

# Introduction to WAD Dot Net Framework

Prepared for V<sup>th</sup> semester DDU-CE students  
2025-26 WAD

Apurva A Mehta

Chess was invented in \_\_\_\_.

# COs

1. Apply Core C# language concepts and build console based applications using reusable library (DLL).
2. Design, develop and deploy database driven web applications in .NET framework in Visual Studio Editor by using various server and validation controls, session management techniques and ActiveX Data Objects(ADO).
3. Design, develop and deploy dynamic, cross-platform web applications in Visual Studio Code editor using .NET Core and MVC design patterns.
4. Utilize Entity Framework - An ORM framework for powering .NET and .NET Core driven web applications.

# What is .NET?

- Framework
- NET is actually a **cluster of technologies** that are designed to help developers build a variety of different types of applications.

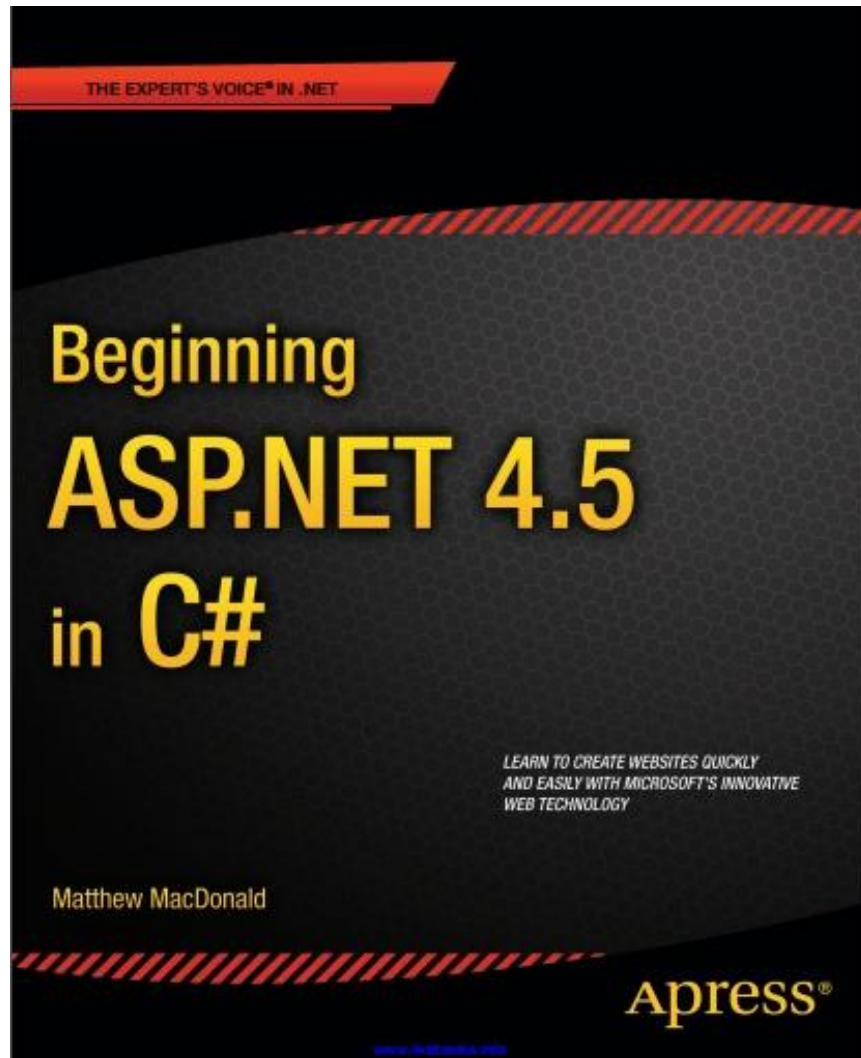
# Types of Application

- An Application can be :
  - **Web Application**
  - Windows Application
  - Mobile Application
  - Web Service
  - Command Line Application
  - Gaming Application
  - AI and ML Application
  - Cloud Application

# ASP.NET<sup>2002-2019-...</sup>

- ASP(Active Server Page )
- Specific Subset of .NET Technologies used to build web applications
- Program dynamic web pages
- Free from low-level implementation details
- Normally **C#** Language is used to build web applications using ASP.NET.

# Text Book



**Author : Matthew MacDonald**

# Web Development

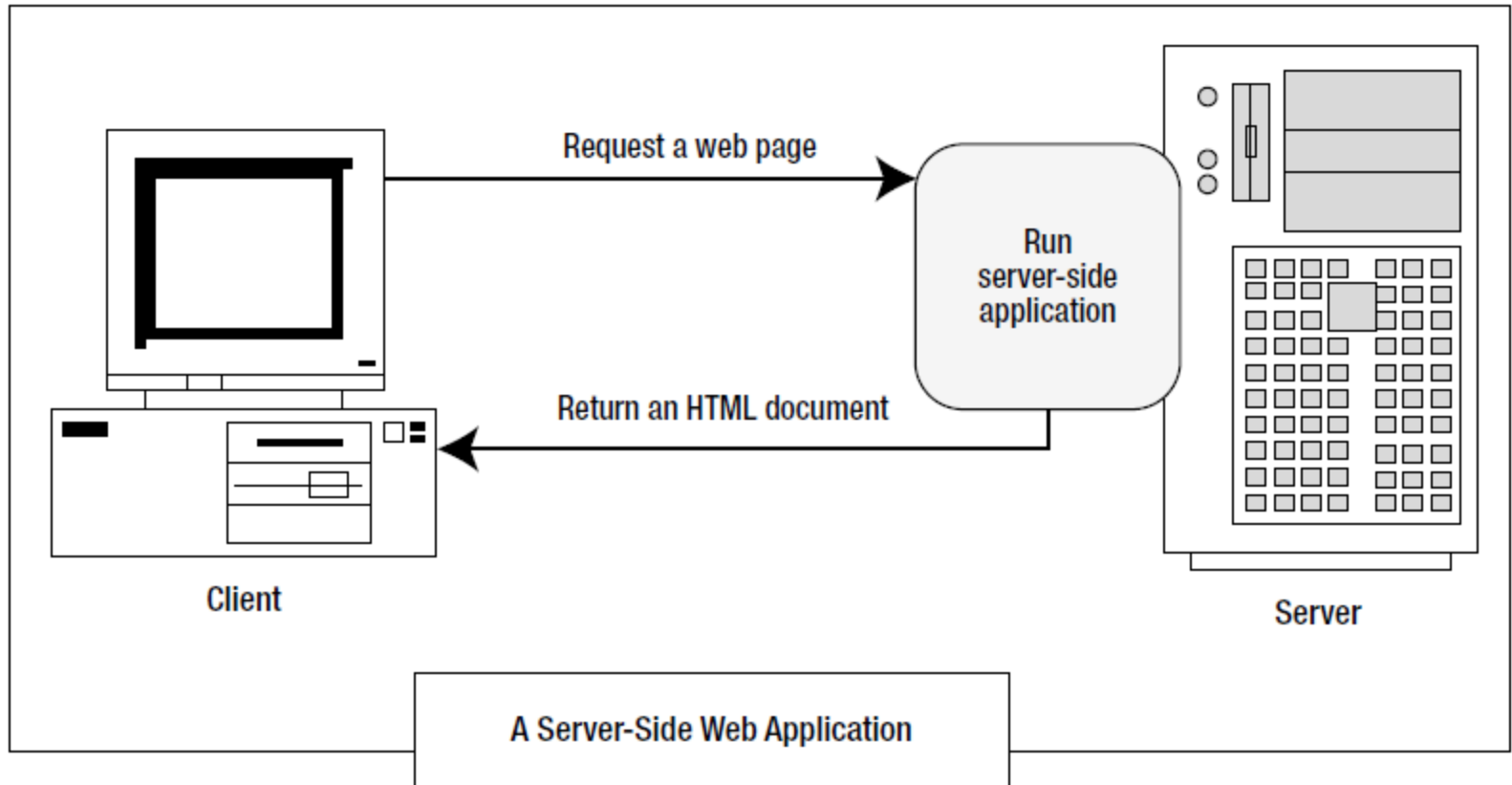
- HTML Page : Text + Elements
- HTML forms allow web developers to design standard input pages
- Server side, a custom application receives and processes the data.



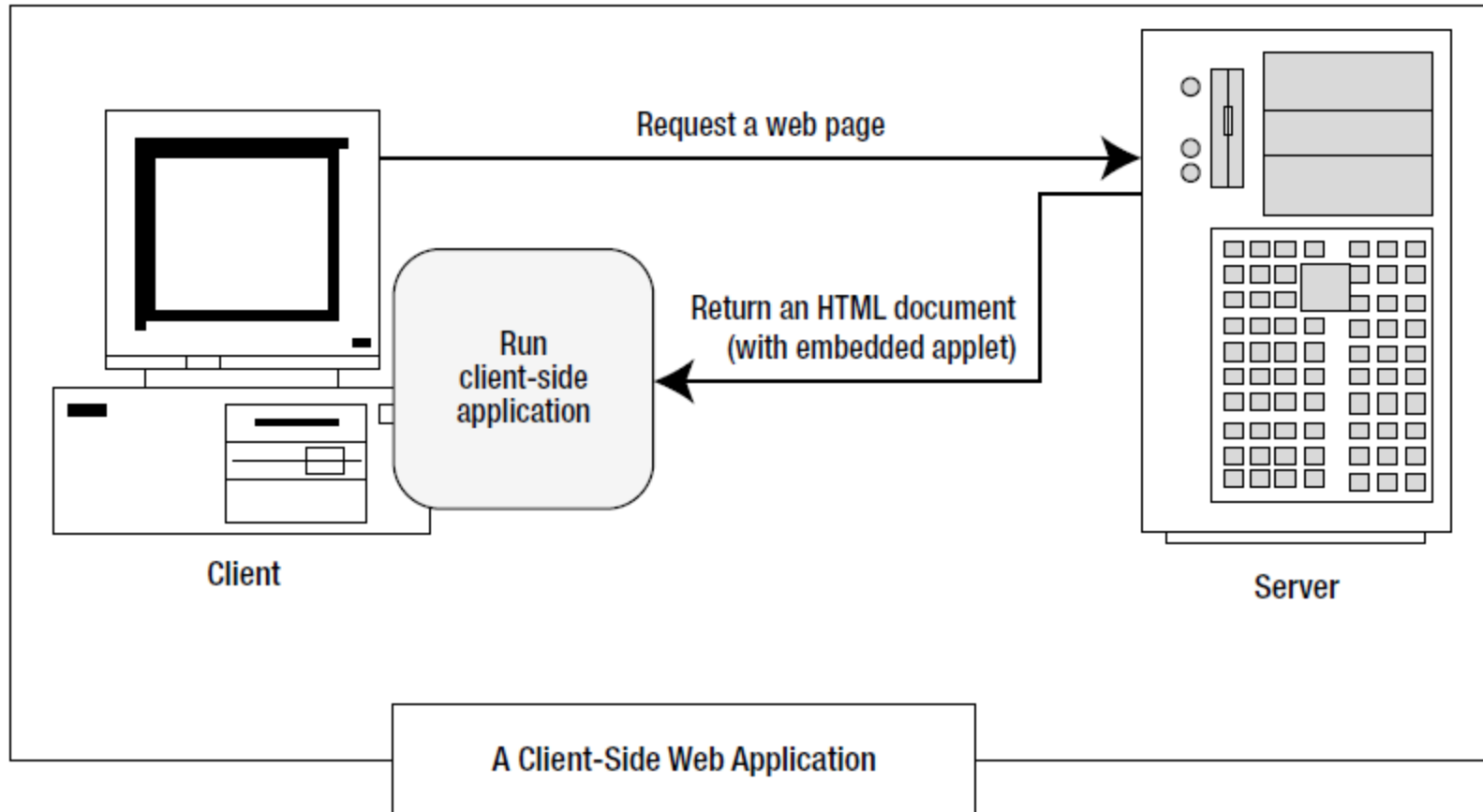
# Client-side vs Server-side Web Application



# Server-side web application



# Client-side web application



# Analysis

- Server side programming works good but there may be **scalability** issues.
- Issues in Client side programming :
  - Isolation
  - Security
  - Thin clients

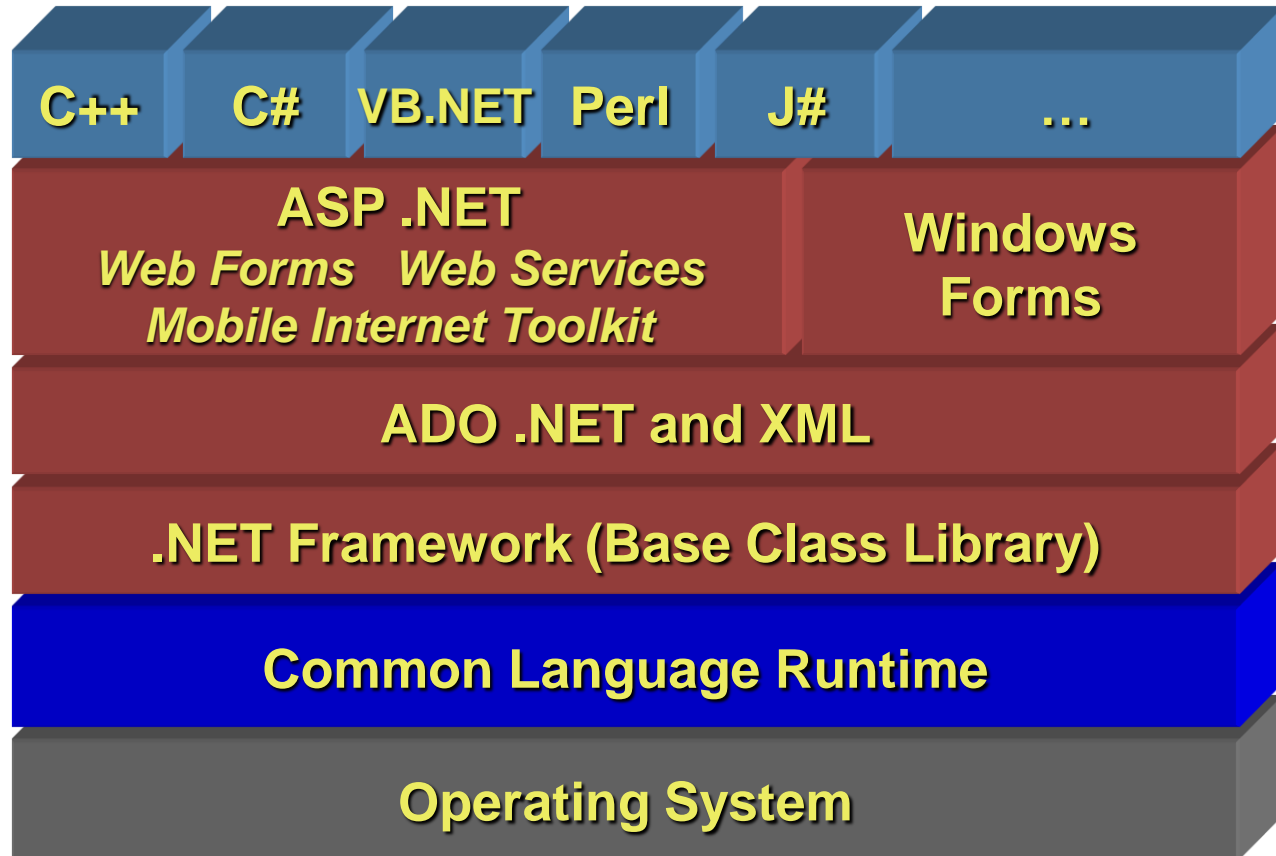
# .NET Framework version

- .NET 1.0: 2002-02-13
- .NET 4.8: 2019-04-18
- **.NET Core 3.1: December 2019**
- **.NET 8 (.NET Core): November 2023**

Which of the following is not an issue in client side programming languages?

- A. Isolation
- B. Scalability
- C. Security
- D. Thin clients

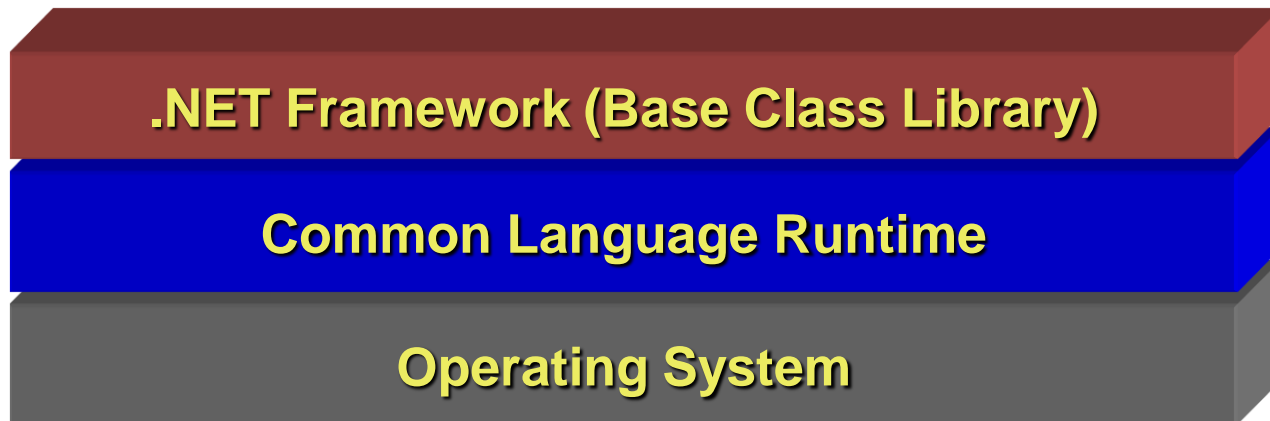
# The .NET Framework



# .NET Framework

## Base Class Library

- Prefabricated functionality for everything from reading an XML file to sending an e-mail message
- Well-stocked programmer's toolkit
- Any .NET language can use the .NET class library's features by interacting with the right objects.

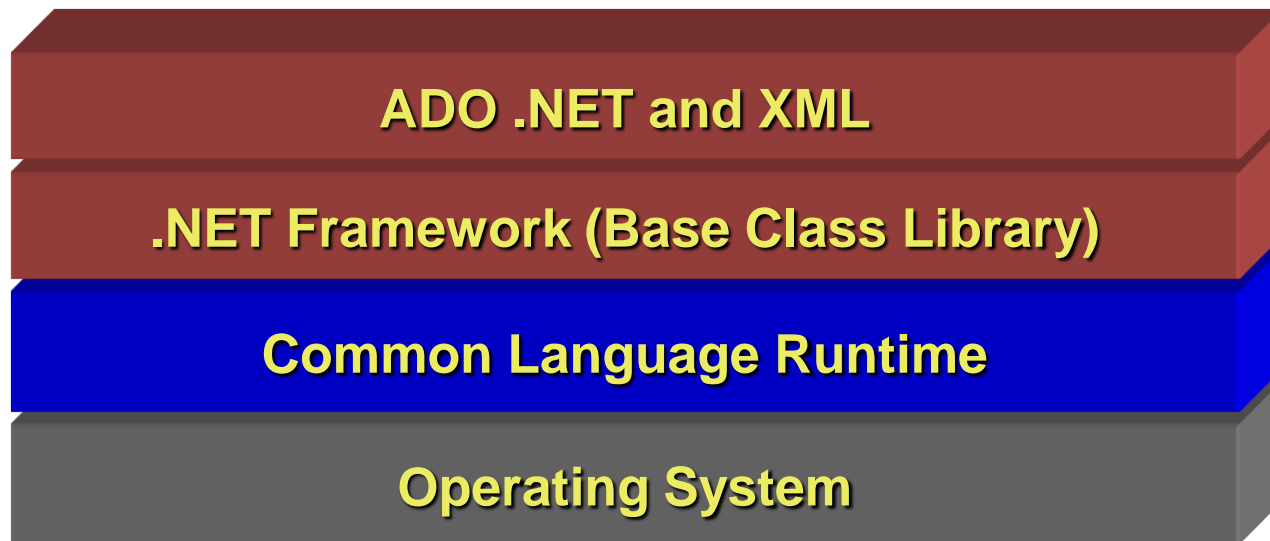




# .NET Framework

## Data Access Layer

- XML : universally adopted language for internet message passing.
- ADO.NET : handling database connection and maintenance with front-end .NET technology.

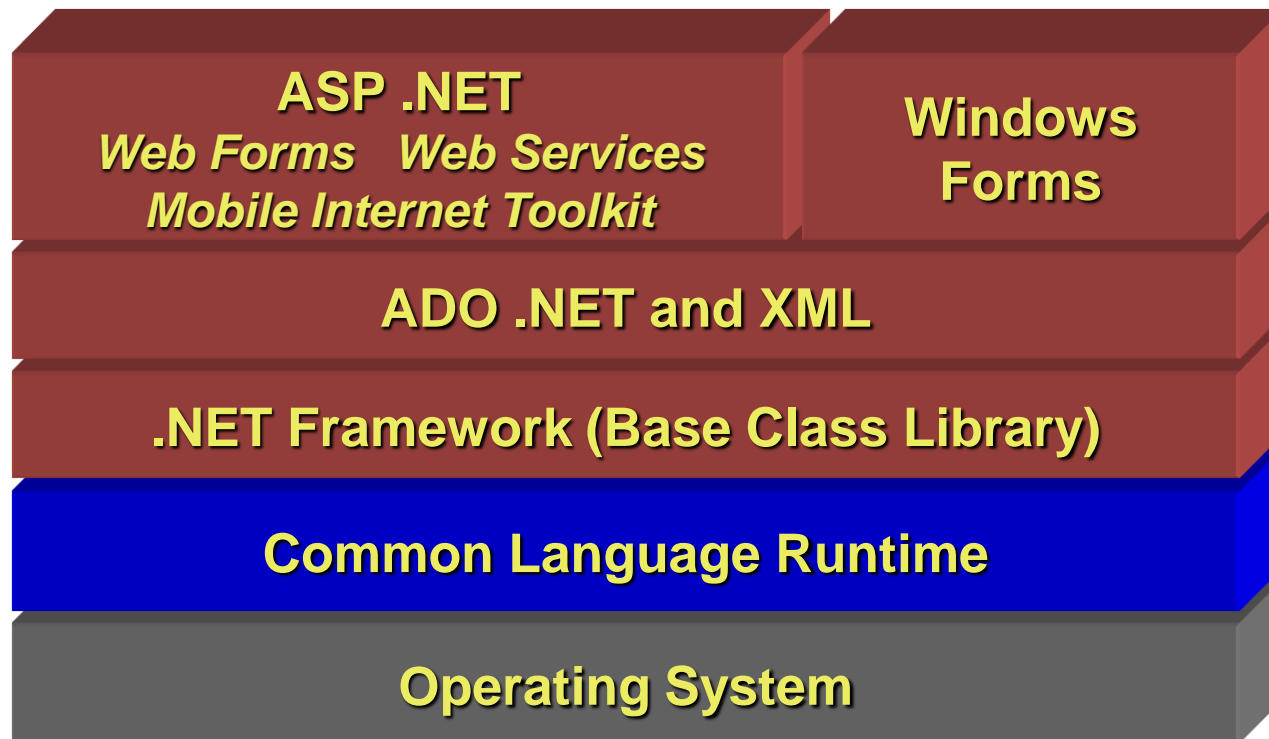


Which of the following is not true for Base Class Library of .NET framework?

- A. It supports String handling, IO operations, graphics, text handling, file operations and many more.
- B. It helps encourage consistency among different .NET languages and removes the need to install numerous components on your computer or web server.
- C. Infrastructure is available, just focus on business specific codes.
- D. It is the runtime environment.

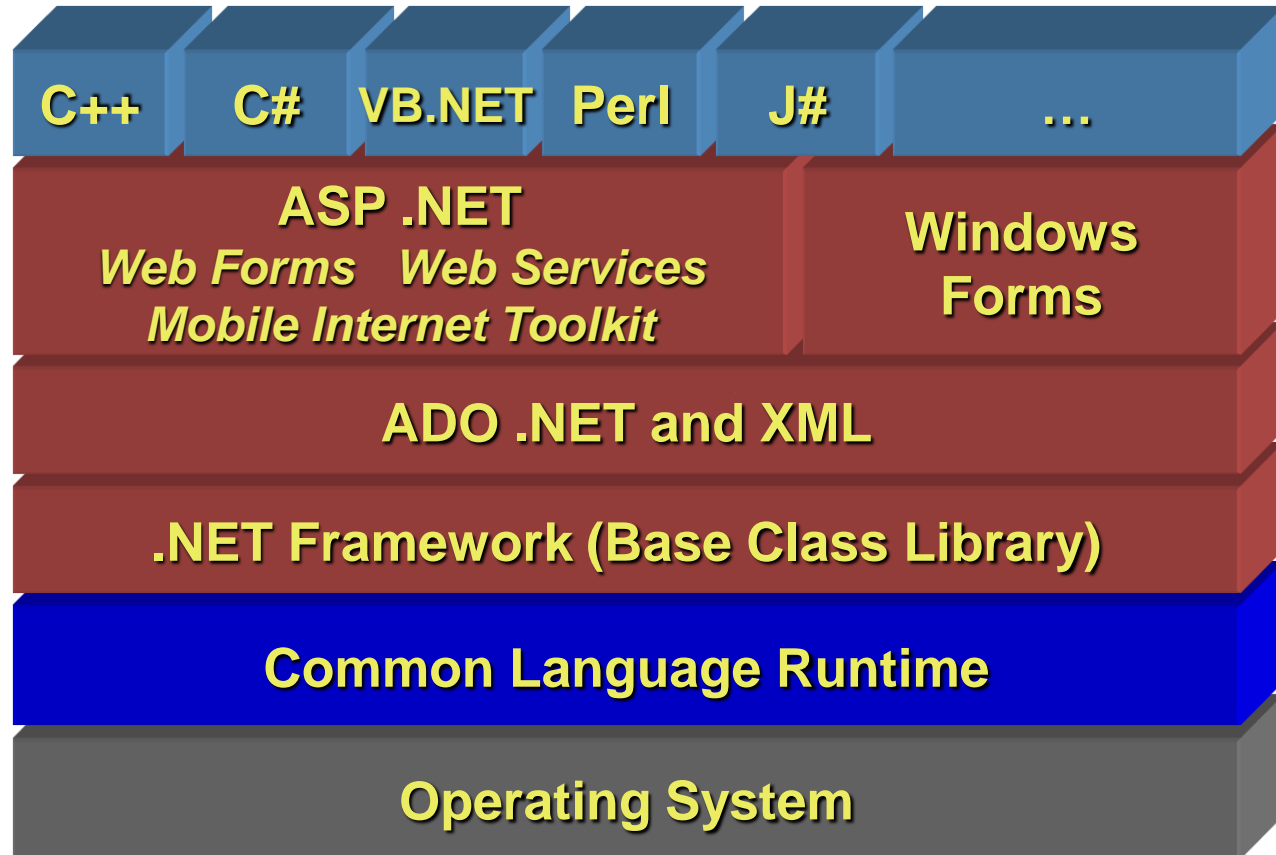
# .NET Framework

## ASP.NET & Windows Forms



# .NET Framework

## Programming Languages



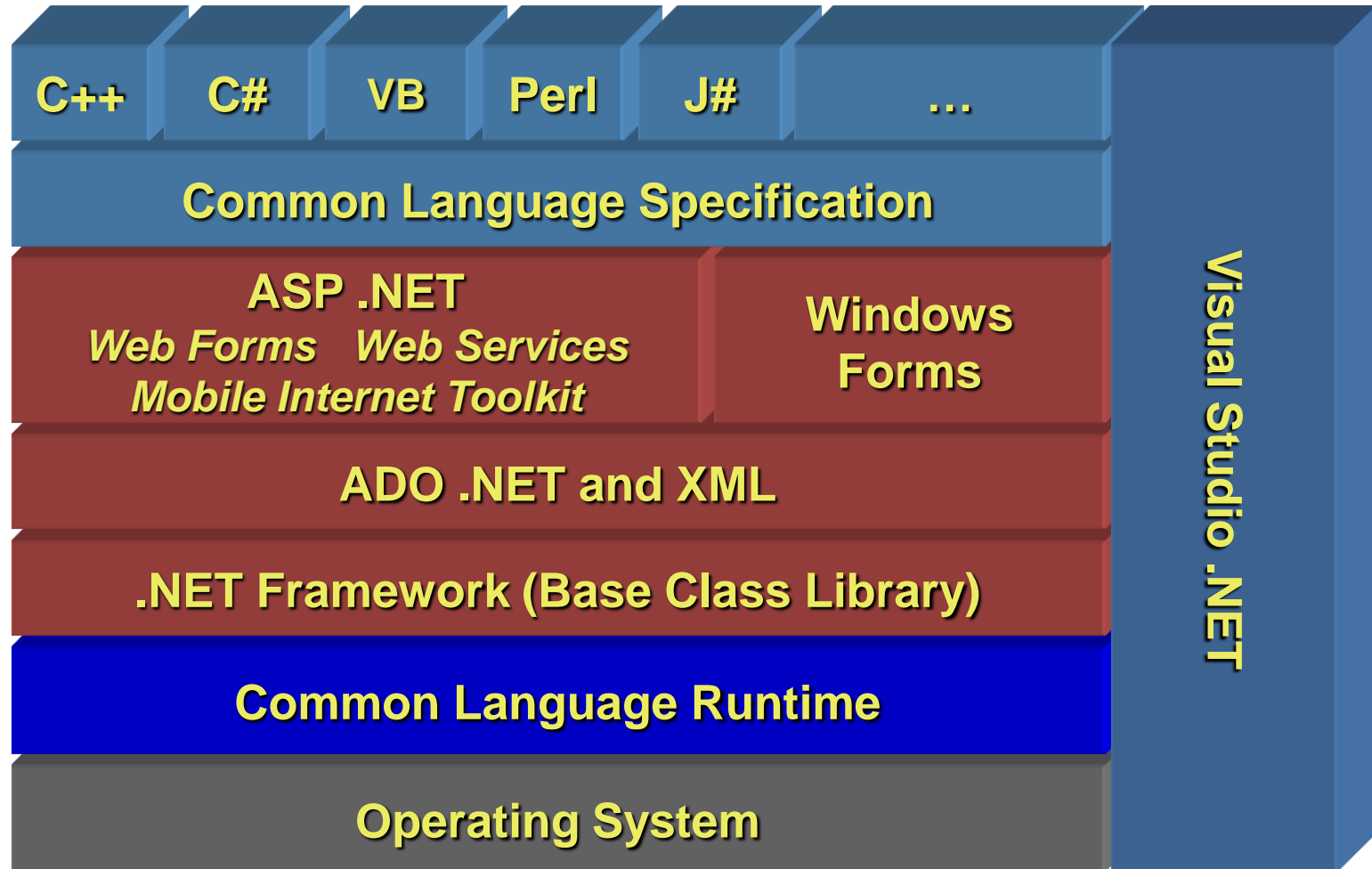
Can any third party language be used with .NET framework?

A. Yes

B. No

# .NET Framework

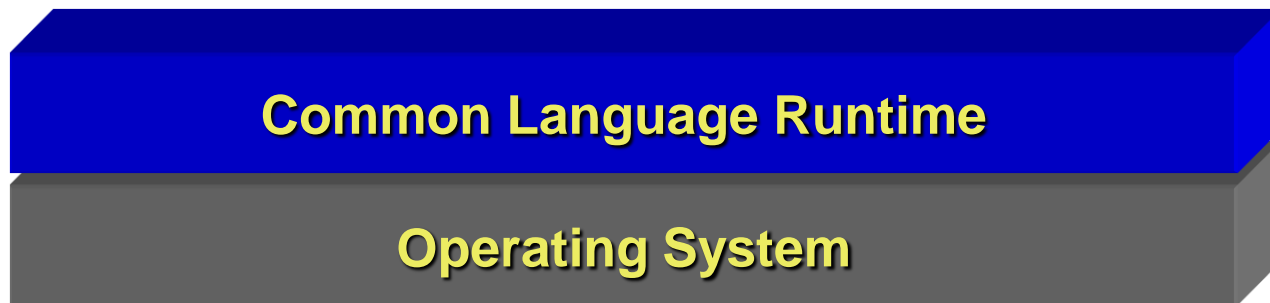
Visual Studio .NET



# .NET Framework

## Common Language Runtime

- Layer between OS and application written in any .NET specific language
- CLR is a runtime environment
- Also supports cross-language interoperability



# .NET Framework

## Common Language Runtime

- Main Function
  - Managed Code → Native Code
  - CLR's JIT converts MSIL code into Native code at runtime



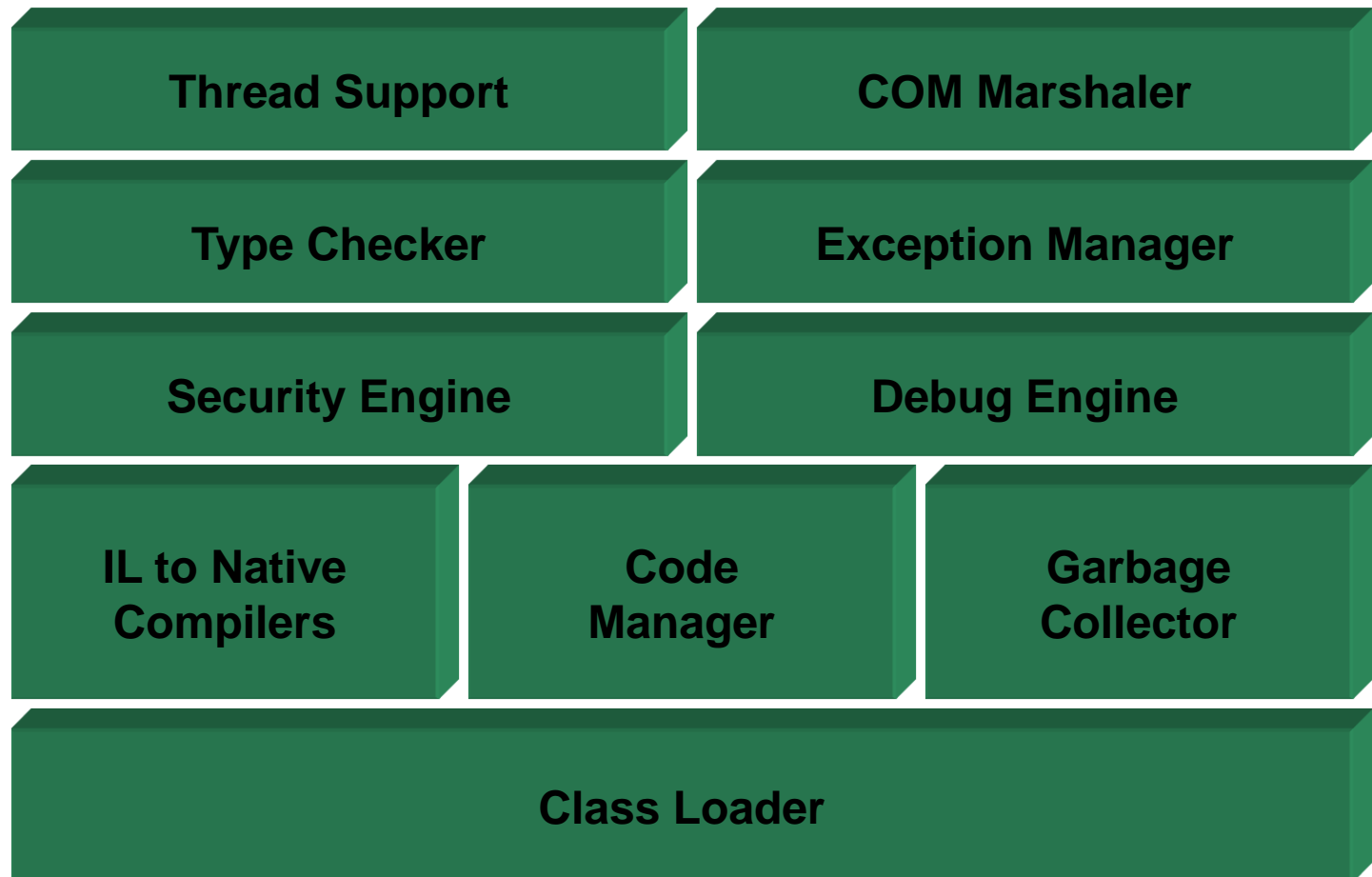
# .NET Framework

## Common Language Runtime

- **Native Code**
  - code compiled to processor-specific machine code.
- **Managed Code**
  - must supply the metadata necessary for the CLR to provide services
  - all code based on Microsoft Intermediate Language (MSIL) executes as managed code.
- **Microsoft Intermediate Language (MSIL)**
  - used as the output of a number of compilers and as the input to a Just-In-Time (JIT) compiler.

# .NET Framework

## Common Language Runtime



Which of the following is not true for CLR of .NET framework?

- A. It supports cross language interoperability.
- B. It is a runtime environment in which programs written in C# and other .NET language are executed.
- C. It provides class for handling database connection and maintenance with front-end .NET technology.
- D. It converts MSIL code into Native code at runtime.
- E. It is virtual machine component of .NET framework.

CLR's \_\_\_\_\_ converts MSIL code into Native code at runtime.

Match the following:

1. Native Code	A. Contains IL and metadata for CLR
2. Managed Code	B. Code compiled to processor specific machine code.
3. IL	C. Output of number of compilers and input to JIT of CLR
4. MSIL	

Which of the following is done by CLR?

- A. Code management (loading and execution)
- B. Application memory isolation
- C. Verification of type safety
- D. Conversion of IL to native code
- E. Access to metadata (enhanced type information)
- F. Managing memory for managed objects
- G. Enforcement of Code Access Security
- H. Exception handling including cross-language exceptions
- I. Interoperation between managed code, COM objects and pre-existing DLLs (unmanaged code and data)
- J. Support for developer services (profiling, debugging, etc.)
- K. Above all, Shaktimaan!

# Managed Code

- Code that targets the CLR is referred to as managed code
- All managed code has the features of the CLR
  - Object-oriented
  - Type-safe
  - Cross-language integration
  - Cross-language exception handling
  - Multiple version support
- Managed code is represented in special Intermediate Language (IL/CIL/MSIL)

# Example of MSIL Code

```
.method private hidebysig static void Main() cil
managed
{
    .entrypoint
    // Code size          11 (0xb)
    .maxstack 8
    IL_0000: ldstr        "Hello, world!"
    IL_0005: call         void
        [mscorlib]System.Console::WriteLine(string)
    IL_000a: ret
} // end of method HelloWorld::Main
```

**IL Disassembler**

**ildasm.exe**

**IL Assembler**

**ilasm.exe**

# IL: Intermediate Language

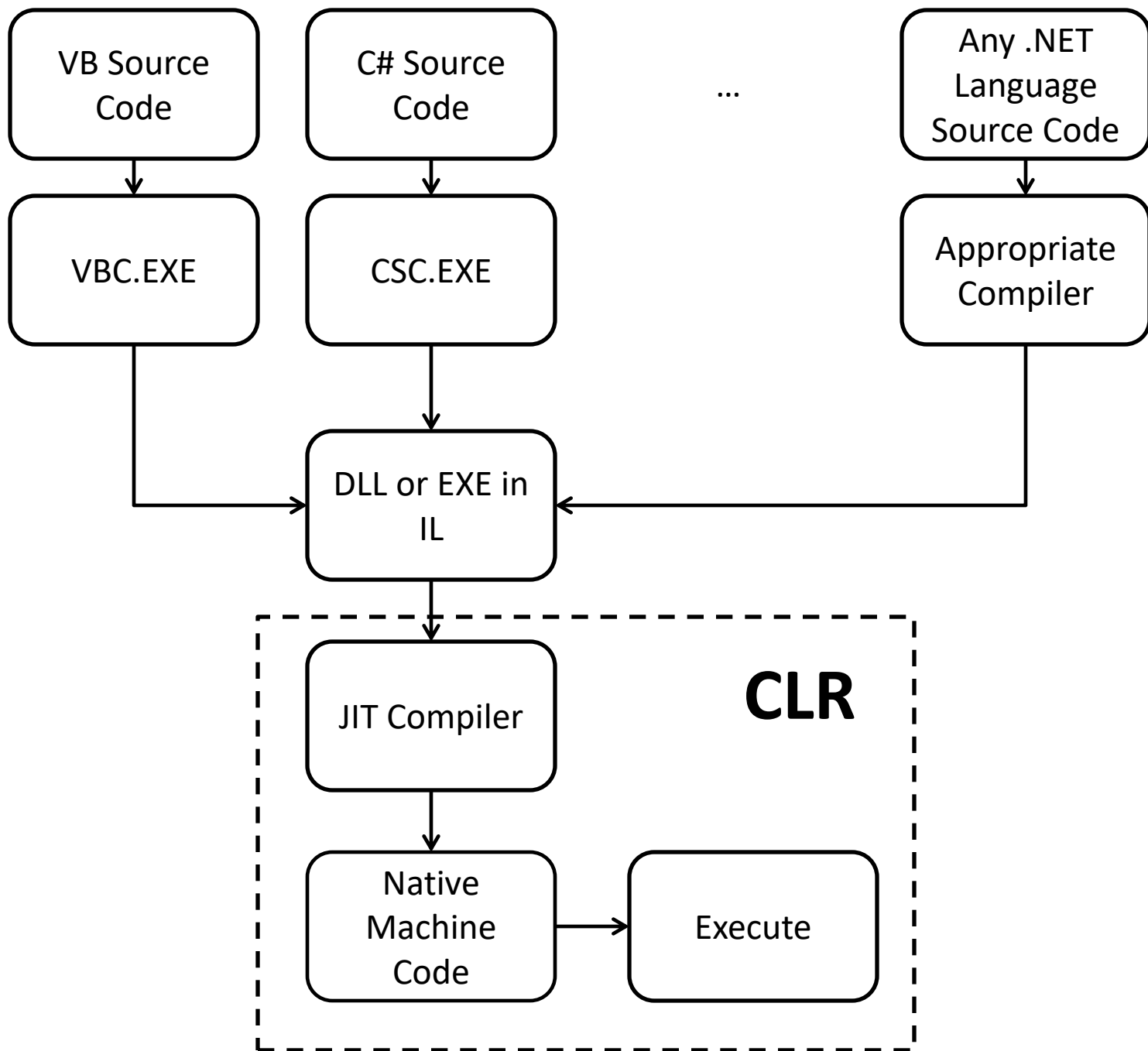
- .NET Language → IL/CIL/MSIL → CLR
- CLS : Common Language Specification
  - Contract ; Specification
  - Defines OO ingredients
- CTS : Common Type System
  - Compatible types on the intermediate level
  - No conversion, calling one language from another



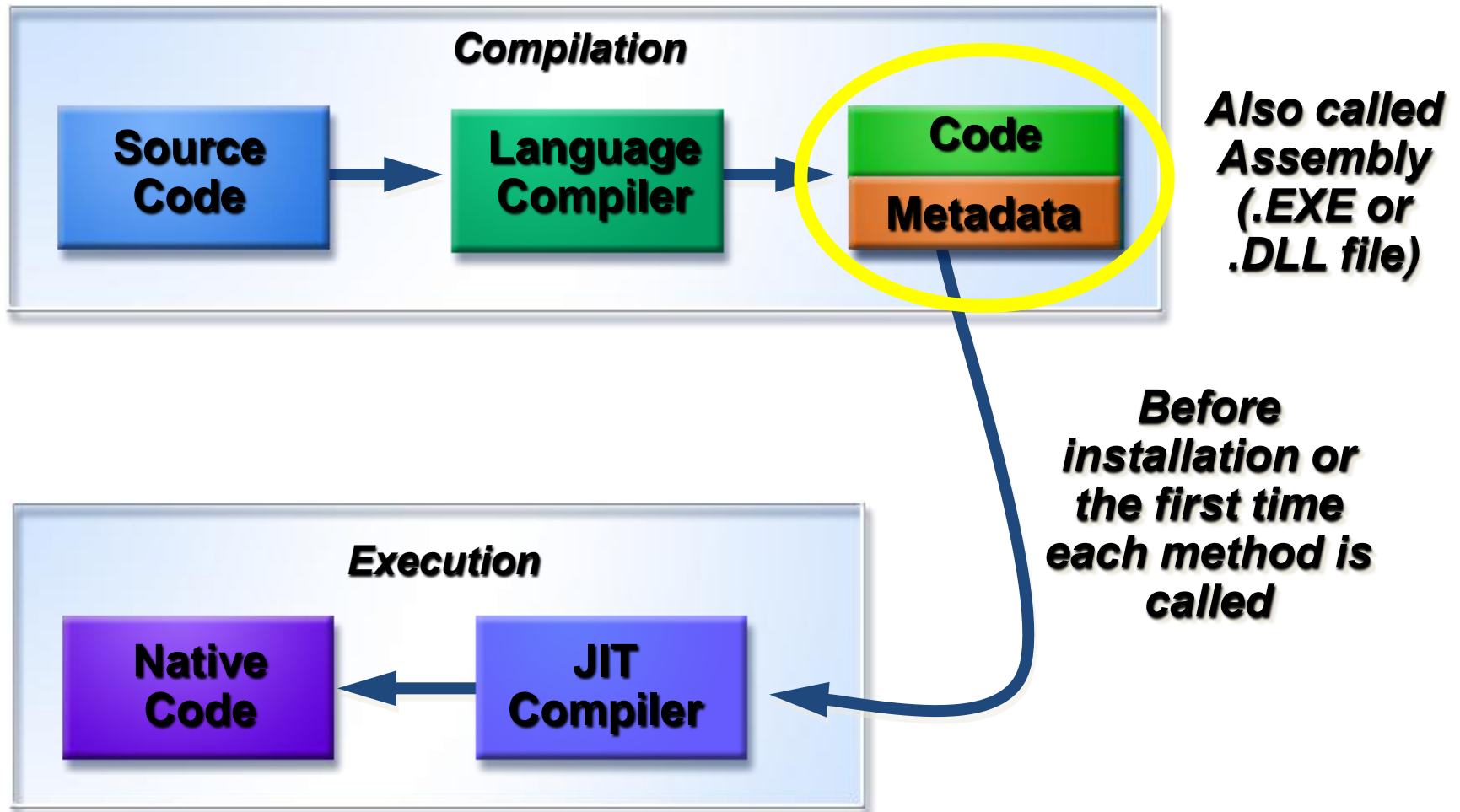
# Cross Language Interoperability

- Ability of code to interact with code that is written by using a different programming languages
  - Maximize code reuse
  - Improved development process
- CLR → CTS
- CLR → CLS

1. Any language that conforms to the CLS is a \_\_\_\_ language.
2. Due to \_\_\_\_, .NET supports cross-language interoperability.
3. An \_\_\_\_ is a computer program that translates human-readable assembly language source code into machine language instructions that can be executed by the computer hardware.



# Code Compilation and Execution



# CLR activities during Execution.

- Source Code → MSIL/CIL
- MSIL is assembled into byte code.
- CIL is then verified for safety during runtime, providing better security and reliability.
- JIT involves turning MSIL into code immediately executable by CPU – Native code
- JIT compiler uses metadata, which is data of data to verify any illegal access and violations appropriately.
- JIT compiles MSIL as and when needed, this saves time and space in memory.
  - But results into performance hit.
- NGEN ( Native Image Generator) compilation eliminates this step at run time. It compiles entire MSIL generated.