

# BÀI 1: LẬP TRÌNH ỨNG DỤNG .NET

Thời gian: 180 phút

Giảng viên: PHẠM PHÚ KHƯƠNG

Email: [phamphukhuong@dtu.edu.vn](mailto:phamphukhuong@dtu.edu.vn)

Điện thoại: 0905635421



# Nội Dung

- ❖ Giới thiệu .NET Framework
- ❖ Tổng quan ngôn ngữ C#





# Giới thiệu .NET Framework

- ❖ CLR, CTS, CLS, MSIL...
- ❖ Garbage collection
- ❖ Namespace



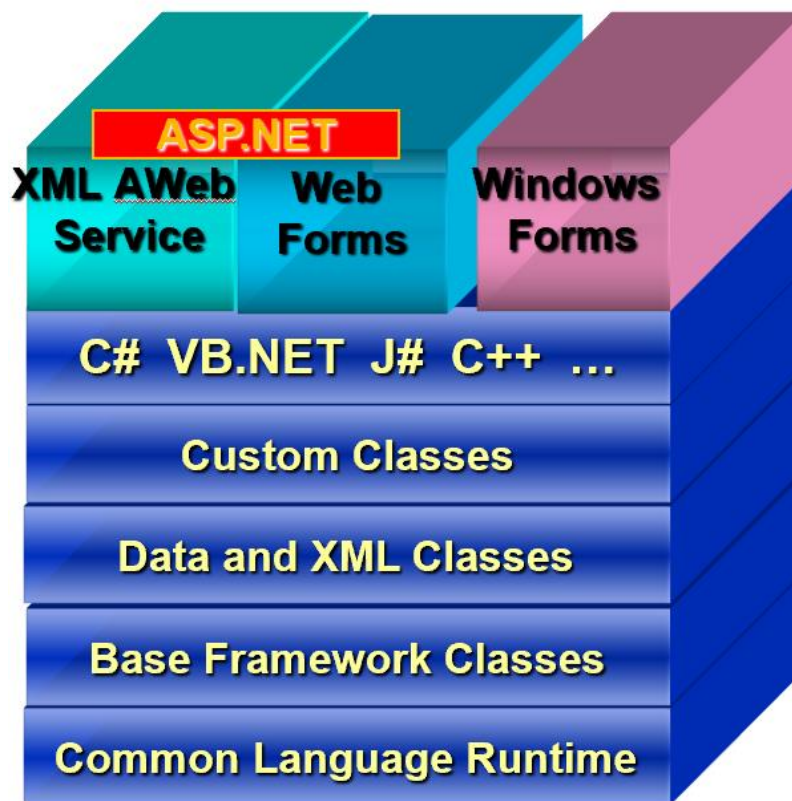
## Video tham khảo

- <https://www.youtube.com/watch?v=fA46IbOJtBQ&t=349s>



# Giới thiệu .NET Framework

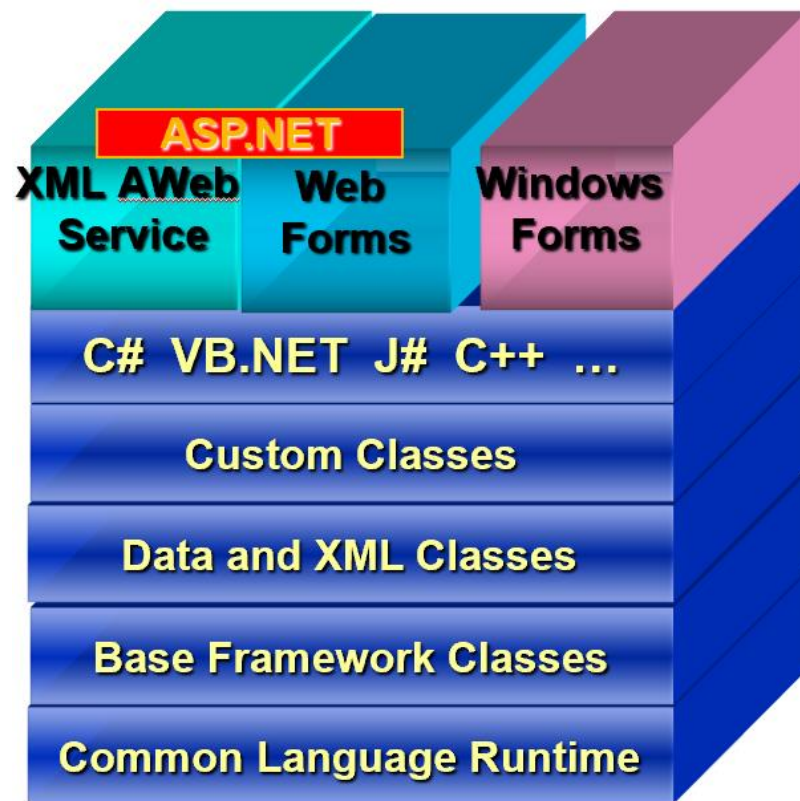
- ❖ Chương trình nền tảng cho công nghệ .NET
- ❖ Cung cấp tập hợp class library thường dùng
- ❖ Quản lý sự thực thi của các chương trình .NET





# Câu hỏi nhỏ?

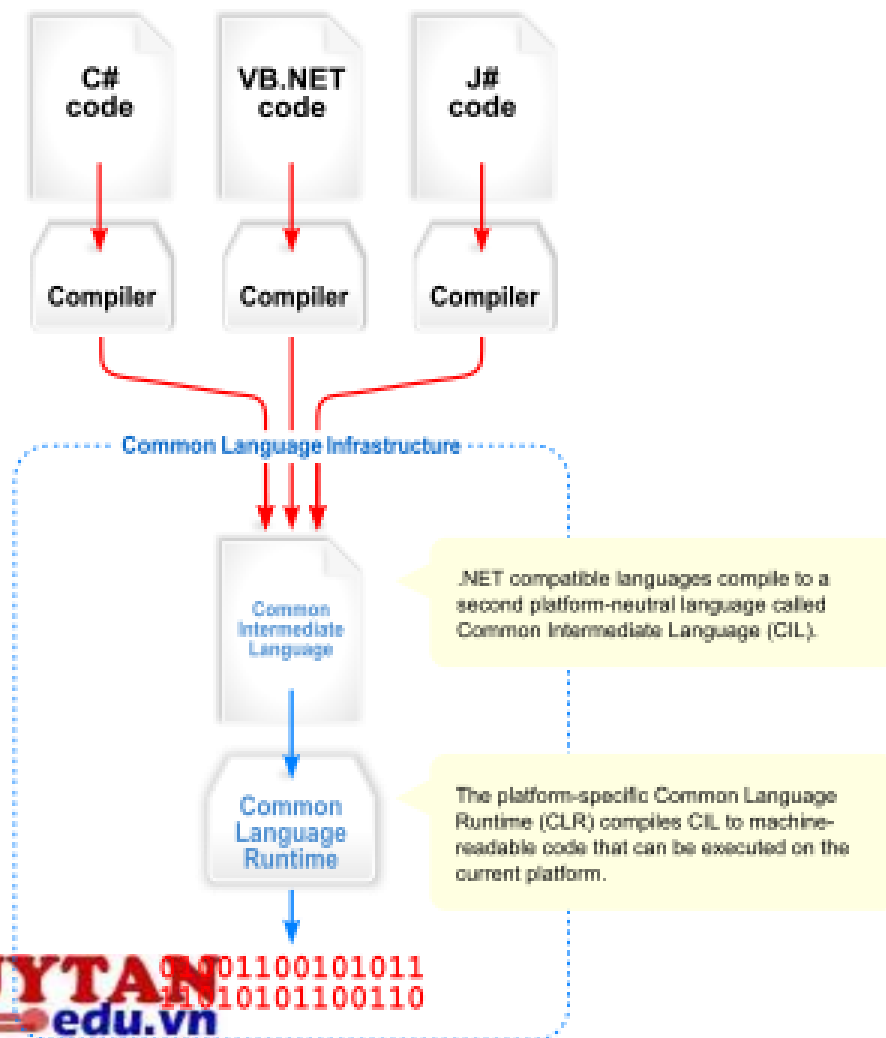
**Câu hỏi:** Hãy so sánh mô hình của Microsoft với mô hình của JAVA mà các em đã học trước đây?





# .NET Framework - Architecture

## ❖ Common Language Infrastructure (CLI)





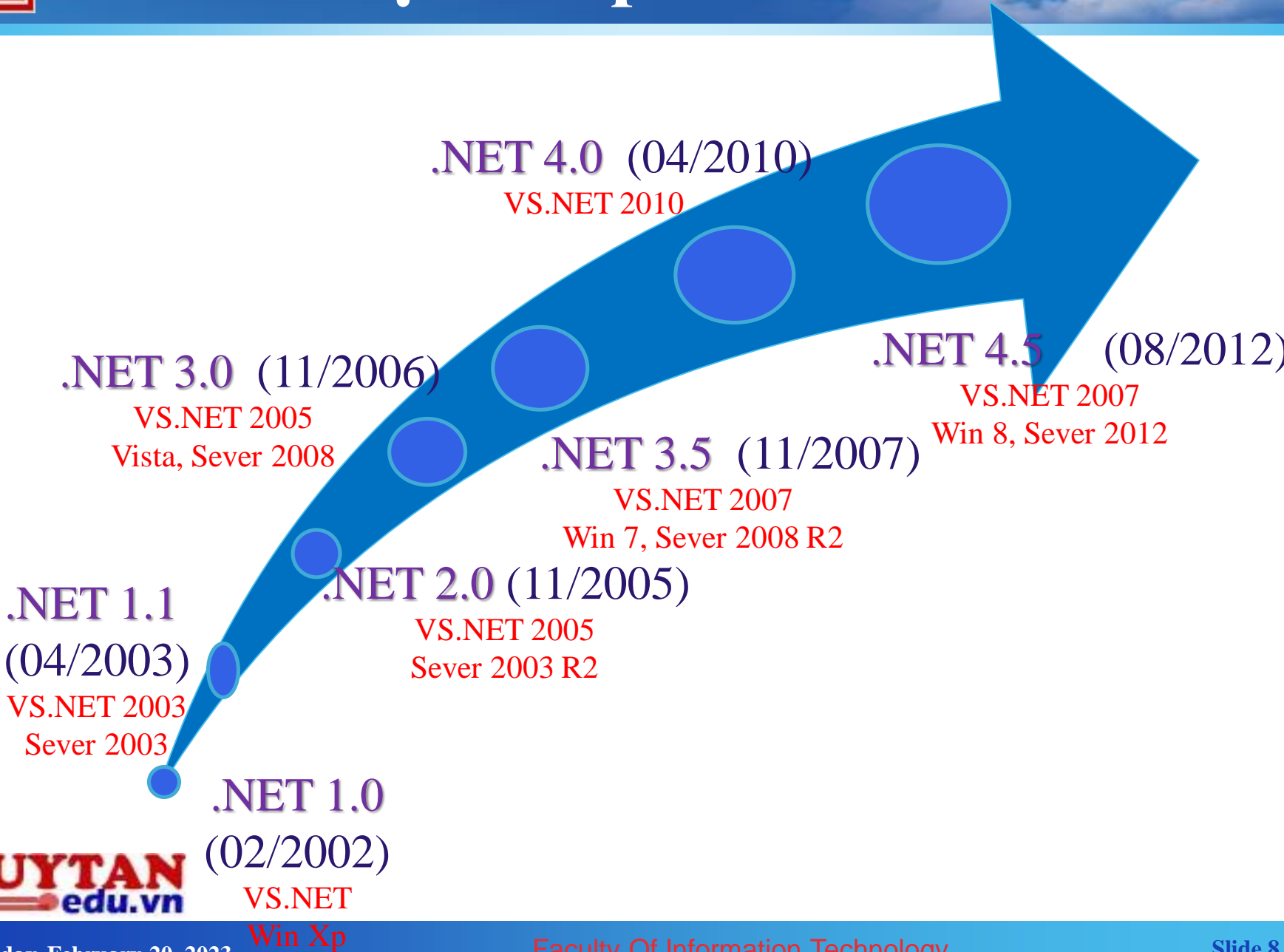
## ❖ Mục đích của CLI

- Cung cấp một nền tảng ngôn ngữ trung lập để phát triển và thực thi ứng dụng, bao gồm cả chức năng để xử lý các lỗi, thu gom rác thải (garbage collection), bảo mật và khả năng tương tác.
- Bằng cách thực hiện những khía cạnh cốt lõi của .NET Framework trong phạm vi của mình, CLI sẽ không được gắn với một ngôn ngữ duy nhất mà sẽ có mặt trên nhiều ngôn ngữ được hỗ trợ bởi .NET framework





# Lịch sử phát triển





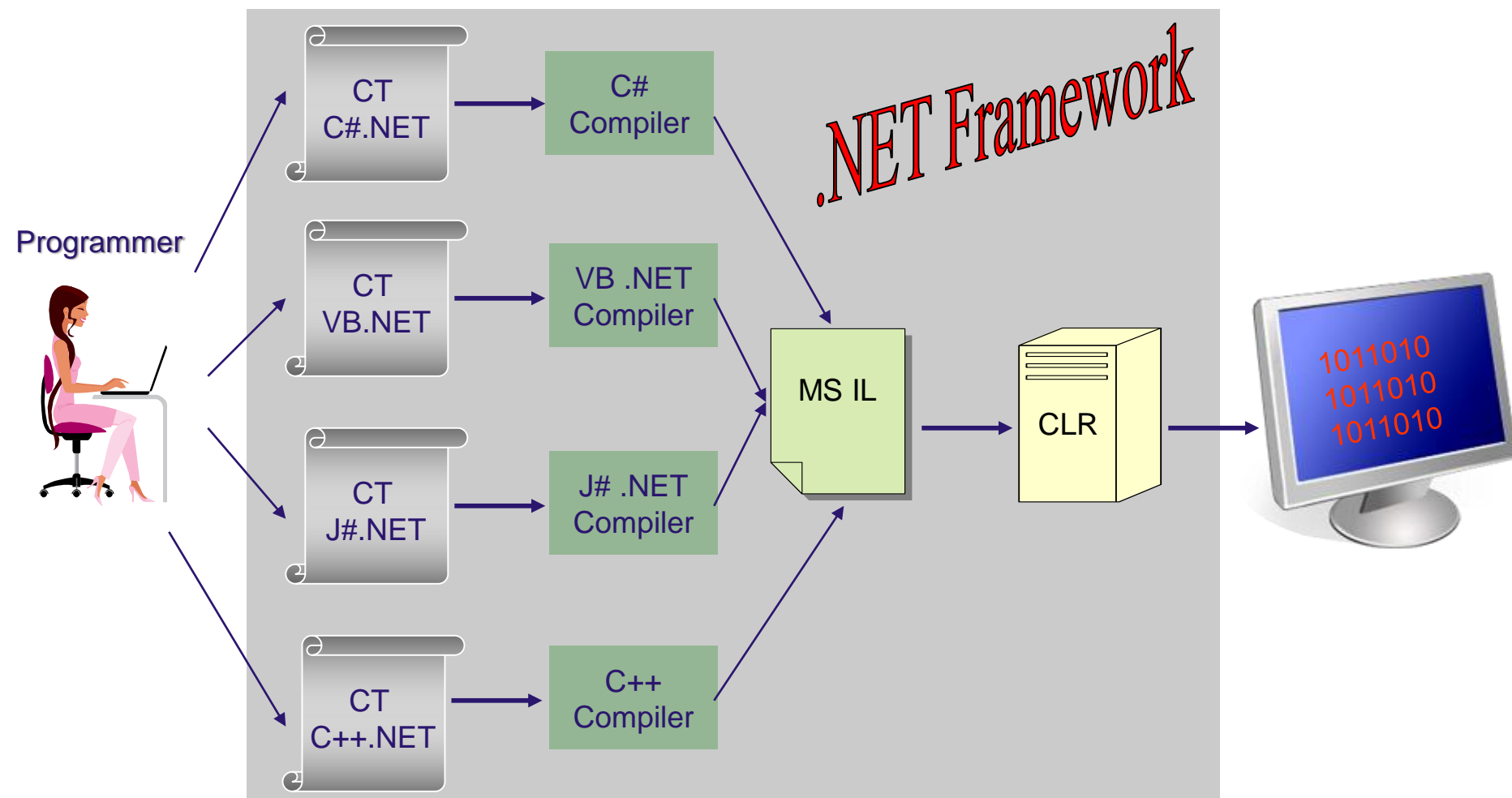


# Đặc điểm của ứng dụng .NET

- ❖ Chạy trên nền (.NET framework)
- ❖ Mã nguồn được biên dịch qua MSIL
- ❖ MSIL được thông dịch qua mã máy lúc thực thi nhờ vào CLR
- ❖ Độc lập nền tảng
  - Về lý thuyết có thể chạy trên mọi nền!
- ❖ Install .NET Framework redistribute packadge  
(dotnetfx.exe) để chạy ứng dụng .NET trên máy client.



# Đặc điểm của ứng dụng .NET





# .NET Framework - CLR

- ❖ Theo quan điểm của người lập trình, .NET có thể hiểu như môi trường thực thi mới và thư viện lớp cơ sở cải tiến.
- ❖ Môi trường thực thi là: Common Language Runtime - CLR
- ❖ Vai trò chính CLR: locate, load, manage .NET types
- ❖ CLR còn quản lý những phần ở mức thấp như: memory management, security check



# .NET Framework - CLR

❖ **CLR là nền tảng của .NET Framework, nó đảm nhận các công việc sau:**

- Là công cụ thực thi mã trung gian (tựa JVM)
- Biên dịch (Just-in-time compiler)
- Thực thi mã nguồn
- Quản lý bộ nhớ



# .NET Framework - CLR

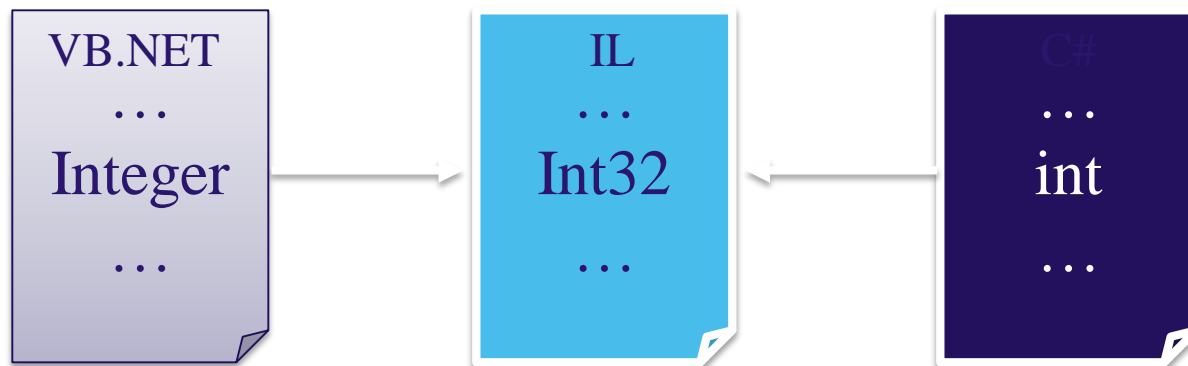
- ❖ **CLR là nền tảng của .NET Framework, nó đảm nhận các công việc sau:**
- Thực thi luồng (Thread execution)
  - Xử lý lỗi (Error-handling)
  - Xác nhận mã nguồn an toàn và các hình thức khác của việc chính xác mã nguồn (managed code).



# .NET Framework - CTS

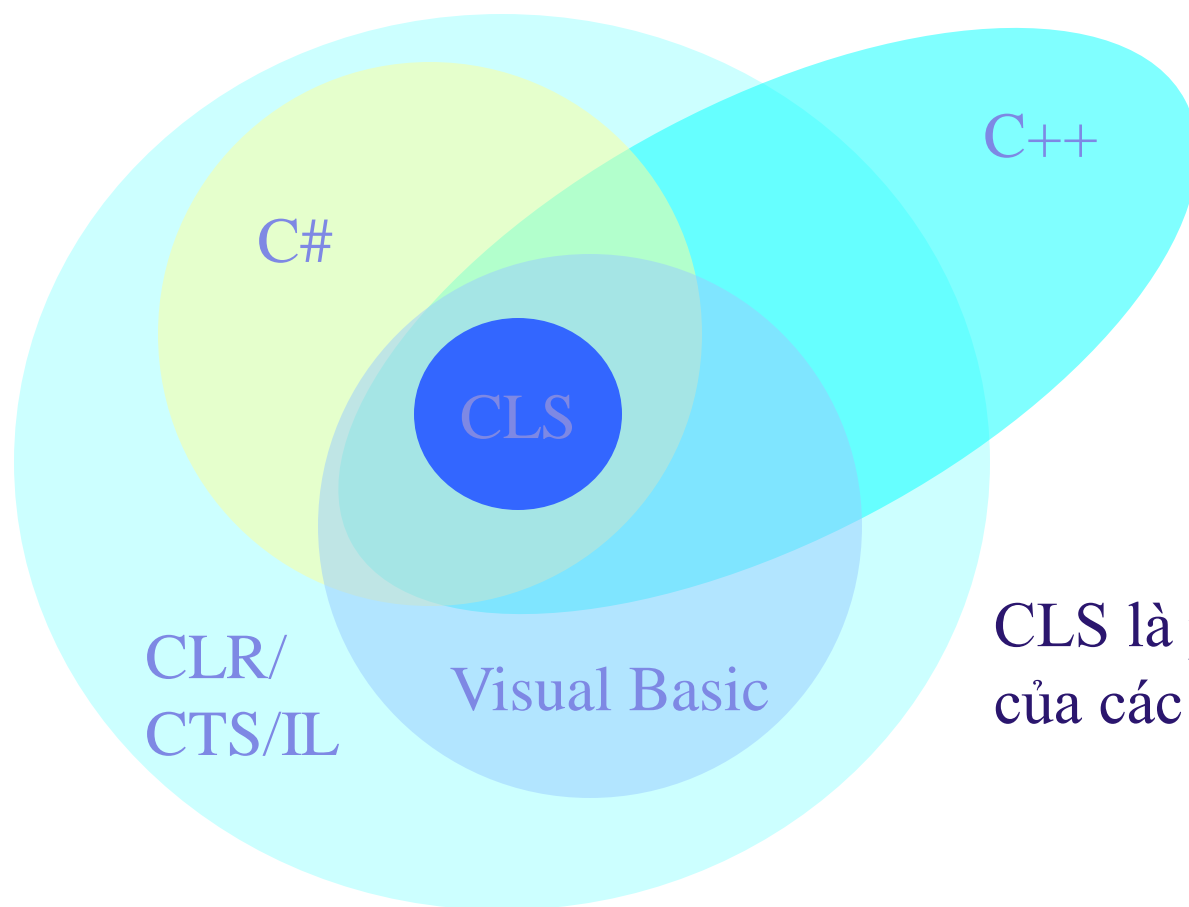
## ❖ Common Type System (CTS):

- Mục đích hỗ trợ thực thi chéo ngôn ngữ
- Định nghĩa kiểu dữ liệu tiên định và có sẵn trong IL:
  - Tất cả ngôn ngữ .NET sẽ được sinh ra mã cuối trên cơ sở kiểu dữ liệu này





# .NET Framework - CLS



CLS là phần chung nhỏ nhất của các ngôn ngữ





# .NET Framework - CLS

## ❖ Common Language Specification:

- Đảm bảo sự thực thi chéo
- Tất cả compiler hướng .NET đều phải tuân thủ theo CLS
  - Có thể viết mã non-CLS nhưng sẽ ko đảm bảo thực thi chéo
  - IL phân biệt loại ký tự, VB.NET ko phân biệt, CLS báo rằng ko cho phép 2 định danh chỉ khác nhau về kiểu ký tự, do đó VB.NET có thể hoạt động trong CLS





# MS Intermediate Language

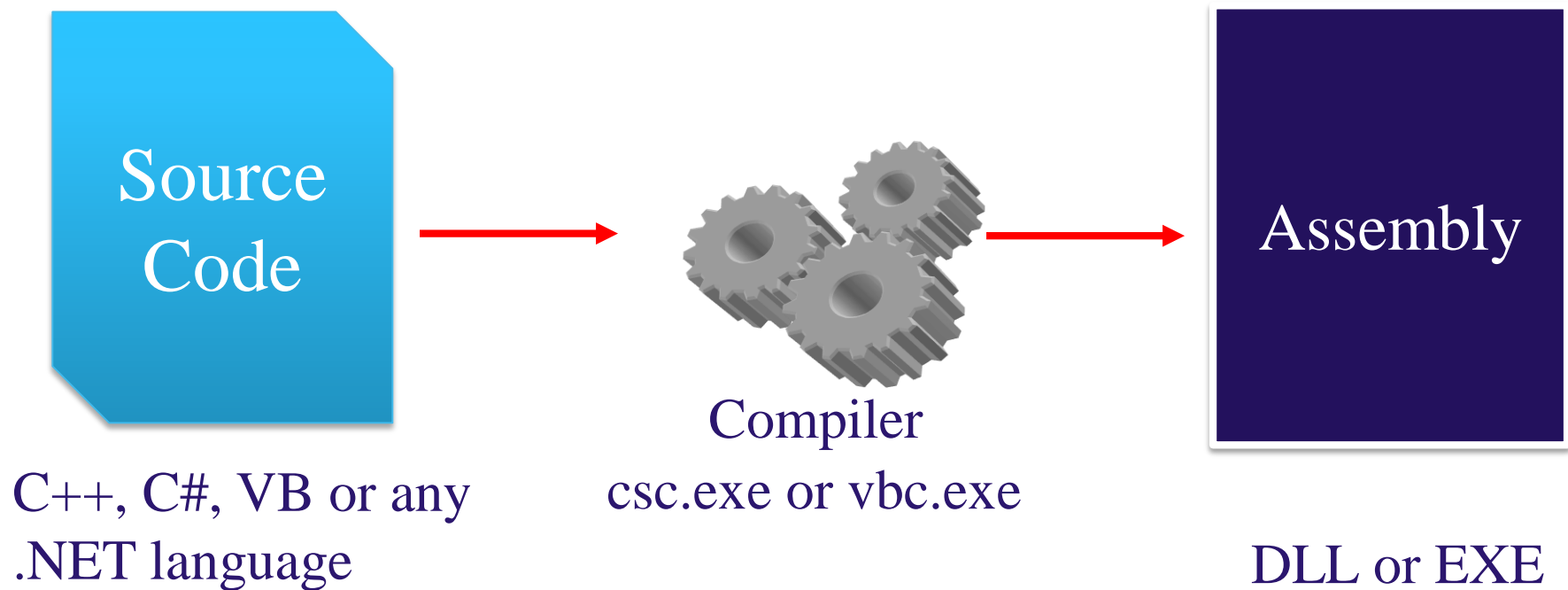
## ❖ IL

- Ngôn ngữ trừu tượng
- Ý tưởng về mã Java byte
- Mã cấp thấp cú pháp đơn giản  $\Rightarrow$  quá trình dịch sang mã máy nhanh hơn

## ❖ CLR chuyển IL thành mã máy lúc runtime

- Sự chuyển này gọi là **Just – In – Time** Compilation hay JIT compiling

# Common Language Runtime - compilation





# Assembly

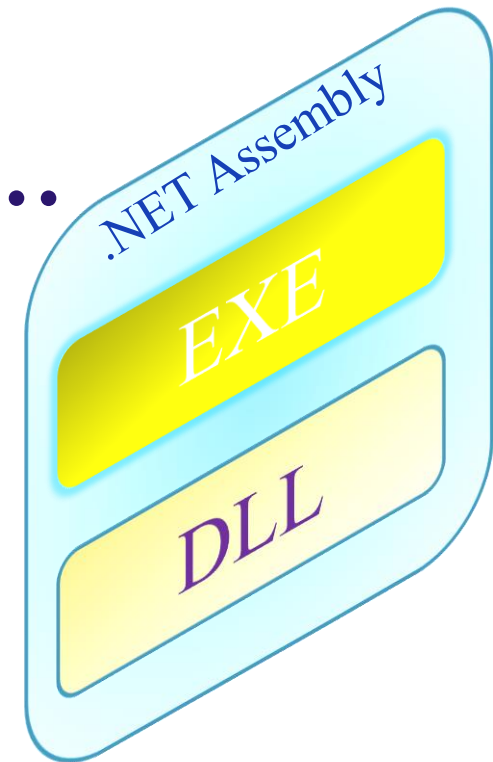
## ❖ Managed Modules

- Chứa IL code để thực thi

## ❖ Chứa resource, metadata...

## ❖ Portable Executable (PE)

- Process assembly (EXE)
- Library assembly (DLL)





# Managed Code

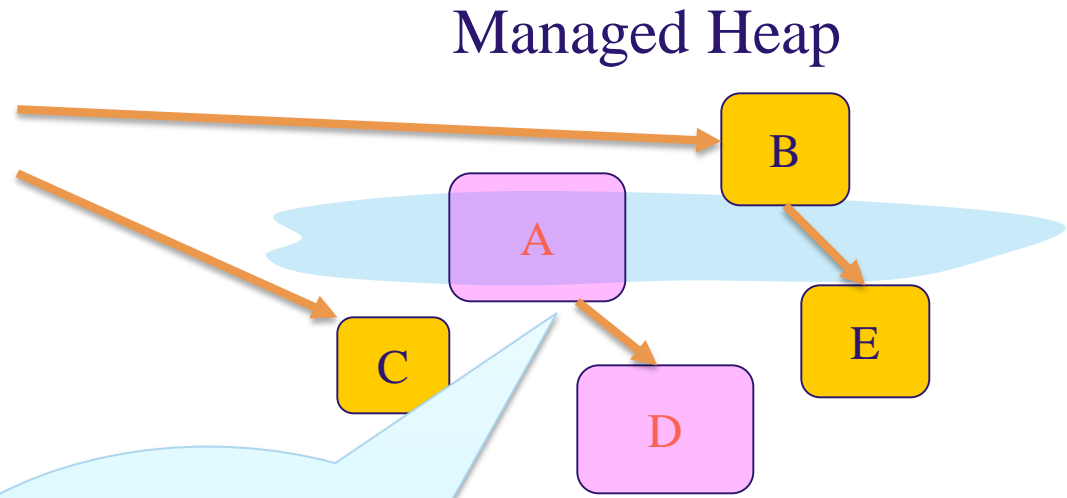
- ❖ Phần mềm được viết để sử dụng trong .NET Framework
- ❖ PM khác thì gọi là Unmanaged code
- ❖ “Managed”: do chạy dưới sự giám sát của cơ chế thực thi (CLR)





# Garbage collection

```
class myClass{  
    void Method(){  
  
        variable v1;  
        variable v2;  
  
        do{  
            ....  
        }  
    }  
}
```



A và D sẽ bị xóa  
do ko còn tham  
chiếu hay truy  
cập tới



# Garbage collection

- ❖ GC xuất hiện (ko định trước) khi ko đủ bộ nhớ để cung cấp cho ứng dụng.
- ❖ GC thực hiện việc tìm kiếm những đối tượng trong managed heap, xoá nếu ko còn tham chiếu tới.
- ❖ Có thể gọi GC một cách tường minh





# Namespace

- ❖ Tránh xung đột tên
- ❖ Cho phép nest
- ❖ Truy cập đầy đủ qua tên
- ❖ Tất cả data type có tiếp đầu ngữ là tên namespace

## Namespace CS1

Class A

Class B

Class C

CS1.A...

CS2.A...

## Namespace CS2

Class A

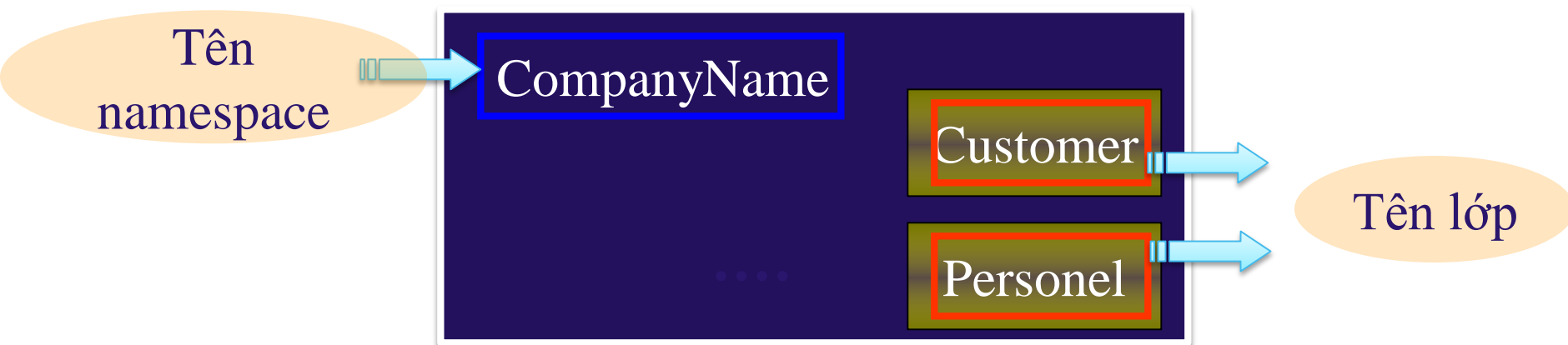
Class B

Class C



# Namespace

- ❖ Hầu hết các lớp cơ sở chung của .NET đều thuộc namespace System
  - Lớp CS Array thuộc System → System.Array
- ❖ .NET đề nghị tất cả kiểu do user định nghĩa phải nằm trong 1 namespace

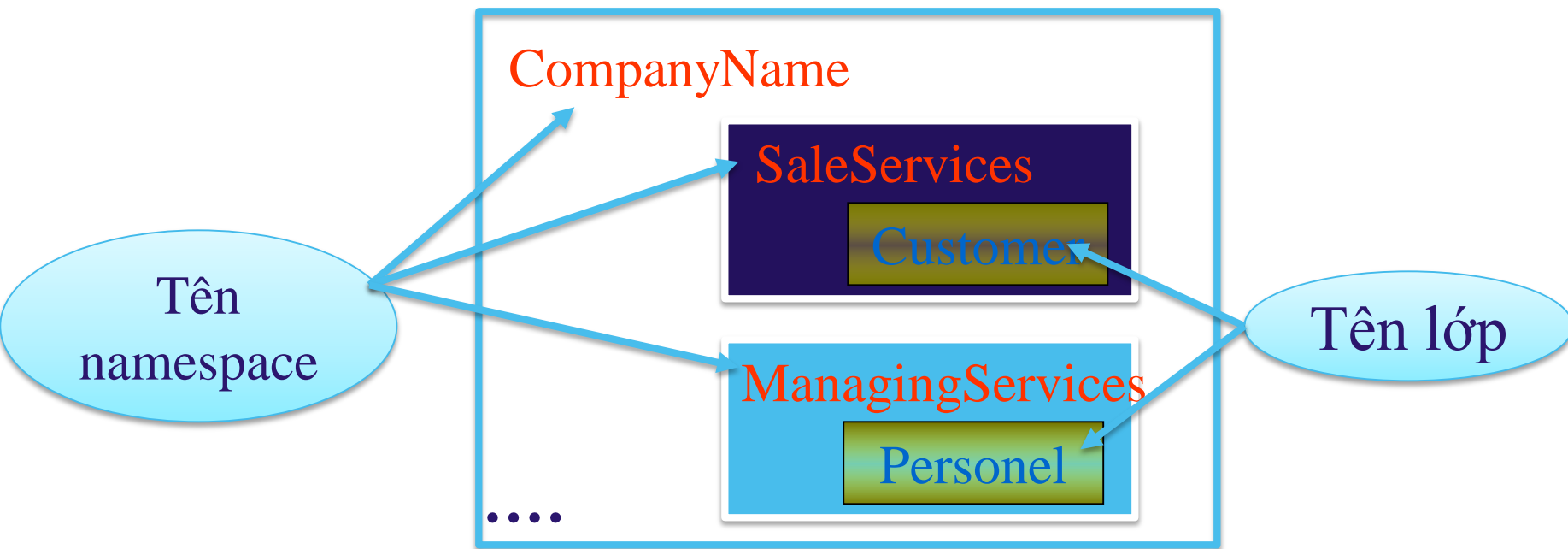




# Namespace

## ❖ Nên phân thành 2 cấp:

- Công nghệ
- Gói phần mềm chức năng





# Các lớp .NET Framework

- ❖ Thư viện lớp cơ sở .NET là managed code
- ❖ Khá đa dạng & đầy đủ:
  - Cho phép kế thừa để phát triển UD
- ❖ Bao bọc tất cả các hàm API
  - Dễ sử dụng (khác với VC++ trước đây)



# Các lớp .NET Framework

## ❖ Các lớp .NET bao gồm các vấn đề

- Đặc tính lõi cung cấp IL, kiểu dữ liệu trong CTS
- Hỗ trợ Win GUI và control
- WebForm (ASP.NET)
- Data Access (ADO.NET)
- Directory Access



# Các lớp .NET Framework

## ❖ Các lớp .NET bao gồm các vấn đề

- File System, registry access
- Networking and web browsing
- .NET attributes and reflection
- WinOS access
- COM interoperability



# Framework (Base) Class Library

## ❖ Các Namespace cơ bản của FCL

Namespace	Description
System.Math	Chứa lớp toán học, chuyển đổi dữ liệu
System.IO	Các lớp cho thao tác Input và Output
System.Net	Các lớp liên quan đến network protocol
System.Collections	Chức các lớp liên quan đến xử lý tập hợp
System.Data	Các lớp của ADO.NET
System.Drawing	Các lớp thực thi chức năng GUI
System.Threading	Các lớp lập trình MultiThread
System.Web	Các lớp liên quan đến HTTP protocol
System.Xml	Các lớp liên quan XML





# Tóm tắt

- .NET Framework : nền tảng cho ứng dụng mới của MS
- Tất cả các chương trình viết bằng ngôn ngữ khác nhau (trên .NET) sẽ được chuyển về mã thống nhất MSIL
- Cho phép thực thi chéo giữa các ngôn ngữ
- Khái niệm Managed Code
- Cơ chế thu gom vùng nhớ tự động
- Các lớp thư viện .NET phong phú & mạnh mẽ.



# TỔNG QUAN NGÔN NGỮ C#



# Ngôn ngữ C#

## ❖ Ngôn ngữ ra đời cùng với .NET

- Kết hợp C++ và Java
- Hướng đối tượng
- Hướng thành phần
- Mạnh mẽ (robust) và bền vững (durable)
- Anders Hejlsberg và MS team xây dựng C#



# Ngôn ngữ C#

- ❖ Mọi thứ trong C# đều Object oriented
  - Kể cả kiểu dữ liệu cơ bản
- ❖ Chỉ cho phép **đơn kế thừa**
  - Dùng interface để khắc phục
- ❖ Lớp **Object** là cha của tất cả các lớp
  - Mọi lớp đều dẫn xuất từ Object (System.Object)



# Ngôn ngữ C#

- ❖ Cho phép chia chương trình thành các thành phần nhỏ độc lập nhau
- ❖ Mỗi lớp gói gọn trong một file, không cần file header như C/C++
- ❖ Bổ sung khái niệm namespace để gom nhóm các lớp
- ❖ Bổ sung khái niệm "*property*" cho các lớp
- ❖ Khái niệm delegate & event



# C# - mạnh mẽ & bền vững

## ❖ Garbage Collector

- Tự động thu hồi vùng nhớ không dùng

## ❖ Kiểm soát và xử lý ngoại lệ exception

- Đoạn mã bị lỗi sẽ không được thực thi

## ❖ Type – safe

- Không cho gán các kiểu dữ liệu khác nhau

## ❖ Versioning

- Đảm bảo sự tương thích giữa lớp con và lớp cha



# Vai trò C# trong .NET Framework

- ❖ **.NET runtime sẽ phổ biến và được cài trong máy client**
  - Việc cài đặt App C# như là tái phân phối các thành phần .NET
  - Nhiều App thương mại sẽ được cài đặt bằng C#
- ❖ **C# tạo cơ hội cho tổ chức xây dựng các App Client/Server n-tier.**
- ❖ **Kết nối ADO.NET cho phép truy cập nhanh chóng & dễ dàng với SQL Server, Oracle...**





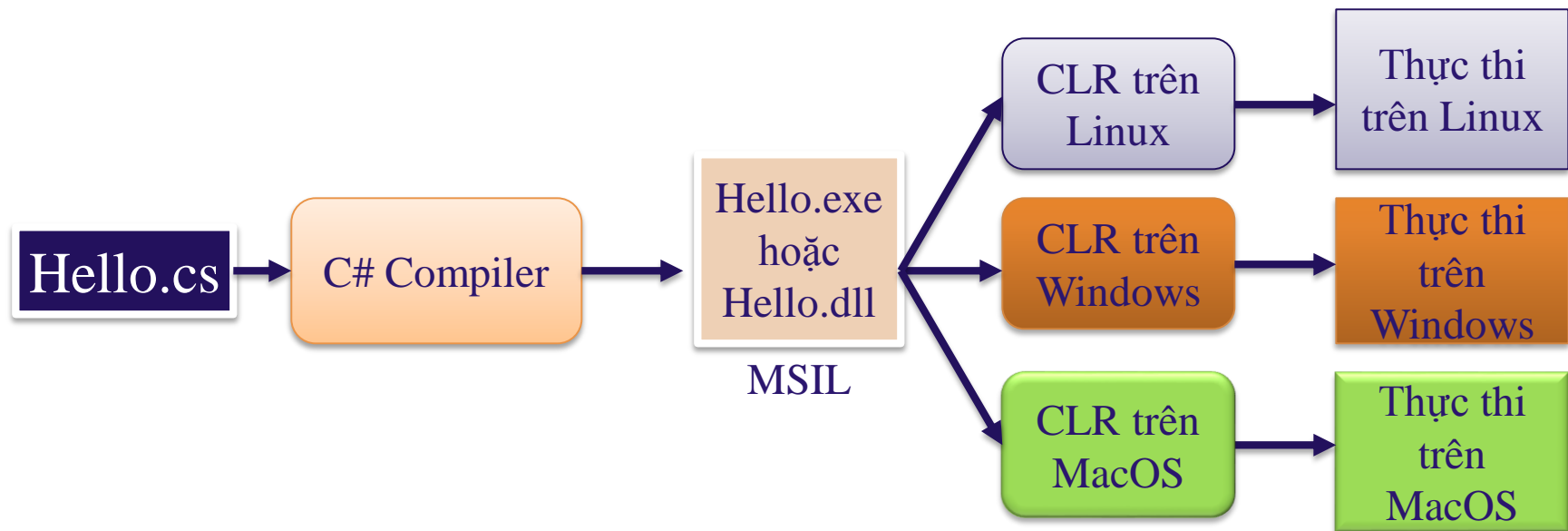
# Vai trò C# trong .NET Framework

- ❖ Cách tổ chức .NET cho phép hạn chế những vấn đề phiên bản
  - Loại bỏ “*DLL Hell*”...
- ❖ ASP.NET viết bằng C#
  - GUI thông minh và chạy nhanh hơn (đặc tính của .NET)
  - Mã ASP.NET ko còn là mới hỗn độn
  - Khả năng bấy lỗi tốt, hỗ trợ mạnh trong quá trình xây dựng App Web.



# Quá trình dịch CT C#

- ❖ Mã nguồn C# (tập tin \*.cs) được biên dịch qua MSIL
  - MSIL: tập tin .exe hoặc .dll
- ❖ MSIL được CLR thông dịch qua mã máy
  - Dùng kỹ thuật JIT (just-in-time) để tăng tốc độ





# Các loại ứng dụng C#

## ❖ Chương trình Console (TUI)

- Giao tiếp với người dùng bằng bàn phím
- Không có giao diện đồ họa (GUI)

## ❖ Chương trình Windows Form

- Giao tiếp với người dùng bằng bàn phím và mouse
- Có giao diện đồ họa và xử lý sự kiện

## ❖ Chương trình Web Form

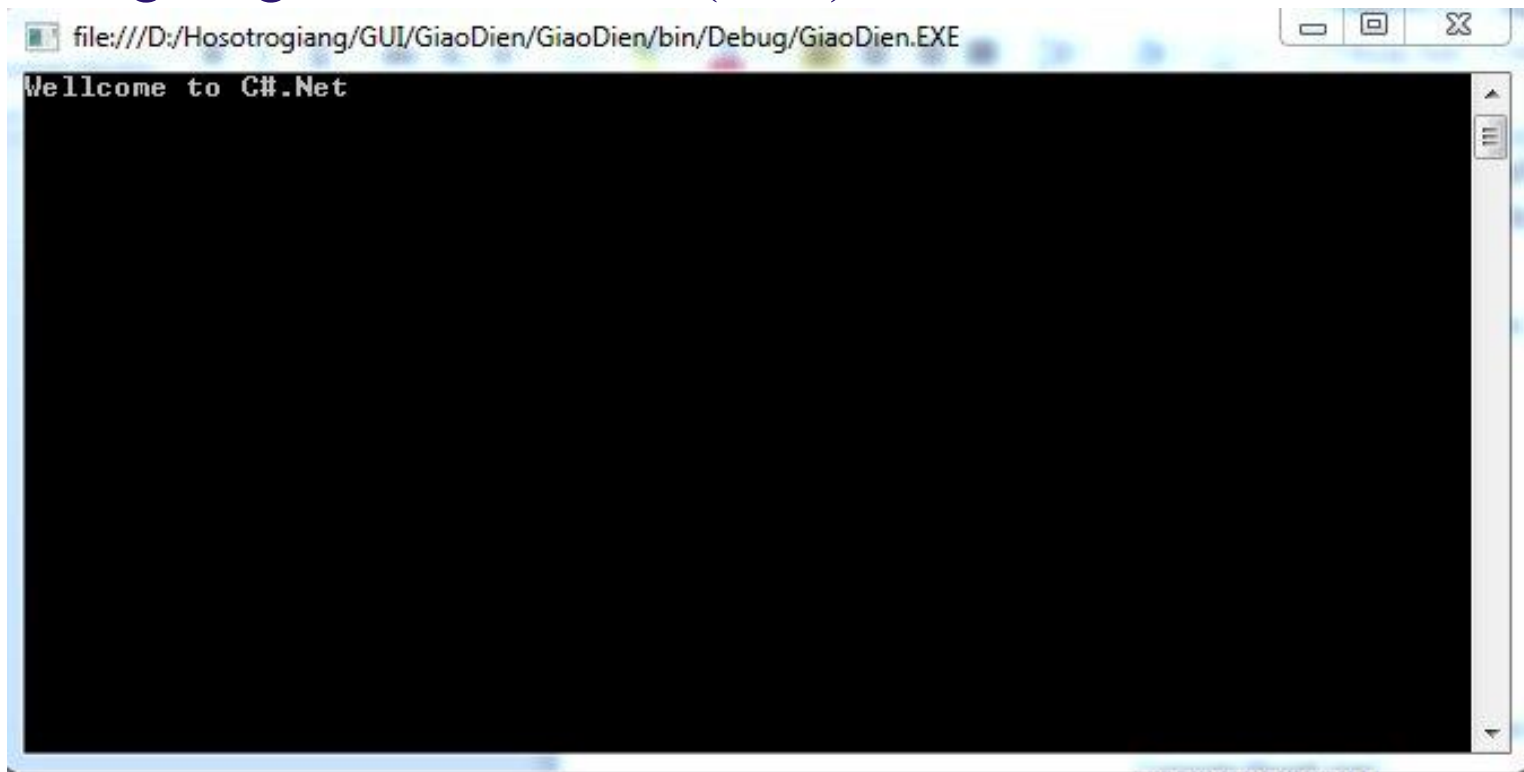
- Kết hợp với ASP .NET, C# đóng vai trò xử lý bên dưới (underlying code)
- Có giao diện đồ họa và xử lý sự kiện



# Các loại ứng dụng C#

## ❖ Chương trình Console (TUI)

- Giao tiếp với người dùng bằng bàn phím
- Không có giao diện đồ họa (GUI)





# Các loại ứng dụng C#

## ❖ Chương trình Windows Form

- Giao tiếp với người dùng bằng bàn phím và mouse
- Có giao diện đồ họa và xử lý sự kiện





# Các loại ứng dụng C#

## ❖Chương trình Web Form

- Kết hợp với ASP .NET, C# đóng vai trò xử lý bên dưới (underlying code)
- Có giao diện đồ họa và xử lý sự kiện





# Cấu trúc chương trình C#

## ❖ Chương trình C# đơn giản

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HelloWorld
{
    References
    class Program
    {
        References
        static void Main(string[] args)
        {
            Console.WriteLine("Wellcome to C#.NET");
            Console.ReadLine();
        }
    }
}
```



# Cấu trúc chương trình C#

- ❖ Phần khai báo dùng namespace (option)

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

- ❖ Phần định nghĩa namespace và lớp

```
namespace HelloWorld  
{  
    References  
    class Program  
    {  
        References  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Wellcome to C#.NET");  
            Console.ReadLine();  
        }  
    }  
}
```





# Cấu trúc chương trình C#

- ❖ Các câu lệnh được viết trong thân của phương thức (ở đây là phương thức Main)
- ❖ Thực hiện một công việc nào đó
- ❖ Kết thúc bởi dấu chấm phẩy (;)

```
namespace HelloWorld
{
    References
    class Program
    {
        References
        static void Main(string[] args)
        {
            Console.WriteLine("Wellcome to C#.NET");
            Console.ReadLine();
        }
    }
}
```

Phương thức Main →

Các câu lệnh →



# Khoảng trắng

## ❖ Bao gồm

- Ký tự trắng, ký tự xuống dòng, ký tự tab
- Dòng trống

## ❖ Sử dụng hợp lý $\Rightarrow$ chương trình dễ đọc

```
namespace HelloWorld
{class Program
{
static void Main(string[] args)
{
    Console.WriteLine("Hello World!");
    Console.ReadLine();}
}
```

```
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            Console.ReadLine();
        }
    }
}
```



# Chú thích

- ❖ Chú thích (comment) được dùng để giải thích về chương trình và các câu lệnh
- ❖ Giúp cho chương trình dễ hiểu hơn
- ❖ Được bỏ qua khi biên dịch
- ❖ Không ảnh hưởng tới kết quả thực thi của chương trình
- ❖ Có thể phát sinh ra documentation của chương trình qua **chú thích XML**



# Hai cách tạo chú thích cơ bản

- ❖ Gõ phần chú thích sau cặp ký tự //
- ❖ Gõ phần chú thích giữa cặp ký tự /\* và \*/

```
/* Chương trình C# đầu tiên
   In ra câu chào "Hello World" */

using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!"); // Xuất ra câu chào
            Console.ReadLine(); // Chờ nhấn Enter
        }
    }
}
```



# XML Comment

- ❖ Cho phép phát sinh ra sơ liệu dạng XML
- ❖ Thích hợp cho việc viết sơ liệu của dự án lớn
- ❖ Chú thích XML bắt đầu với triple slash (“///”) và các tag của XML
- ❖ Chú thích XML dùng cho
  - User defined types
  - Class, delegate, enum and struct
  - Member of user defined types



# XML Comment

```
using System;
namespace XMLCommentDemo
{
    public class Temperature
    {
        public static int CelsiusToFahrenheit(int degreesCelsius)
        {
            return ((int)((9/5)*degreesCelsius) + 32);
        }

        public static int FahrenheitToCelsius(int
degressFahrenheit)
        {
            return ((int)((5/9)*(degressFahrenheit - 32)));
        }
    }
}
```

*C# Code without XML Comment*



# XML Comment

## ❖ Comment for class

```
/// <summary>  
/// Class temperature provides functions which convert  
/// among various temperature scales.  
/// </summary>  
public class Temperature
```

## ❖ Comment for method

```
/// <summary>  
/// Converts degrees Celsius to degrees Fahrenheit  
/// </summary>  
/// <param name="degreesCelsius">Degrees Celsius</param>  
/// <returns>Returns degrees Fahrenheit</returns>  
public static int CelsiusToFahrenheit(int degreesCelsius)
```



# XML Comment

```
using System;
namespace XMLCommentDemo{
    /// <summary>
    /// Class temperature provides functions which convert among various
    /// temperature scales.
    /// </summary>
    public class Temperature {
        /// <summary>
        /// Converts degrees Celsius to degrees Fahrenheit
        /// </summary>
        /// <param name="degreesCelsius">Degrees Celsius</param>
        /// <returns>Returns degrees Fahrenheit</returns>
        public static int CelsiusToFahrenheit(int degreesCelsius) {
            return ((int)((9/5)*degreesCelsius) + 32);
        }
        /// <summary>
        /// Converts degrees Fahrenheit to degrees Celsius
        /// </summary>
        /// <param name="degressFahrenheit">Degrees Fahrenheit</param>
        /// <returns>Returns degrees Celsius</returns>
        public static int FahrenheitToCelsius(int degressFahrenheit) {
            return ((int)((5/9)*(degressFahrenheit - 32)));
        }
    }
}
```

*C# Code with XML Comment*





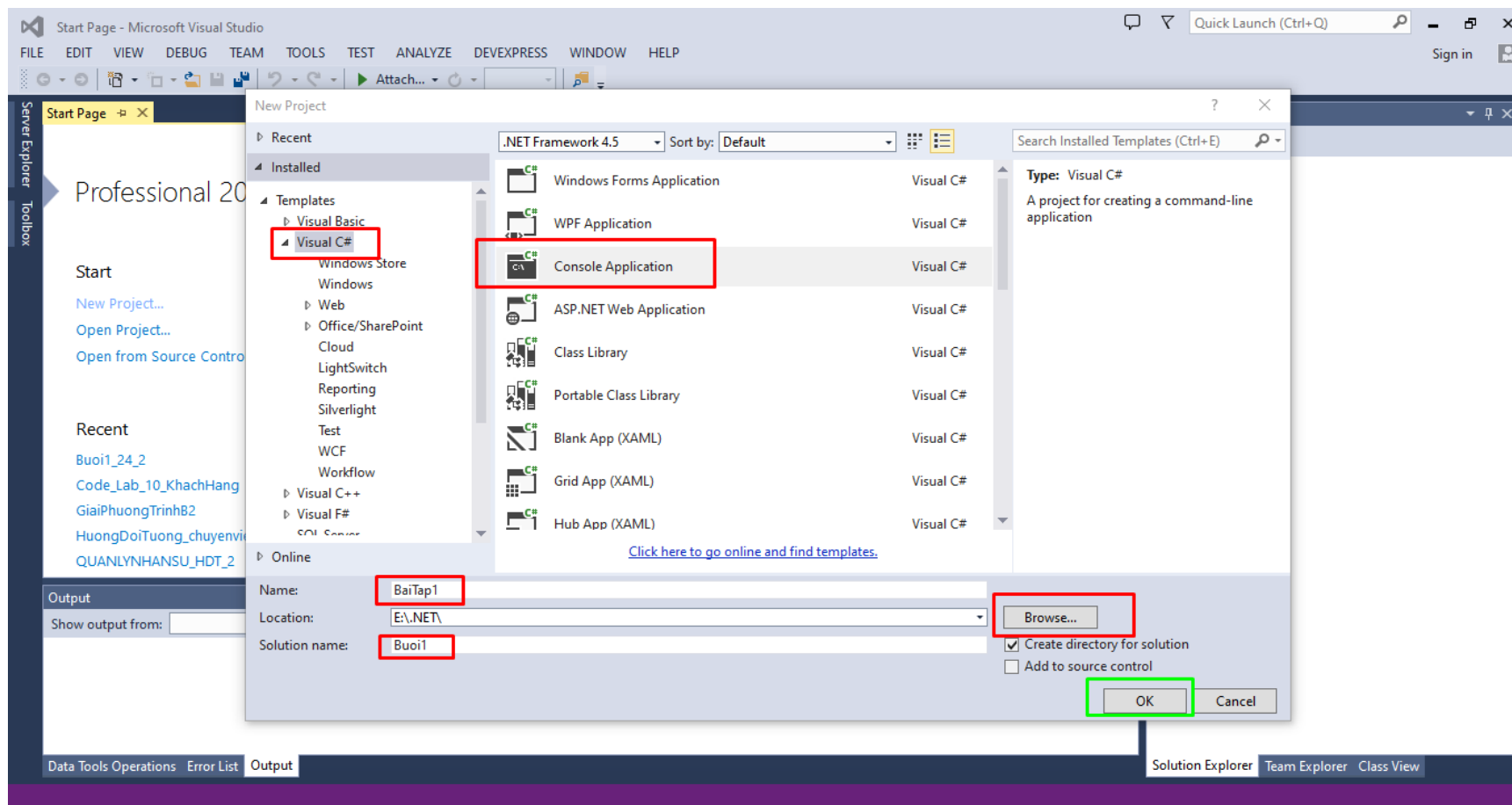
# Biên dịch chương trình C#

- Bước 1: Khởi động Visual Studio 2013
- Bước 2: Vào menu File > New Project



# Biên dịch chương trình C#

## • Bước 3: Khai báo





# Biên dịch chương trình C#

- ❖ Mặc định: Visual Studio 2008 sẽ tạo ra một tập tin **Program.cs** chứa một namespace tên Hello và trong namespace này chứa một class tên **Program**
  - Bước 4: Trong phương thức Main, gõ đoạn mã lệnh sau

```
Console.WriteLine("Wellcome to C#.NET");  
Console.ReadLine();
```

- Bước 5: Bấm phím F5 hoặc  để chạy chương trình



# Biên dịch chương trình C#





# Tóm tắt

- ❖ C# là một ngôn ngữ lập trình của .NET
  - Là ngôn ngữ hiện đại, hướng đối tượng, an toàn kiểu, mạnh mẽ, bền bỉ, kết hợp nhiều ưu điểm và sẽ trở nên phổ biến
- ❖ Có ba loại chương trình C#
  - Console, Windows Form, Web Form



# Câu hỏi

1. Hai thành phần chính của .NET Framework là gì?
2. CLR (Common Language Runtime) là gì?
3. Một chương trình C# có thể chạy trên bất cứ máy nào?
4. Nêu các lý do tại sao ngôn ngữ C# là một ngôn ngữ lập trình tốt?
5. Tập tin chương trình mã nguồn C# có phần mở rộng gì?
6. Soạn thảo và biên dịch, thực thi chương trình xuất ra màn hình một bài thơ nào đó



# YOUTUBE

1. <https://www.youtube.com/watch?v=79UWvR734wI>
2. <https://www.youtube.com/watch?v=tIjgasYPaj0>