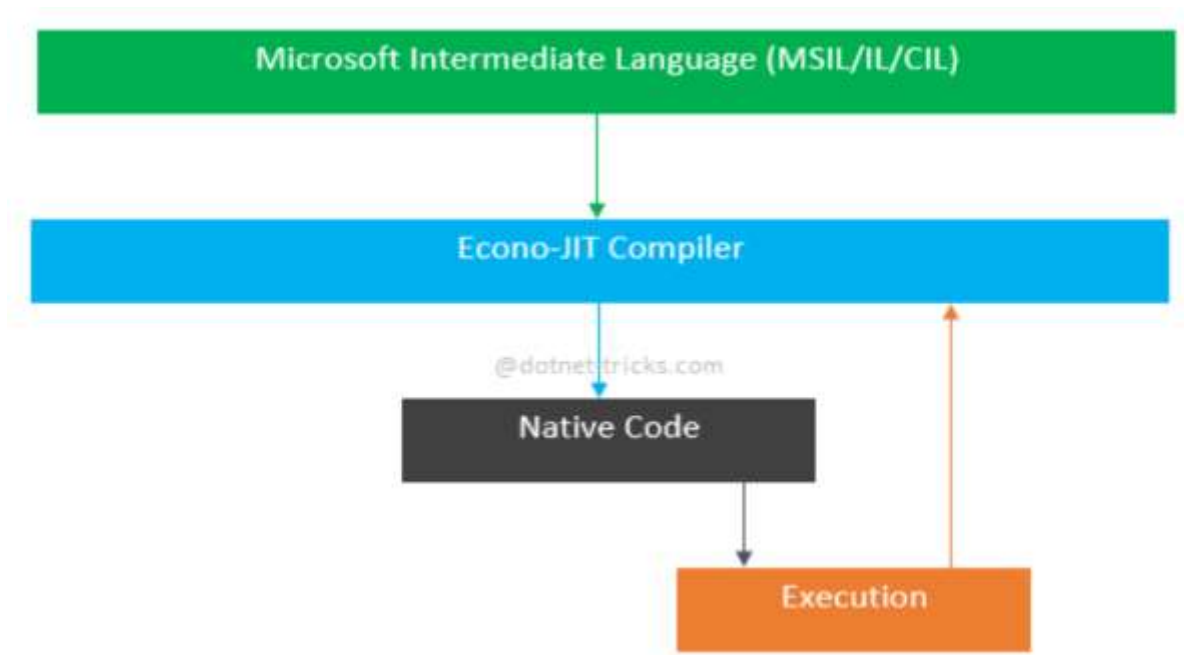
- 1) Normal JIT:

It will only convert the called code and will store in cache so that it will not require converting code again. Normal JIT is fast.



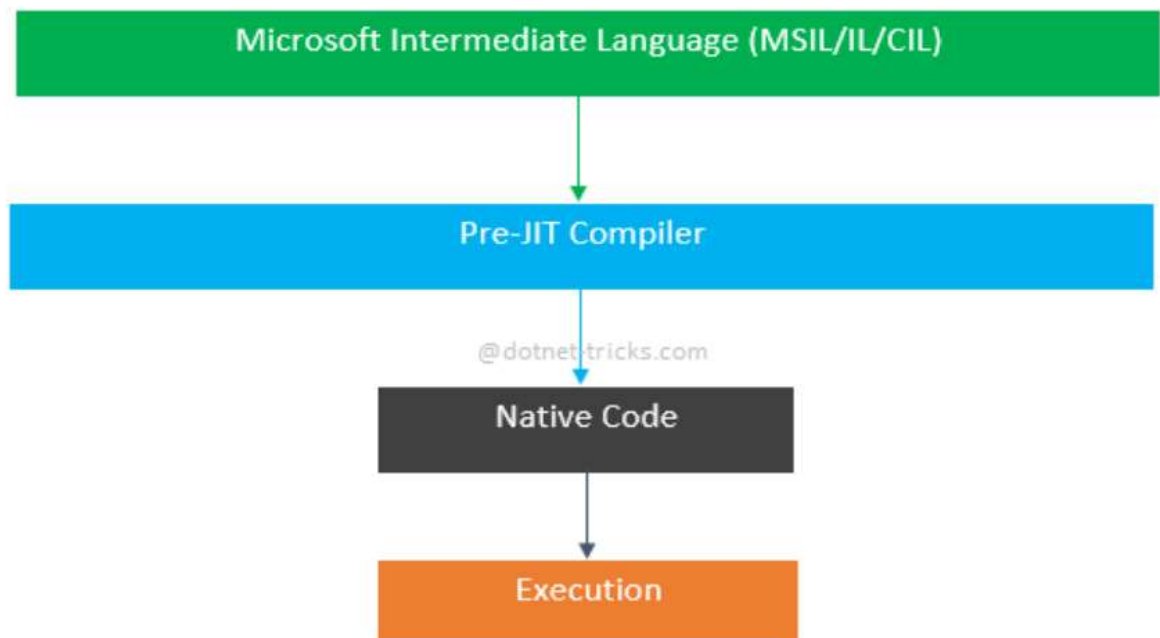
- 2) Econo JIT Compilation:

It will convert the called executable code only. But it will convert code every time when a code is called again.



- 3) Pre-JIT Compilation

It converts all the code in executable code and it is slow.



4)Assembly v/s Namespace.

Sno.	Assembly	Namespace
1	A .NET assembly provides a fundamental unit to physical code grouping.	A .NET namespace provides a fundamental unit of logical code grouping.
2	Assemblies are your DLL and EXE files.	Namespaces are just the dot notation that'll help you find your classes as you write code.
3	Scope for a particular type is defined at run time using Assembly.	Namespace is the logical naming decided at design time by the developer.
4	Assembly contains code of the form MSIL (Microsoft Intermediate Language)	Namespace contains set of unique names.
5	Logical units are physically grouped together as assembly.	Classes available in your program will be logically grouped together under a namespace.
6	An assembly can contain types belonging to different namespaces.	Namespace can include multiple assemblies.
7	Assembly can be classified as private assembly and public assembly. Private assembly is specific to a single application but shared/public assembly contains libraries which can be used by multiple applications.	Namespace doesn't have any classification.

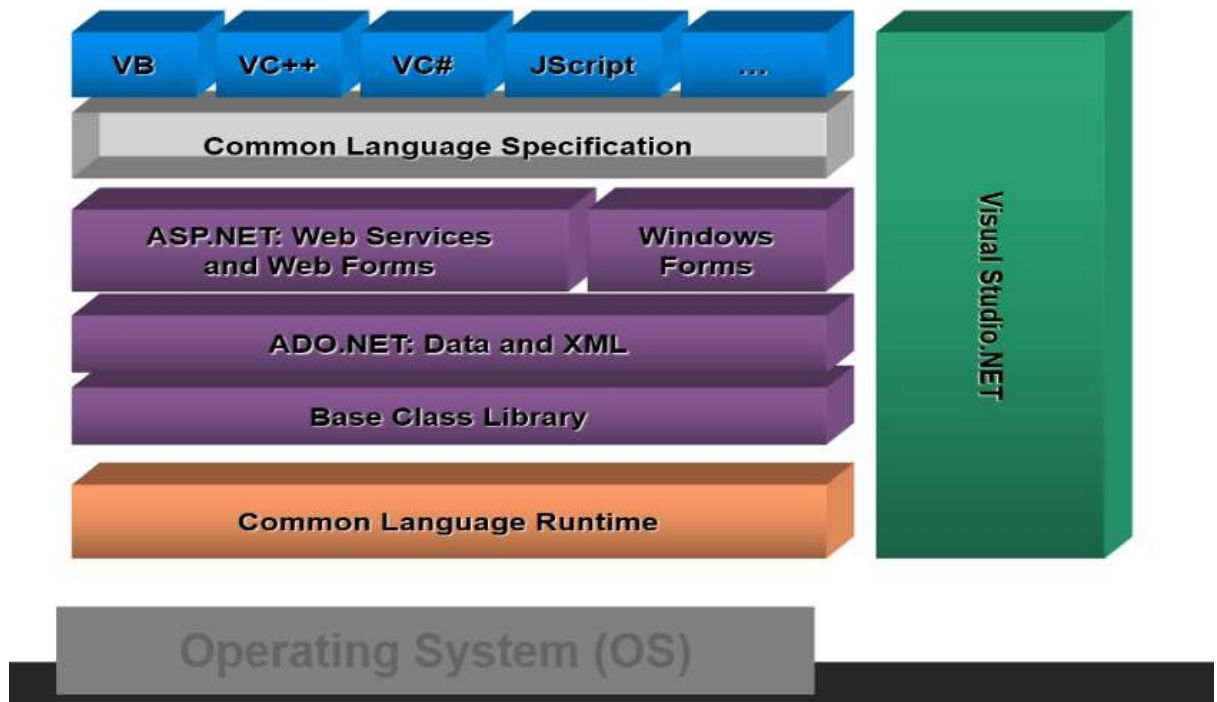
8	Assemblies need not be explicitly specified. They are automatically described in metadata and manifest files.	Namespaces have to be mentioned in Project-Properties.
9	Such nesting is not permissible in assemblies.	Namespaces can be nested. For example: <pre>namespace sampleApp1 { namespace SampleApp2 { class sampleClass { ... } } }</pre>

5) What is .NET? Explain Architecture of it.

- The .NET Framework Architecture is an extension to the Microsoft .NET platform that provides a set of common programming interface elements and services. This platform is designed to enable software developers to build robust, reliable, high-quality applications that can run on any platform that supports the .NET Framework Architecture..

The OS manages the resources, the processes and the users of the machine

- It is Provides to the applications some services (threads, I/O, GDI+, DirectX, COM, COM+, MSMQ, IIS, WMI, ...)
- CLR is a separate process in the OS

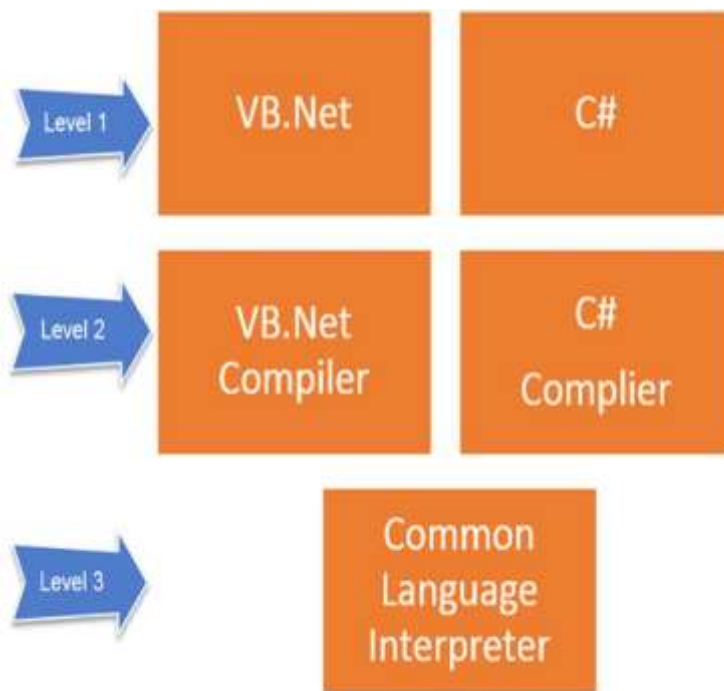


- **CLR:**

CLR sits on top of OS to provide a virtual environment for hosting managed applications CLR loads modules containing executable and executes their code. Code might be managed or unmanaged

Exception Handling , Garbage Collection ,Manages the memory ,security.





- **Base Class Library:**

Rich object-oriented library with fundamental classes

Input-output, collections, text processing, networking, security, multi-threading, ...

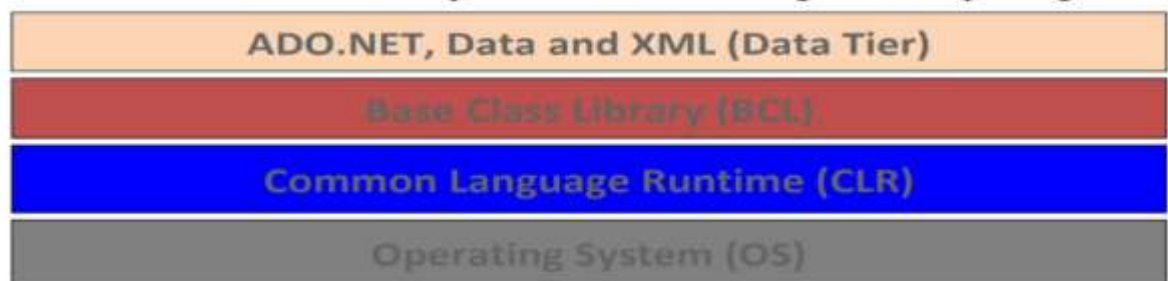


- **ADO.NET, XML: ActiveX Data Objects**

ADO.NET provides consistent access to data sources such as SQL Server and XML, and to data sources exposed through OLE DB and ODBC. Data-sharing consumer applications can use

ADO.NET to connect to these data sources and retrieve, handle, and update the data that they contain.

XML is a software- and hardware-independent tool for storing and transporting data.



- **Web Services and Web form / Windows Forms:**

In this part developer can create a GUI based interface using Web Forms, and Window Forms.

With the use of this service we can design and create web page and desktop page.



CLS:

Since there are numerous different languages, .NET has specified those commonalities in something called the Common Language Specification (CLS). CLS defines a set of features that are needed by many common applications.

