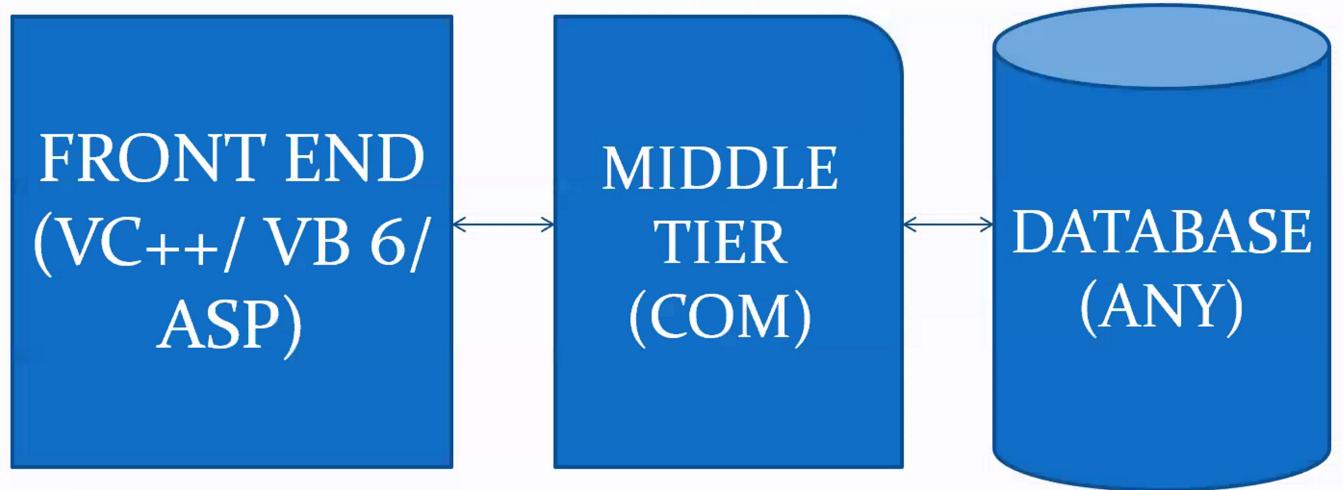


Before .Net (Microsoft)



Front end : UI

Middle tire : Component object -> call to front end

Problems (Pre .Net)

- VC++ -> Had OO and threading but Complex
- VB6 -> Not OO and no threading but Simple
- ASP -> Script based, Interpreted, Late Bound, Difficult to maintain and debug, not OO
- COM -> DLL HELL! (Mainly versioning and deployment)

OO :- object oriented

ASP :- web best development . Active server pages used to create web best development , it is slow

COM:- is a technology , called as DLL HELL

DLL HELL :- problem facing with .net that are versioning and deployment

.Net Features

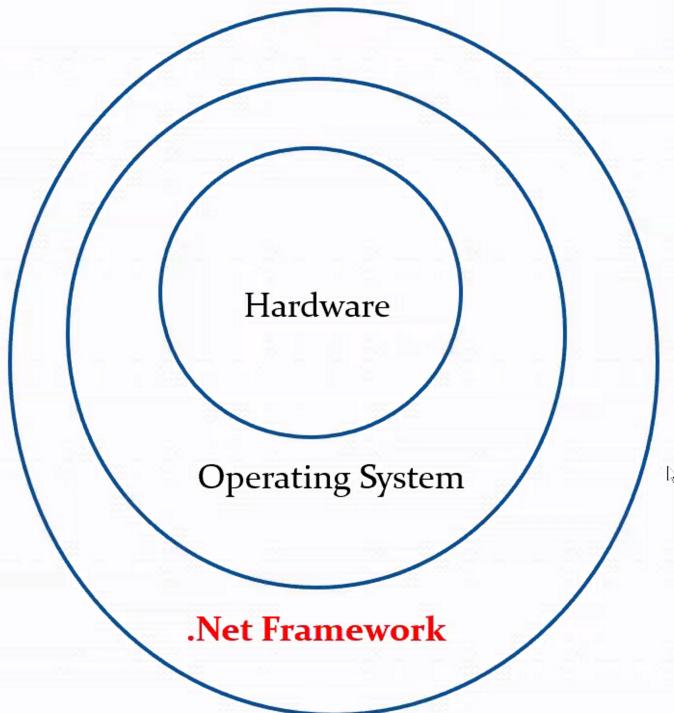
- OO Code
- Multiple Languages
- Multiple platforms*
- Multiple project types(eg web based, desktop based, etc)
- Better Security
- Improved Performance

50 languages mai use kr sakte hai .net ko

We can write in c#, python

Mutiple platforms* :-

.Net framework



99% it will work with the OS

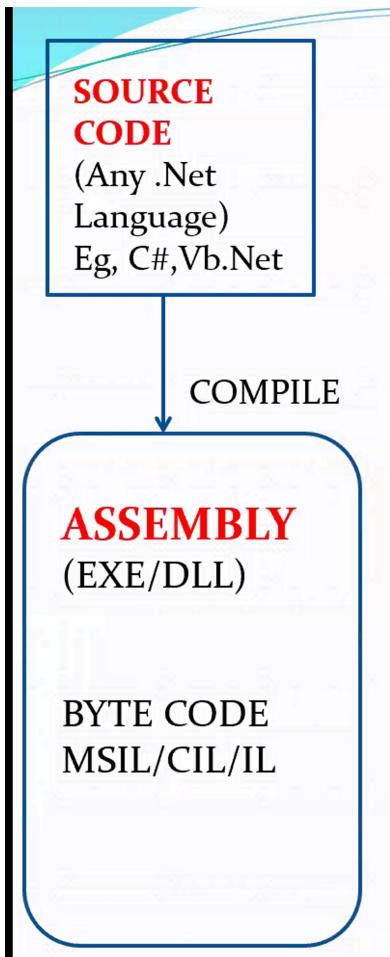
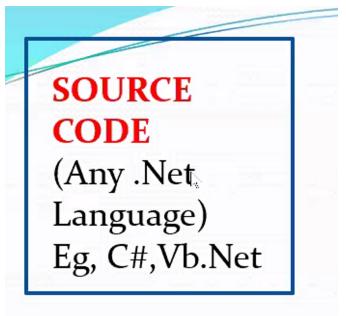
Web Application, Windows Forms, Console Apps, Web Services, WCF, WPF, Workflow, ASP.Net MVC, .Net Core, Xamarin, Windows Services, Web API

WCF - windows communication foundation

WPF - windows page foundation

.Net Base Class Library

System.Dll, System.Data.Dll, System.Xml.Dll etc



Assembly entry point define then EXE
If not define then DLL

Main function : EXE

Not Main Function : DLL

Assembly : compile code, it is in byte code formate

Name byte code :

MSIL : Microsoft Intermediate Language

CIL : Common Intermediate Language

IL : Intermediate Language

Common Language Runtime(CLR)

In java we have JVM Here we have CLR (Common language runtime)
.net have same runtime on compile time it will convert to IL

JIT compilation : just in time converting IL into native code

To do jit compilation one compiler is there jit compiler it also called **jitter**

1. Fast load time
2. Overhead of jit compiler
3. .net code is faster as compare to cpp
4. Cpp use in game code because directly pointing to hardware
5. Wpf used in gamming creation

Memory Management :

1. Run an app
2. Allocating memory
3. Loading memory
4. Decide were heap or stack

Garbage Collection :

1. Responsible for release the freeing memory
2. No longer being in reference
3. Run when system is slow
4. Non deterministic finalization
5. .net destructor get called when garbage collection called
6. When does GC run : system is slow on resource (Non deterministic finalization)
7. We do not write code in destructor because we don't know when destructor called
8. We should not write code in destructor

9. When should write code in destructor : don't ever write code in destructor, NO DON'T, NEVER

Common Language Runtime(CLR)

JIT Compilation

Memory Management

Garbage Collection

App Domain Management

```
Main()
{
    class1 o = new class1();
```

```
I           o.Dispose();
//o = null;
```

.....

}

Dispose present in interface idisposable

Dispose we called explicitly

Destructor is automatically called when GC run

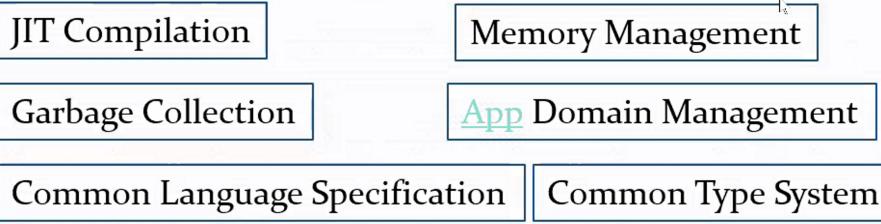
App Domains



App Domain Management :

1. App domain is an area within an app where .net runs
2. Before .net it requires process
3. Within process in which your application runs.
4. In app domain : If one process crashes then others will not get affected

Common Language Runtime(CLR)



Common Language Specification :

1. In db .net multiple inheritance is not allowed
2. CLS : common set of rules that all .net language must follow.

- 3. Not run if cls rule not followed**
- 4. .net manage c++ language is having multiple inheritance -> choose no GC to run**
- 5. Cls rule only single inheritance**

Common type system :

```
CSharp code  
int DoSomething(int a) -----> System.Int32  
{  
...  
}
```

```
VB.Net code  
  
Dim i as Integer -----> System.Int32  
i = 100  
i = DoSomething(i) ---->>> program crashes
```

1. Common data type in .net

Thread Management

Security Management

Debugging

Exception Handling

Thread Management :

Security management :

1. Role based security

a. Login credention

2. Identity based sercurity : who are are

3. Code access security : where are you , running from same machine or internet

a. What permisstion is given

b. Caspol code access policy

c. Or go through UI

Common Language Runtime(CLR)

JIT Compilation

Memory Management

Garbage Collection

[App](#) Domain Management

Common Language Specification

Common Type System

Thread Management

Security Management

Debugging

Exception Handling

.net Framework

1. Base class +
2. CLR +
3. Utilities (ex. compiler)

To run we need Base Class and CLR

Mono is work on linux but it is not developed by MS

Difference between .Net Framework, .Net Core, Mono and Xamarin

- .Net Framework used for Windows platform mainly
- Mono used for Linux
- Xamarin used for Mobile platforms(Android, iOS and Windows)
- .Net Core used for all platforms

.Net Core

- Upto Version 2.2 only supports ASP.NET MVC and Web APIs
- Version 3+ supports Winforms and WPF also
- Currently version 5 (.Net 5)

What will we be working with

- Visual Studio 2019 Community Edition- .Net framework 4.7
- C#
- Console Apps/Class Library
- WPF intro
- Asp.Net MVC
- Web Services
- WCF
- Web Api
- .Net Core Intro

Managed vs Unmanaged code

- Managed code is run by CLR (All .net code)
- Unmanaged code is not run by CLR
 - Egs Windows DLLs (PInvoke) and COM Apps (using COM Interop)