

# Assignment No. 1

- Q.1) Write a note on .Net Framework in detail?
- .Net is a framework to develop SW appn.
  - It is designed & developed by Microsoft.
  - So the first beta version released in 2000.
  - It is used to develop appn's for web, windows, console-based, phone. Moreover, it provides a broad range of functionalities & support.
  - The .Net framework is a pure object oriented, that similar to the Java language. But it is not a platform independent, as the Java. So its appn runs only to the windows platform.
  - It is a virtual machine that provides a common platform to run an appn that was built using the diff' language such as C#, VB.NET, Visual Basic, etc.
  - The main objective of this framework is to develop an appn that can run on the windows platform.
  - The current version of the .Net framework is 4.8.

## \* Components of .Net Framework

- CLR (Common Language Runtime)
  - It is an important part of a .N.F. that works like a virtual component of the .Net Framework to executes the diff' languages program like C#, Visual Basic, etc.

- A CLR also helps to convert a source code into the byte code, & this byte code is known as CIL (Common Intermediate Language) or MSIL (Microsoft I.L). After the converting into a byte code, a CLR uses a JIT compiler at run time that helps to convert a CIL or MSIL code into the machine or native code.

### CTS (Common Type System)

- It specifies a standard that represent what type of data & value can be defined and managed in computer memory at runtime.
- A CTS ensures that programming data defined in various languages should be interact with each other to share info.
- For Ex:- In C# we define data type as int, while in VB.NET we define integer as a data type.

### BCL (Base Class Library)

- BCL is a collection of classes, interfaces, & value types that provide a wide range of functionality for common programming tasks.
- It includes classes for working with strings, collections, I/O, networking, XML & more.

### CLS (Common Language Specification)

- It is a subset of CTS that defines a set of rules & regulations that all .Net

- Languages follow to ensure compatibility  
- It specifies naming conventions, data types, & other guidelines for how code should be written in diffn languages to work well together.

### FCL (Framework Class Library)

- FCL is synonymous with BCL.

- It is the collect'n of classes & libraries that developer can use to build .Net app's.

### .NET Assemblies

- Assemblies are the building blocks of .NET app's.

- They contain compiled code, metadata, & resources required for execution.

- Assemblies can be in the form of executable file (.exe) or dynamic link libraries (dll).

### XML Web Service

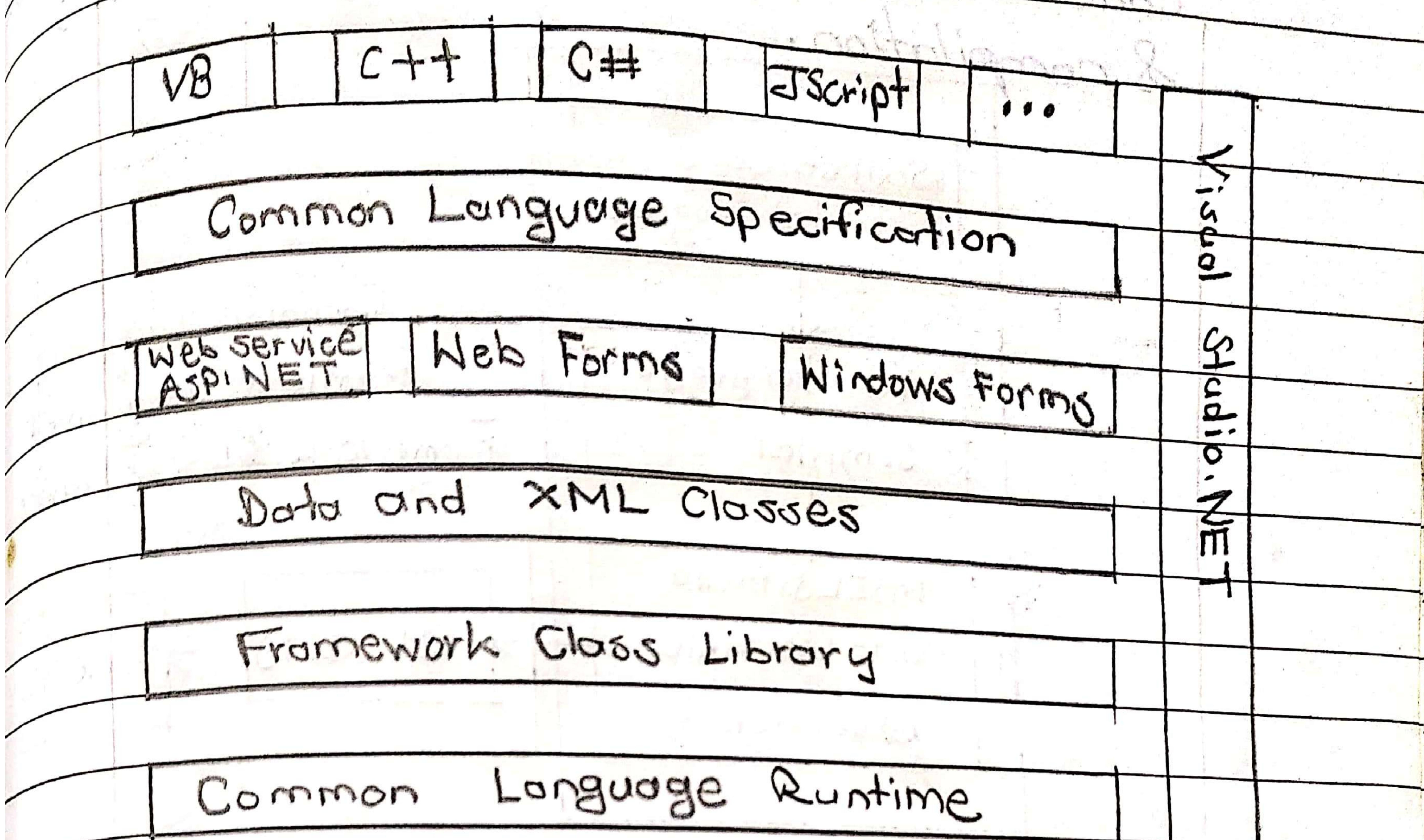
- XML Web services allow apps to communicate & share data over the internet using XML-based standards such as SOAP (Simple obj. Access Protocol) & WSDL (Web Service Descript'n language).

### Windows Services

- W.S. are long running processes that run in the background on Windows OS.

- They can be developed using .Net framework & provides various sys.-level functionalities.

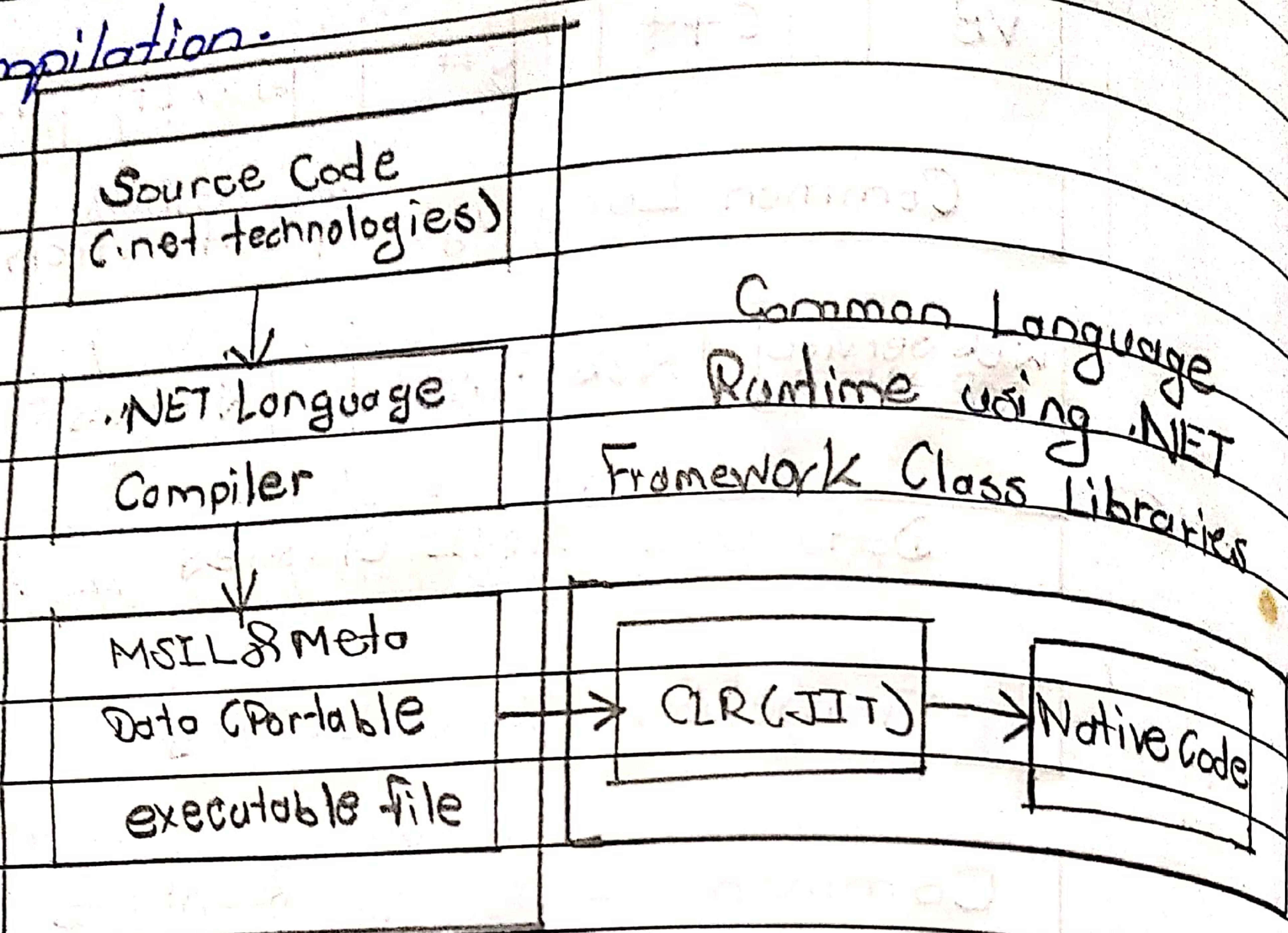
such as file processing, network communication & automated tasks.



Q2 Explain the CLR concept in detail?

- .NET CLR is a runtime environment that manages & executes the code written in any .NET programming language.
- CLR is virtual machine component of the .NET framework. That language's compiler compiles the source code of app's developed using .NET compliant languages into CLR's intermediate language called MSIL.
  - This code is platform independent. It is comparable to byte code in java.
  - A just-in-time compiler component of CLR converts MSIL code into native code of the machine.

- This code is platform-independent.  
 CLR manages memory, threads, exceptions  
 code execution, code safety verifications  
 & compilation.



Converting the source code into native code.

### \* Components of CLR

- Common Language Specification (CLS):
  - It is responsible for converting the "diff" .NET programming language syntactical rules & regulations into CLR understandable formats, it provides language Interoperability.
  - Lang. Inter. means providing execution support to other programming languages also in .NET framework.
- CTS - Common Type System
  - Every prog. lang. has its own data type sys so CTS is responsible for

understanding all the data type sys. of .NET programming languages & converting them into CLR understandable format which will be a common format.

### - Types of CTS

#### 1. Value Types

Value Types will store the value directly into the memory location. These types work with stack mechanism only. CLR allows memory for these at Compile Time.

#### 2. Reference Types

R.T. will contain a memory address of value because the R.T. won't store the variable value directly in memory. These types work with Help mechanism. CLR allows memory for these at Runtime.

### - Garbage Collector

It is used to provide the Automatic Memory Management feature. If there was no G.C., programmers would have to write the memory management codes which will be a kind of overhead on programmers.

### - JIT (Just In Time Compiler)

It is responsible for converting the CIL (Common Intermediate Language) into machine code or native code using the

## Common Language Runtime environment.

### \* Feature / Role of CLR

CLR provides many feature like it loads & executes the code, manages the memory, converts the MSIL code to native code, & handles the exception.

Here are some major feature of CLR.

- Memory Management

- Code Access Security

- Garbage Collection

- JIT Compilation

- Thread Support

- Debug Engine

- Exception Handling

- Code Access Security

CLR protects programs from harmful code by controlling what they can do on the computer.

- Thread Management

CLR helps programs handle multiple tasks at once by managing diff threads of execution.

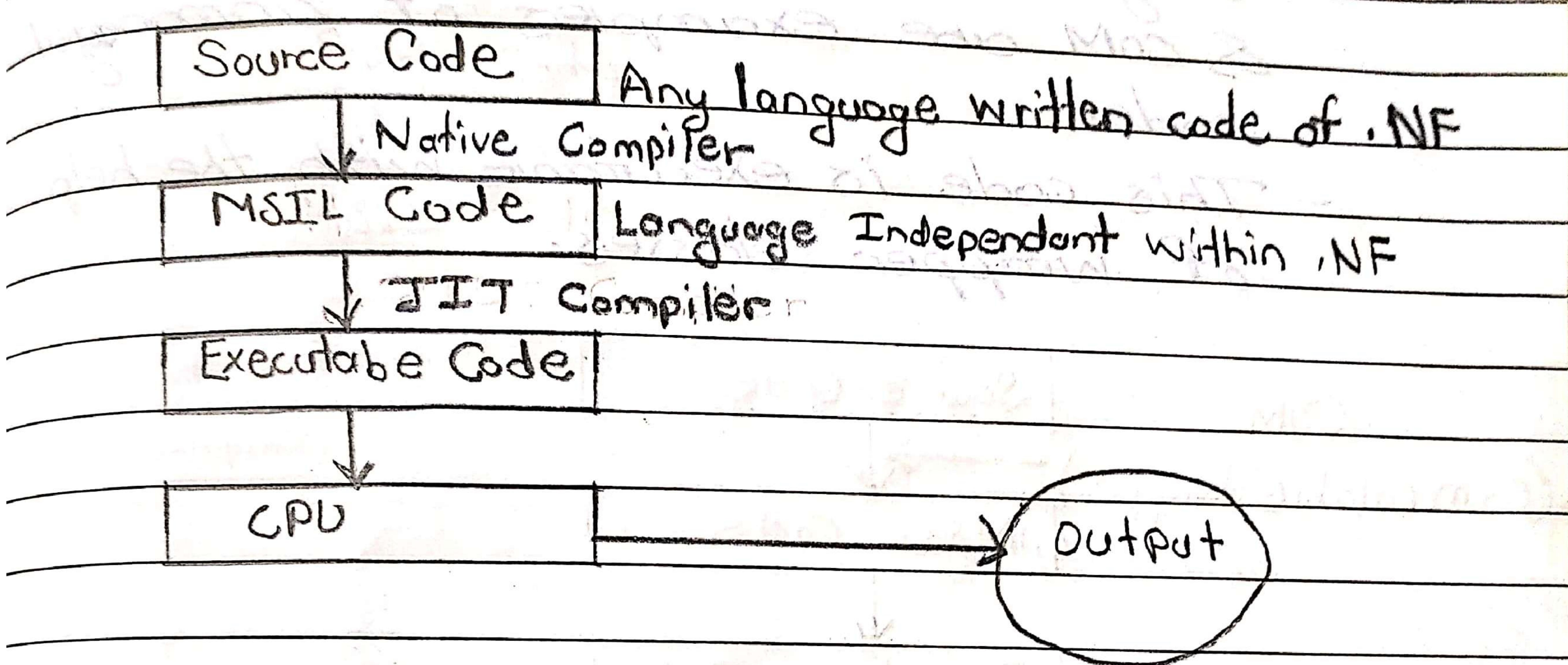
- Exception Handling

CLR helps programs deal with unexpected problems, like errors, in a controlled way.

Write a note on Managed & Unmanaged code?

Managed code

- Any language that is written in the .NET framework is managed code.
- Managed code uses CLRs which looks after your app's by managing memory, handling security, allowing cross-language debugging, etc.
- The process of managed code is shown in the figure:

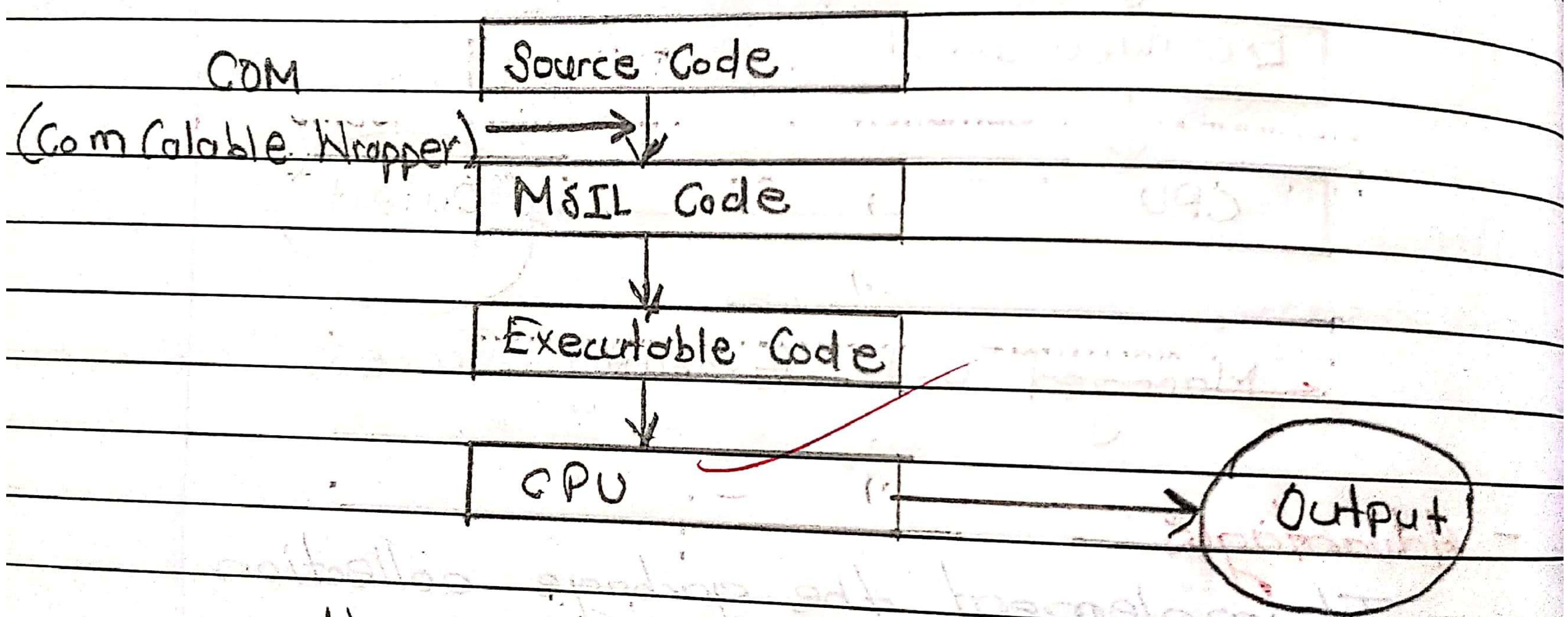


### Managed Code Execution Process

- Advantages
  - It implements the garbage collection automatically.
  - It also provides runtime type checking, dynamic type checking.
  - It also provides reference checking which means it checks whether the reference point to the valid object or not & also check they are not duplicate.

## \* Unmanaged Code

- The code developed outside the .NET Framework is known as unmanaged code.
- Applications that do not run under the control of the CLR are said to be unmanaged.
- Certain languages such as C++ can be used to write such apps, such as low-level access functions of the OS.
- Background compatibility with VB, ASP, & COM are examples of unmanaged code.
- This code is executable with the help of wrapper classes.




## Unmanaged Code Execution Process

- Advantages
  - It provides the low-level access to the programmer.
  - It also provides direct access to the hardware.

- Q.4 Explain the JIT Compiler with its types?
- Just-In-Time Compiler is a part of CLR in .NET which is responsible for managing the execution of .NET programs regardless of any .NET program language.
  - A language specific compiler converts the source code to the intermediate language.
  - This IL is then converted into the machine code by the JIT compiler.
  - This machine code is specific to the computer environment that the JIT compiler runs on.
- Working of JIT Compiler

### Working of JIT Compiler



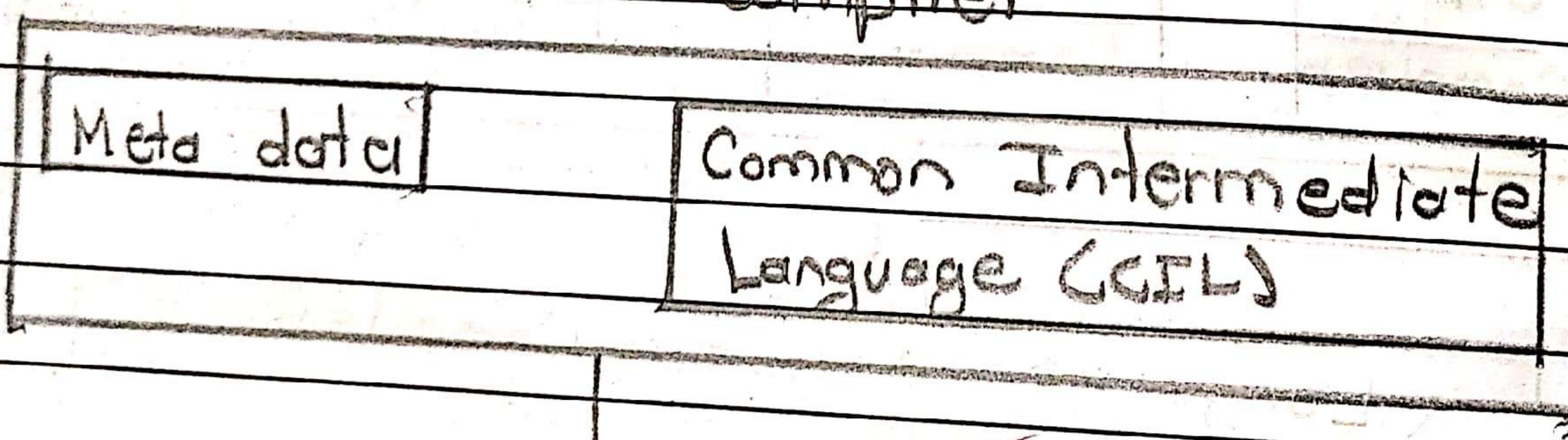
The JIT Compiler is required to speed up the code execution & provide support for multiple platforms. Its working is given as above in diagram.

#### 4 Types of Just-In-Time Compiler

##### - Pre-JIT Compiler

- All the source code is compiled into the machine code at the same time in a single compilation cycle using the Pre-JIT Compiler.
- This compilation process is performed at app's deployment time. And this compiler is always implemented in the Ngen.exe (Native Image Generator).

##### Pre-JIT Compiler



Pre-JIT Compiler Implemented in NGEN

Native Code

Execution

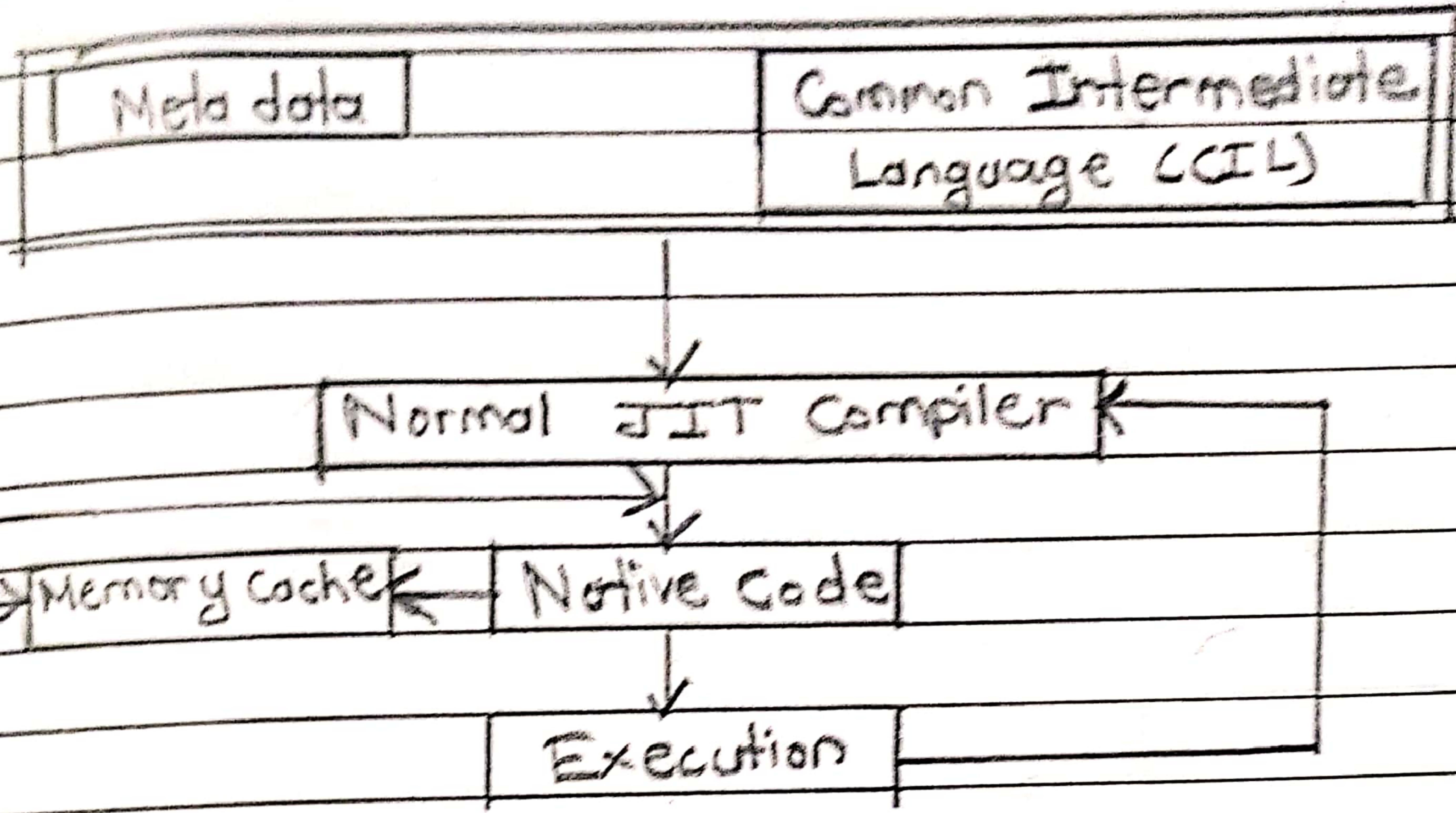
##### - Normal JIT Compiler

- The source code methods that are required at run-time are compiled

into machine code the first time they are called by the Normal JIT Compiler.

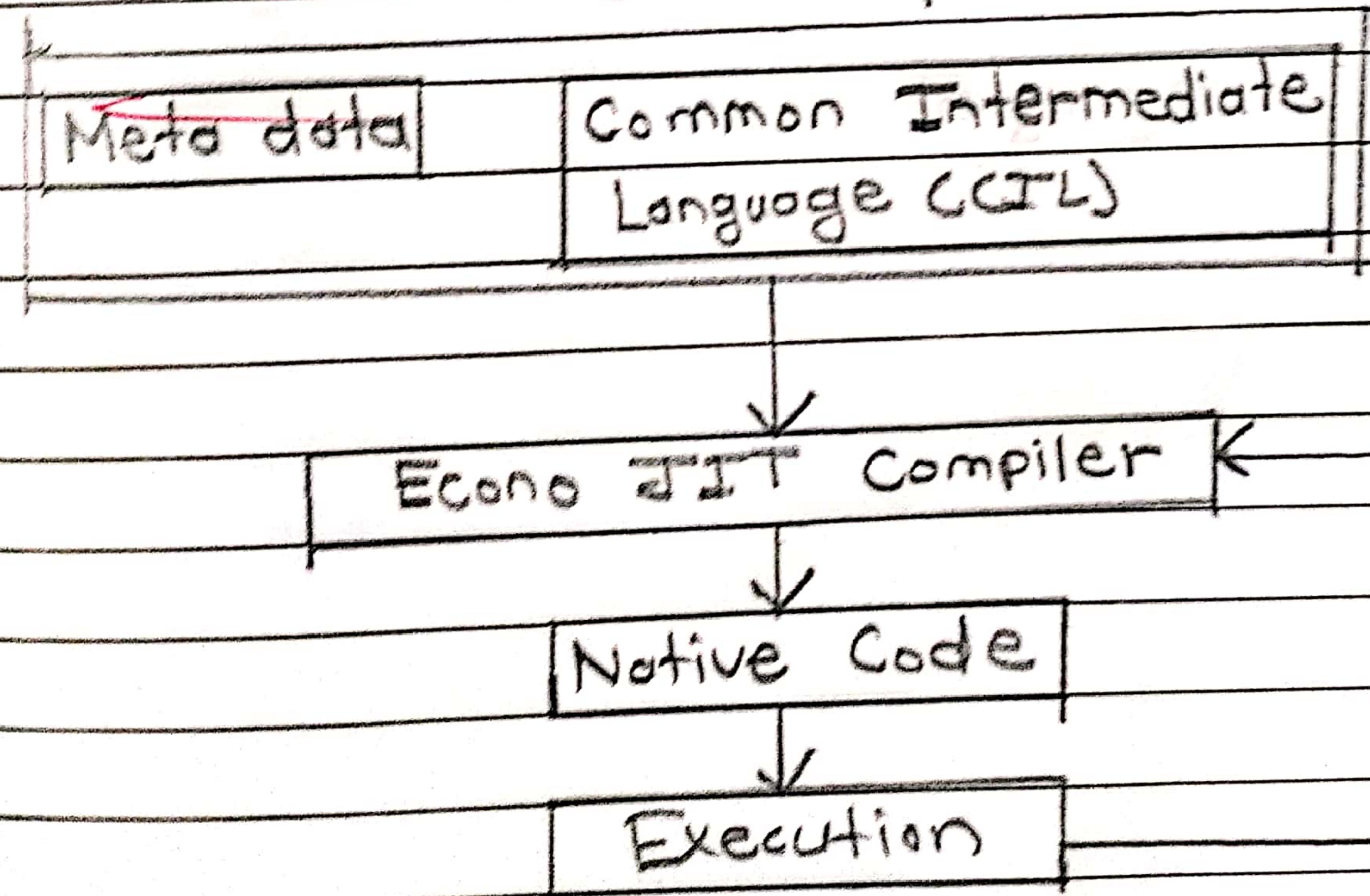
- After that, they are stored in the cache & used whenever they are called again.

### Normal JIT Compiler



### Econo JIT Compiler

#### Econo JIT Compiler



- The source code methods that are required at run-time are compiled into machine code by the E·J·C.
- After these methods are not required anymore, they are removed. This ~~JIT~~ compiler is obsolete starting from dotnet 2.0.

~~Boys~~