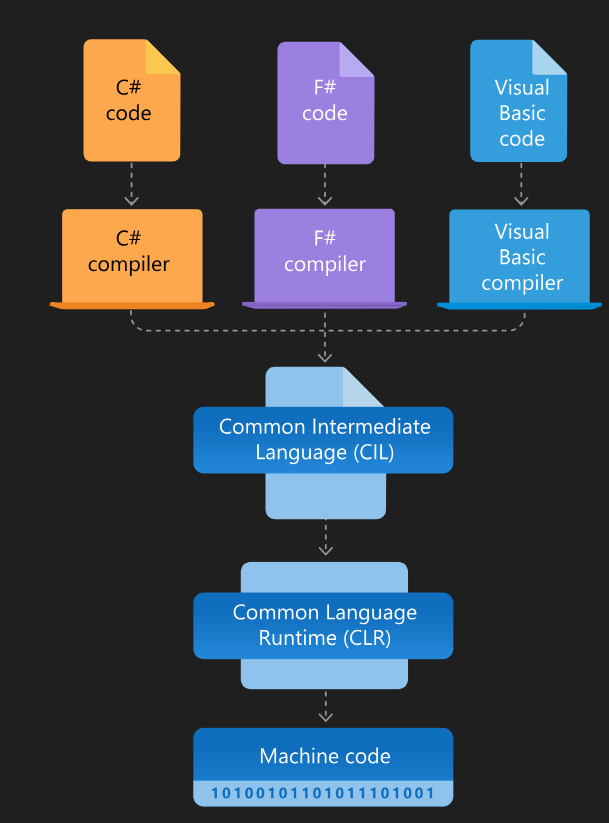
.**NET**

* The first version of .NET is known as the .NET Framework which needs to be deployed to the target computer in its entirely and can only run on Windows platforms. But now, .NET has evolved into an environment that can run on multiple platforms like Windows, Mac OS, Linux, IOS, Android and more.
* The .NET environment became fragmented in 2016 with the release of .NET Core, which enabled .NET to be cross platform and agile in the sense that only your application’s base class library dependencies need to be deployed to the target computer with your application.
* Then in 2020, .NET became unified with the release of NET 5, which meant that the confusion created by having two strands of .NET, namely .NET Framework and .NET Core, was alleviated
* The latest stable release of C# is a highly evolved, sophisticated programming language that allows you to create almost any kind of application that can run on multiple platforms. You can create a single code base that can run on multiple platforms, for example Linux, Mac OS, Android, IOS, in the Cloud, of course Windows operating systems and more.
* You are able to write and build your C# applications using free tools like Visual Studio 2022 Community edition or the cross platform, light weight tool, Visual Studio Code. Visual Studio Code can run on Windows, Mac OS, and Linux platforms.
* C# is a highly versatile programming language. You can build many types of applications, such as web-based applications using ASP .NET, cross platform mobile and desktop applications using the .Net MAUI framework, Internet of things applications, AI applications using ML.NET, cloud native applications, games and more.
* C# has a huge support base, backed by Microsoft, and is constantly evolving. A new version of .NET is shipped every November, which always contains many improvements and enhancements. This means that .NET is forever evolving, improving, and keeping up with the latest trends in technology.
* Actually .net ek collection of language hai jo microsoft ne develop kiya tha taaki har trah ki programming aap ek he jgh kr sako aur aapko sab features ek he platform pe mile
  + To agar jise c aati hai vo C# achhe se seekh sakta hai jise java aati hai use j# jise asp use asp.net to c#.j#,ASP.net ye sab .net platform ki languages hain
* .**NET Framework**: Ye actual runtime environments hain jinmein .NET applications execute ki jati hain. .NET Framework primarily Windows platform ke liye hai
* .**NET Core** : .NET Core cross-platform hai, matlab ki ise Windows, macOS, Linux jaise operating systems par bhi use kiya ja sakta hai.
* NET is a software development platform developed by Microsoft. In this platform, developers are provided tools, libraries, and frameworks to develop applications, web services, and other software.
* This platform is a set which includes components:
* **Common Language Runtime (CLR):** provides code execution, memory management, and other runtime services.
* **Base Class Library (BCL):** This is a collection of pre-defined classes and functions that helps developers perform common tasks.
* **Programming Languages**: Languages like C#, Visual Basic, F# are used in .NET platform. With the help of these languages developers write .NET applications.
* **Development Tools**: Microsoft Visual Studio is a major used for .NET development. Apart from this, command-line tools are also available for .NET development.
* Overall, the .NET platform is an ecosystem that provides developers with the necessary resources for application develop
* Visual studio ide hai jase java k liy eclipse but visual studio code ek editor hai

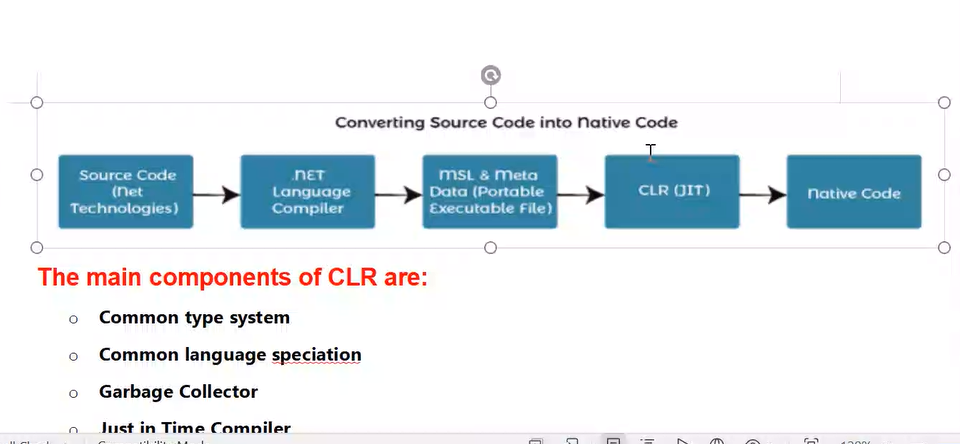


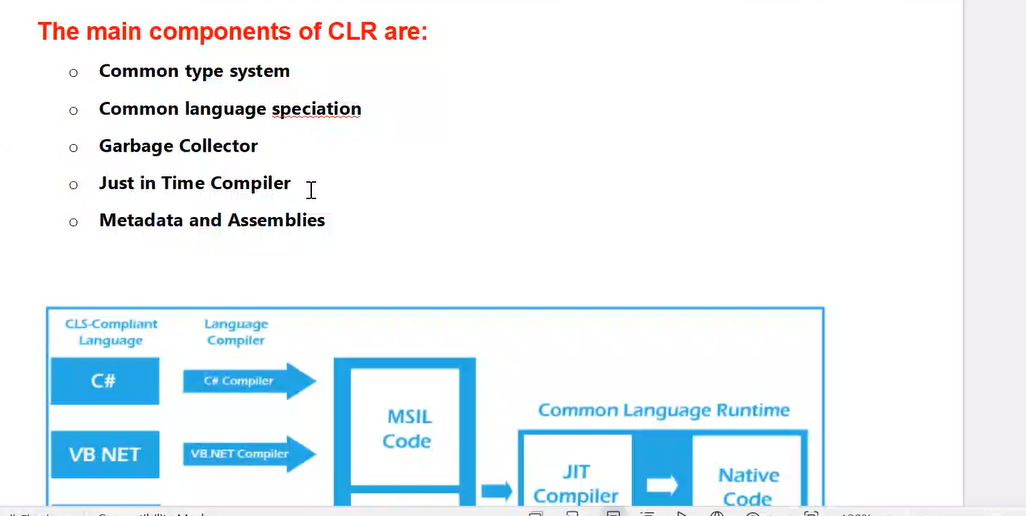
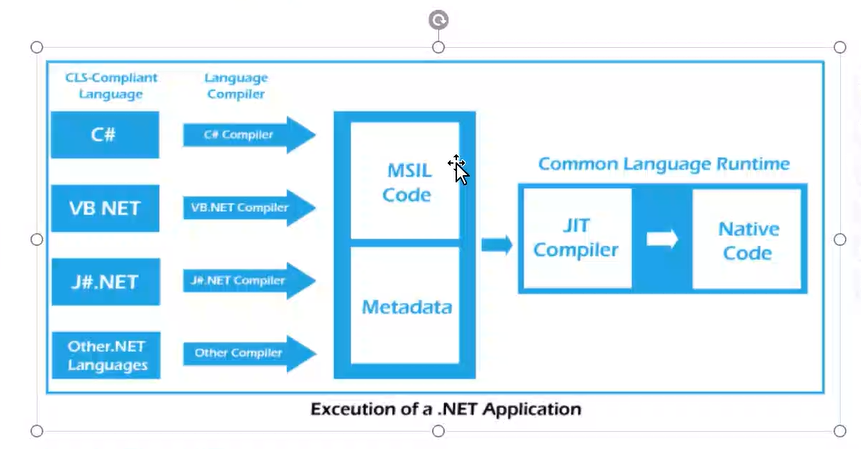
* <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet-framework>
* Ek .net application hai vo myultiple programmimh me likh sakte ho jaise ki ek module C# me bnaya ek f# me aur ek visual basic me fir cil usko ek intermediate language me convert krdega
* .NET Framework is a managed execution environment for Windows that provides a variety of services to its running apps. It consists of two major components:
  + the common language runtime (CLR), which is the execution engine that handles running apps,
  + .NET Framework Class Library, which provides a library of tested, reusable code that developers can call from their own apps

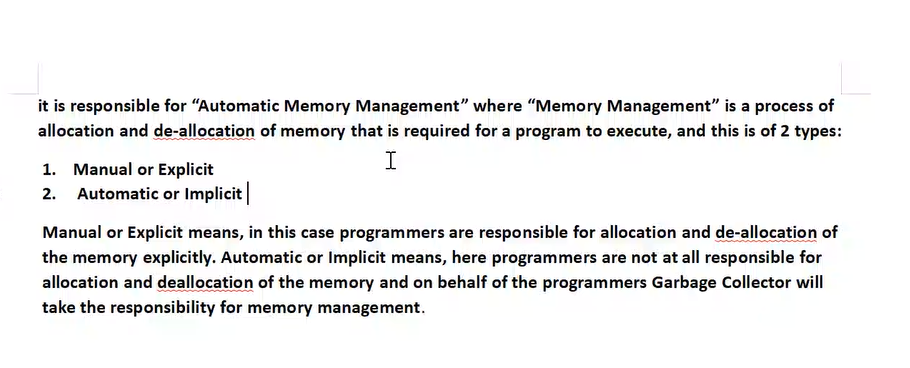
.net is

* Platform dependent
* Language dependent

CLR



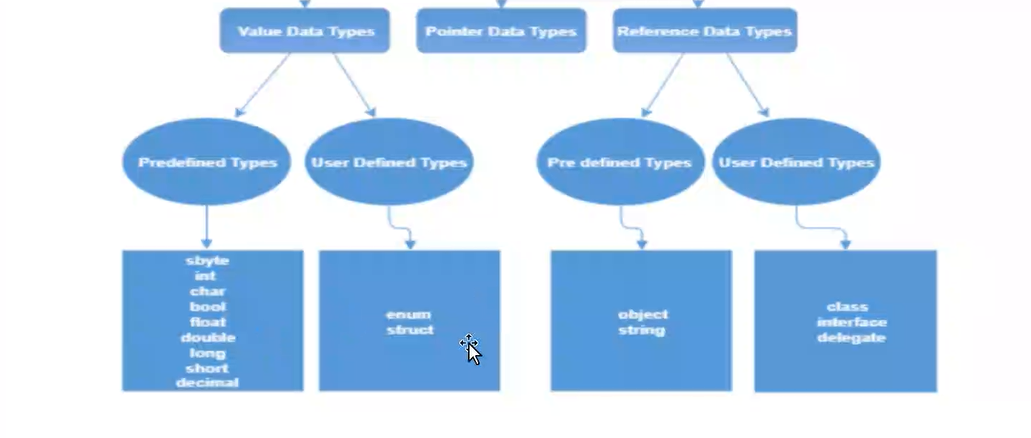




C# cannot be used without the .NET Framework or .NET Core. C# is a programming language developed by Microsoft specifically for the .NET platform. It's tightly integrated with the .NET runtime environment, which provides essential services such as memory management, type safety, and garbage collection.

When you write C# code, you're targeting the .NET runtime, which means your code will be compiled into Intermediate Language (IL) and executed by the Common Language Runtime (CLR) within the .NET framework or .NET Core runtime environment.

While there are other programming languages that can be used without the .NET framework, C# is intrinsically linked to it.



|  |  |
| --- | --- |
| **Managed Code** | **Unmanaged Code** |
| It is executed by managed runtime environment or managed by the CLR. | It is executed directly by the operating system. |
| It provides security to the application written in .NET Framework. | It does not provide any security to the application. |
| Memory buffer overflow does not occur. | Memory buffer overflow may occur. |
| It provide runtime services like Garbage Collection, exception handling, etc. | It does not provide runtime services like Garbage Collection, exception handling, etc. |
| The source code is compiled in the intermediate language known as IL or MSIL or CIL. | The source code directly compiles into native languages. |
| It does not provide low-level access to the programmer. | It provide low-level access to the programmer. |
| Memory is managed by CLR’s Garbage Collector. | Memory is managed by the programmer. |
| Performance is Slightly slower due to the overhead of memory management and JIT compilation. | Performance is Faster due to direct access to system resources and compiled machine-specific code. |
| Debugging is easier due to the availability of CLR’s debugging tools. | Debugging is harder due to lack of debugging tools. |
| Managed code requires installation of CLR on the target machine | Unmanaged code Can be distributed as a standalone executable or DLL file. |
| Interoperates well with other .NET languages and libraries. | Limited interoperability with .NET languages . |
| Runs on any platform that has CLR installed. | Runs only on the specific platform for which it is . |