

* History Of C#:

Developed in

- Microsoft

Developer

- Anders Hejlsberg (Team Leader)

Name

- COOL [C-Like Object Oriented Language] (Jan. 1999)

Re-Name

- C# (July 2000)

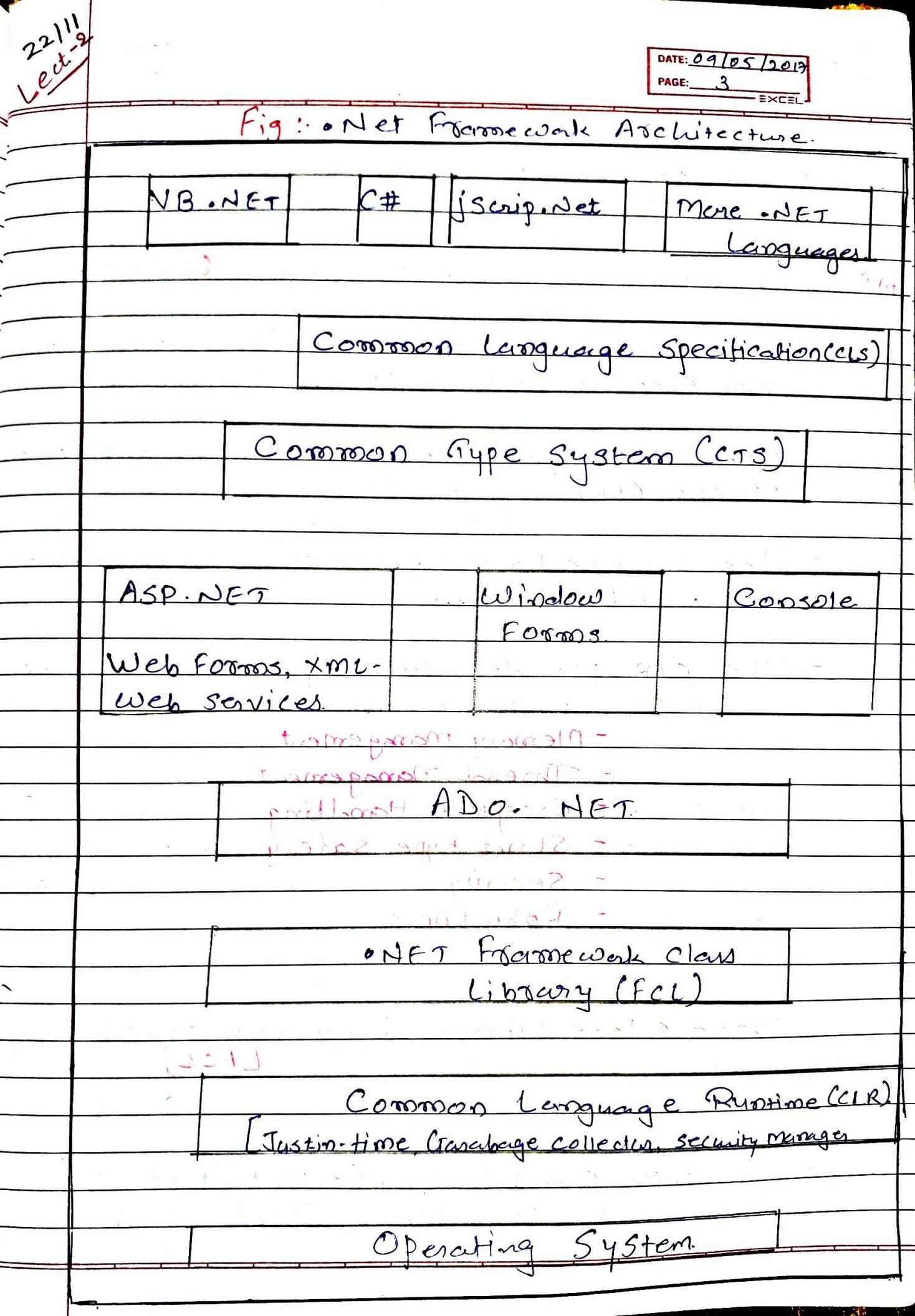
- The C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.
- Portability is very important for source code and programmers, especially those already familiar with C and C++.

* Introduction to .NET Framework:

- The .NET Framework is a technology that supports building and running the next generation application.
- The .NET Framework consists of two parts.
 - Common Language Runtime [CLR]
 - .Net Framework Class Library [FCL]

CLR

- The common language runtime is the foundation of the .Net Framework.
- The concept of code management is a fundamental principle of the runtime.
- [Code that targets the runtime is known as managed code.] Managed code definition.
- [Code that does not target the runtime is known as unmanaged code.] Unmanaged code defi
- The class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) application.



* Components of the .Net Architecture:-

M.M Group 1) Common Language Runtime: (2 to 3 marks)

- The Common Language Runtime (CLR) serves as the execution engine of .Net Framework.
- All .Net programs execute under the supervision of the CLR.
- The common Language Runtime is the foundation of .Net Framework.
- The CLR provides such type of Services like
 - Memory Management
 - Thread Management
 - Exception Handling
 - Strict type safety
 - Security
 - Robustness.

2) Base Class Library / Framework Class Library / [FCL]

Namespace:

- The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library

of functionality available to all languages using .Net Framework.

- The BCL provides classes that encapsulate a number of common functions, including file reading and writing, graphics rendering etc.
- It consists of
 - Classes
 - interfaces of reusable types that integrate with CLR.

3) Manage code and Unmanaged code:-

[1 to 2 Marks]

- Manage code :-

- It runs in the Common Language Runtime.
- Manage code creates by .Net Compiler
- It Compiles to intermediate language (IL), not to machine code that could run directly on your computer.

- Example: Garbage collection, Type safety etc.

- Unmanaged Code :-

- It's directly executed by the computer's CPU.
- Unmanaged code produced by Visual Basic 6.0, Visual C++ and before .Net 2002
- It is compiled directly to machine code that ran on the machine where you compiled it.
- Calling unmanaged code presents a major security risk.

4) Common Type System (**CTS**) and Common Language Specification (**CLS**) :-

[2^{to} 3 may]

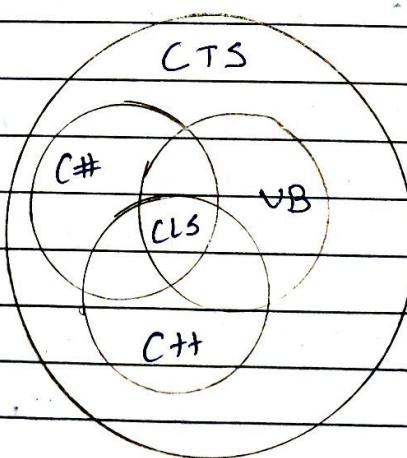


Fig : CTS and CLS

- Common Type System (CTS) describes a set of types that can be used in different .NET languages in common.
(SJD)
- CTS ensure that Object written in different .NET languages can interact with each others.
- There will be two type to communicate between program written in any .NET languages is as below.
 - Value Type
 - Reference Type
- The Value Type are passed by Values and stored in Stack.
- The Reference Type are passed by references and stored in heap.
- Common Type System provides base set of Data Types which is responsible for cross language integration.
(SJD)

5) Common Language Infrastructure (CLI):-

- The purpose of the Common Language Infrastructure (CLI) is to provide a language-neutral platform for application development and execution.
- Microsoft's implementation of the CLI is called the Common Language Runtime (CLR).

6) Microsoft Intermediate Language : (MSIL/IL/CIL)

- MSIL stands for Microsoft Intermediate Language. We can call it as Intermediate Language (IL) or Common Intermediate Language (CIL).
- During the Compile time, the Compiler converts the Source code into MSIL.
- MSIL is CPU independent set of instructions that can be efficiently converted to the native code.

- During the Runtime CLR's "Just in Time" Compiler Converts the MSIL Code into native code to the Operating System.
- When a Compiler produces MSIL it also produces meta-data. MSIL and meta-data contained in a portable executable (PE) file.

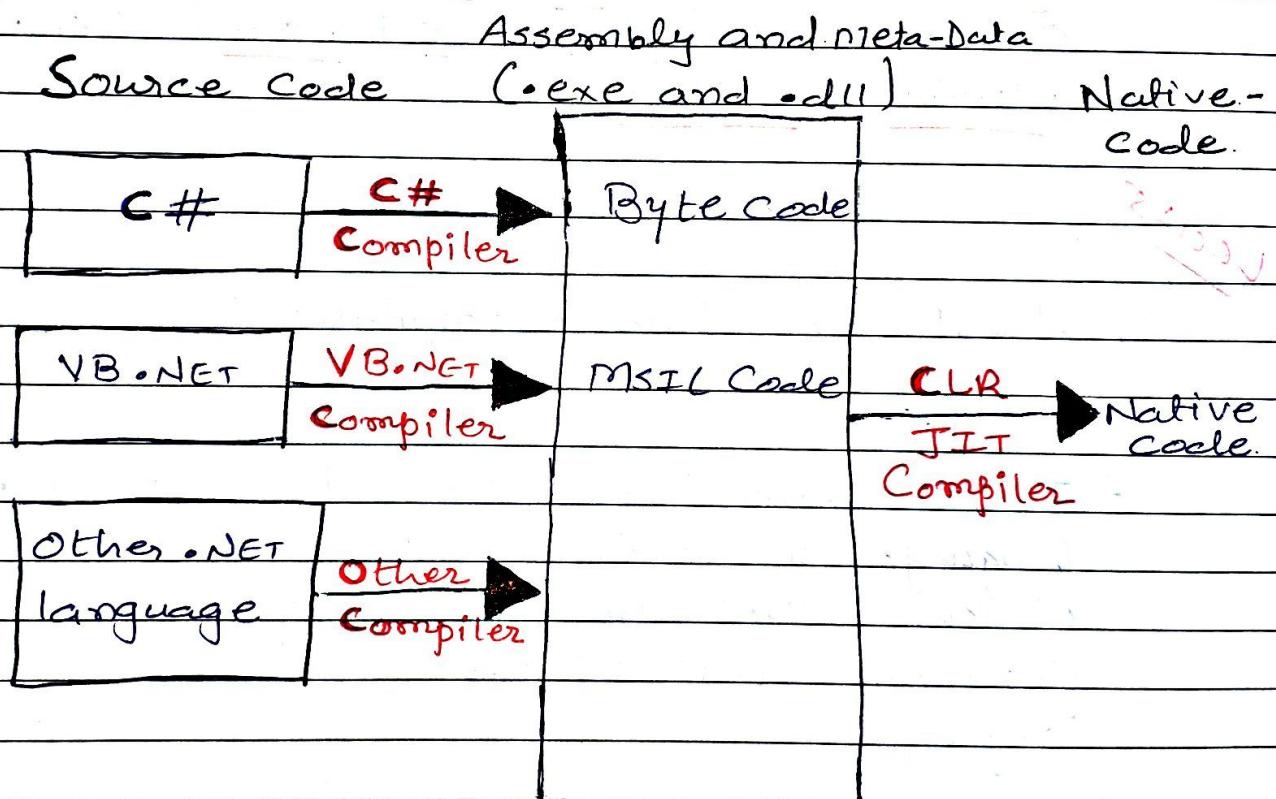


Fig.:~ MSIL, Assemblies and Meta-Data

7) Assemblies and Meta-Data :-

- The CIL code is housed in CLI assemblies.
- Assemblies are stored in the Portable Executable (PE) format, common on the Windows platform for all DLL and EXE files.
- Meta-Data is Data about Data that contains Complete name of an assembly, Contains its simple text name, Version number, Culture.

23/11
Lect-5

8) Just-In-Time Compilation:-

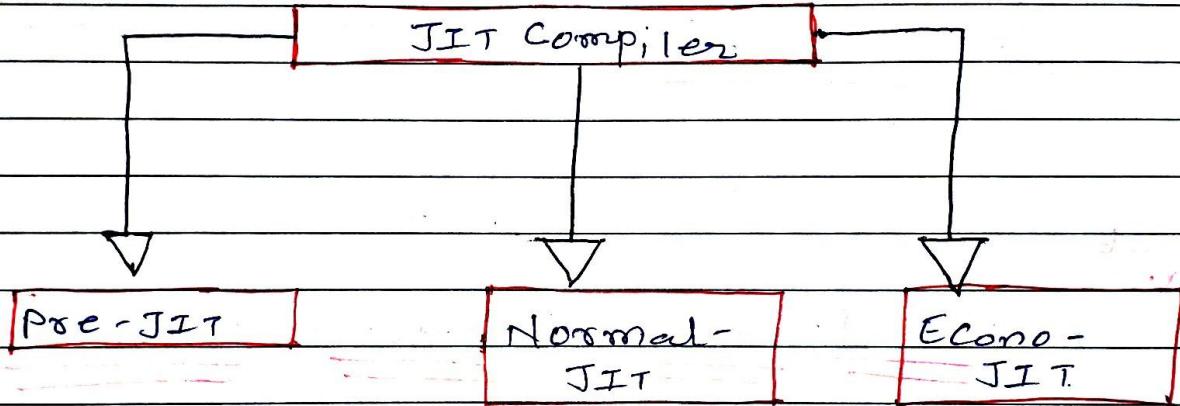
- In the .Net Framework, all the Microsoft .Net languages use a Common Language Runtime, which solves the problem of installing separate runtimes for each of the programming languages.
- The Web Forms page is implicitly compiled, it is actually compiled twice. On the first pass, it is compiled into IL. On the second pass, the assembly in IL is compiled into machine language this process is called Just In Time (JIT) compilation.]

Definition

- It does not occurs until the assembly is on the target machine.

JIT Types:

- There are Three types of JIT Compilers



○ Pre-JIT Compiler:

- Pre-JIT Compiles Complete Source Code into native code in a single compilation cycle.
- This process is done at the time of deployment of the application.

○ Econo-JIT Compiler:

- Econo JIT Compiles only those methods that are called at runtime. However, these compiled methods are removed when they are not required.

① Normal-JIT Compiler:-

- Normal-JIT Compiles only those methods that are called at runtime. These methods are compiled the first time they are called and then they are stored in cache.
- When the same methods are called again, the compiled code from Cache is used for execution.



23/11
lect: 6

* • .Net Framework Features / Advantages :-

1) Multilanguage Development (CTS & CLS)

- • Net user can create application in many languages. You can create your application in C#, VB.NET, VCL etc.
- • Net platforms allows languages to be integrated with one another through the use of MSIL.

2) Platform and Processor independence :- *(Write once and run anywhere.)*

- .Net Application can execute on any platform that supports the .Net Common Language Runtime (CLR).
- It is also provide Automatic Memory Management.

3) Interoperability :-

- Computer Systems Commonly require interaction between newer and older applications. .Net Framework provides means to access functionality implemented in newer and older programs that executes outside .Net environment.

4) Security :-

- The design addressed some of the process, such as buffer overflow, which have been exploited by malicious software.
- .Net provides a Common Security model for all applications.

5) Portability :-

- Any application which are developed under the .Net platforms you can run any third party Computer Systems without that platforms very easily.

* Visual Studio IDE (Integrated Development Environment)

- Visual Studio is Microsoft's Integrated Development Environment (IDE) for creating, running and debugging programs written in a variety of .Net programming languages.
-