

Bài giảng môn học

Bài 2: Tổng quan về .NET framework và môi trường lập trình Visual Studio

Lương Trần Hy Hiến

FIT, HCMUP

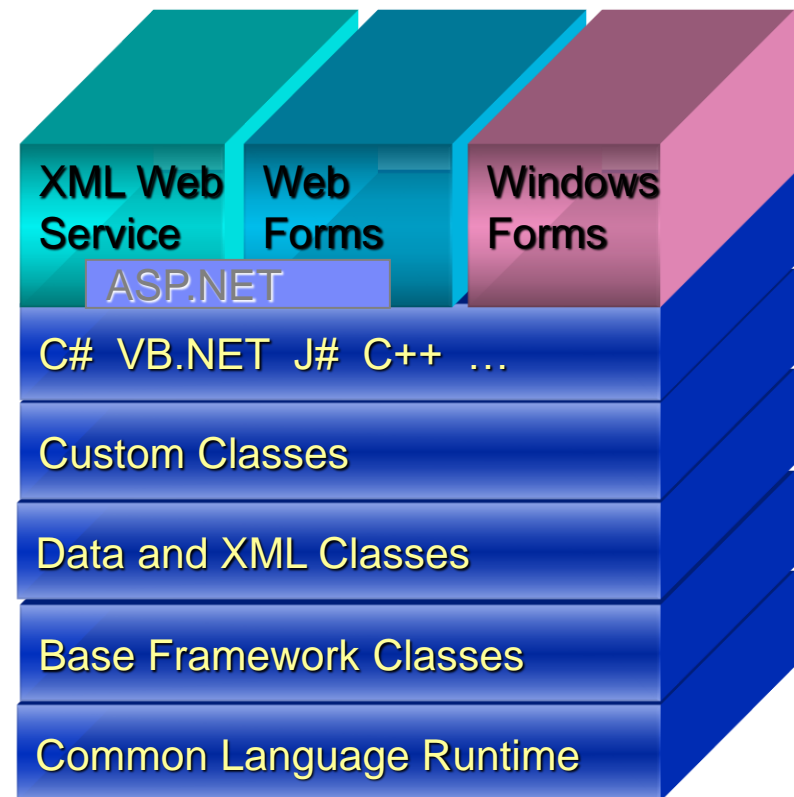
Lập trình Windows Form với C#

Nội dung

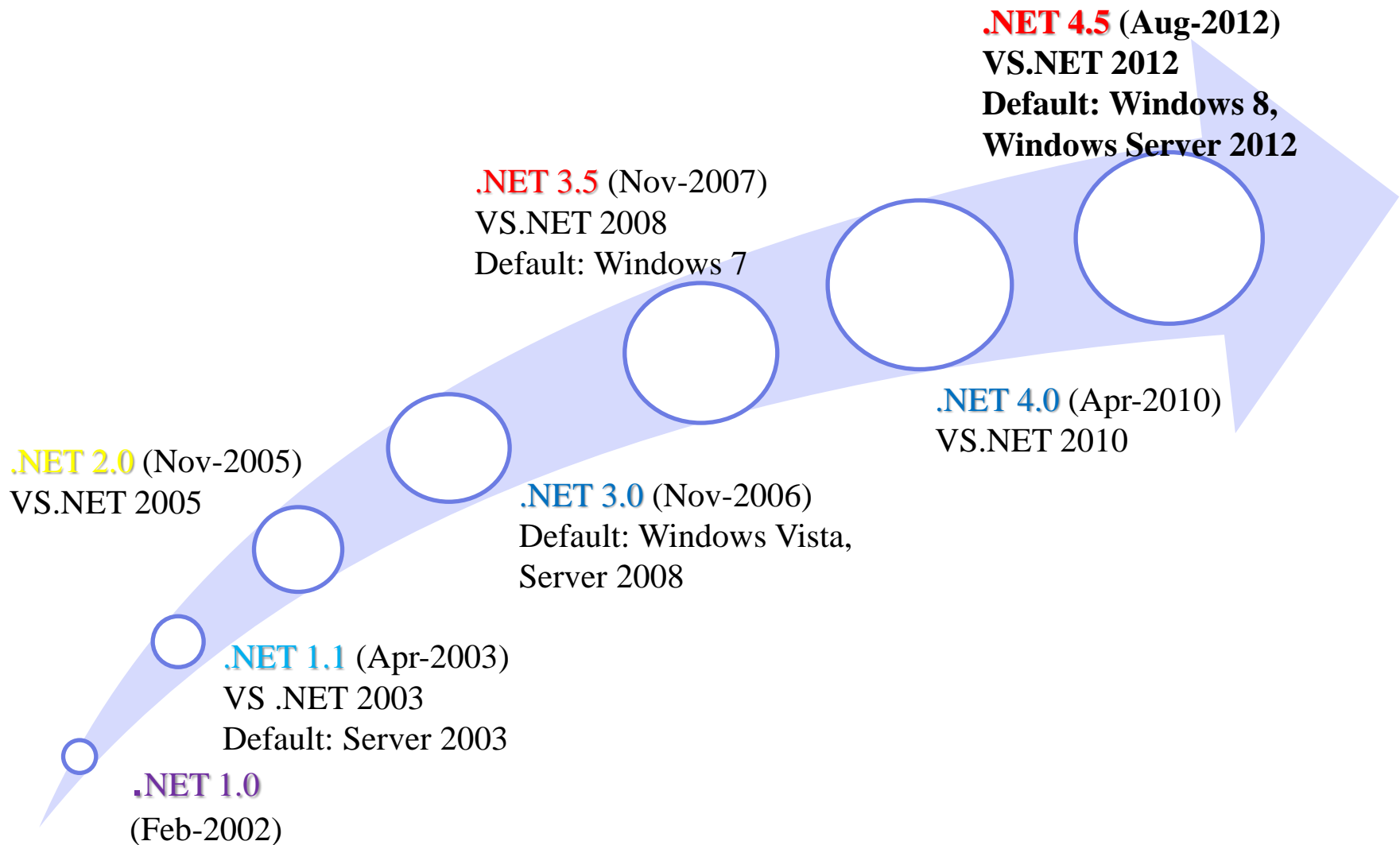
1. **Giới thiệu .NET Framework**
2. **Tổng quan ngôn ngữ C#**
3. **Môi trường lập trình Visual Studio**

.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



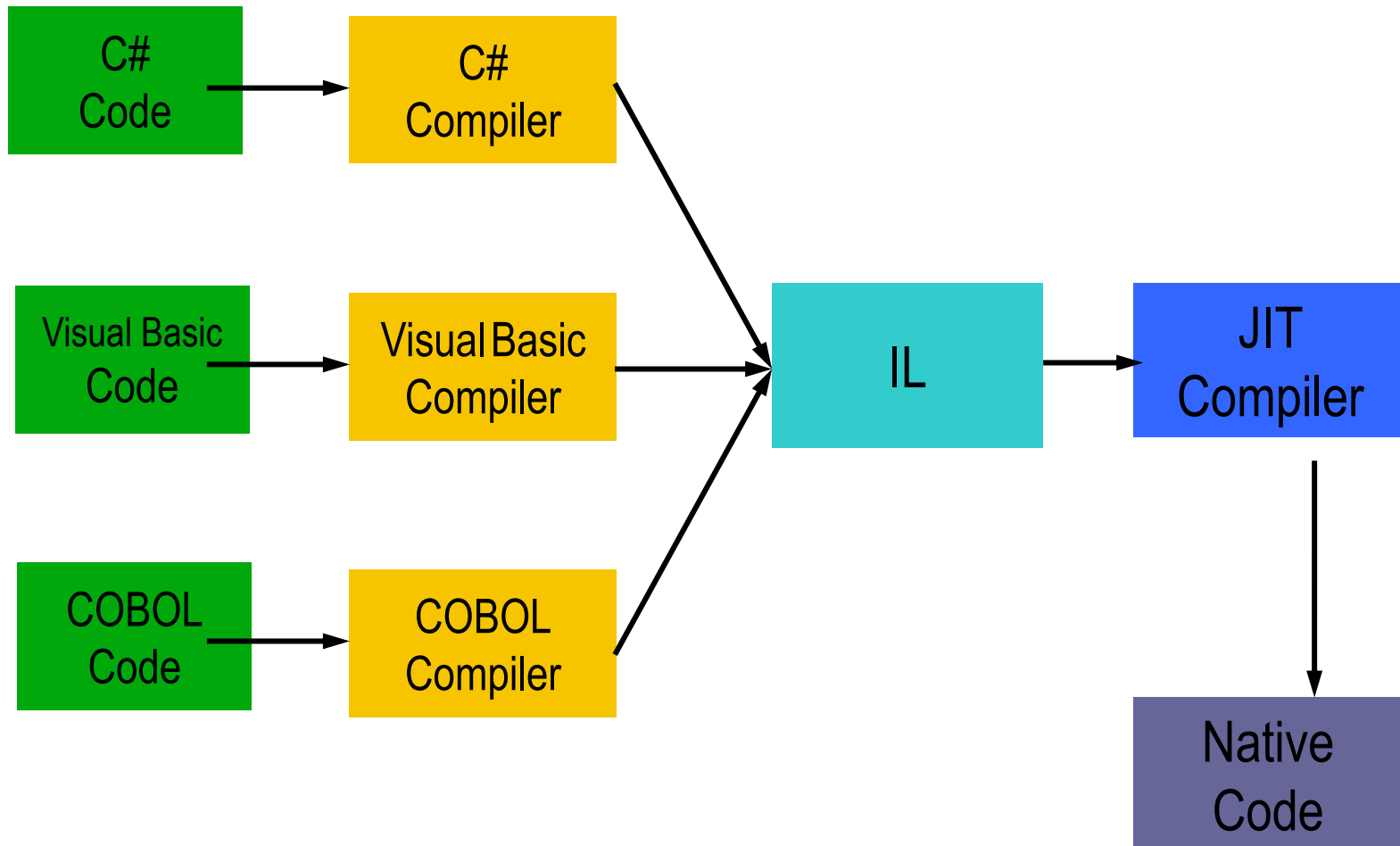
.NET Framework



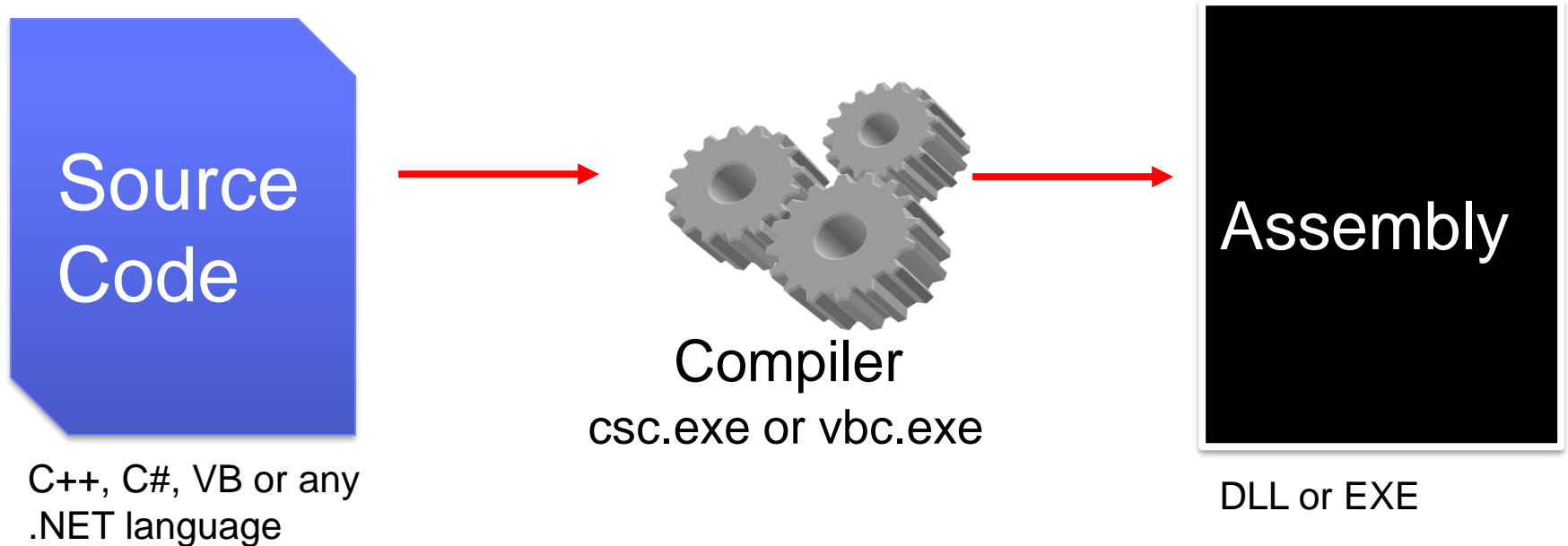
.NET Framework

- **Bộ khung phát triển ứng dụng;**
 - Bốn ngôn ngữ chính: C#, VB.NET, C++.NET, Jscript.NET
 - Common Language Runtime – CLR (.NET Runtime): tương tự máy ảo Java
 - Bộ thư viện Framework Class Library - FCL

Thực thi một chương trình .Net



Common Language Runtime - compilation



Common Language Runtime (CLR)

- Runtime là một agent quản lý mã nguồn khi nó được thực thi
 - Runtime cung cấp các dịch vụ : quản lý bộ nhớ , quản lý tiểu trình , quản lý từ xa
 - Runtime đảm bảo việc thực hiện được bảo mật và mạnh mẽ
 - Mã nguồn mà đích tới là runtime (managed code) , ngược lại (unmanaged code)
- CLR cung cấp môi trường thực thi chung cho các ngôn ngữ trên nền .NET**

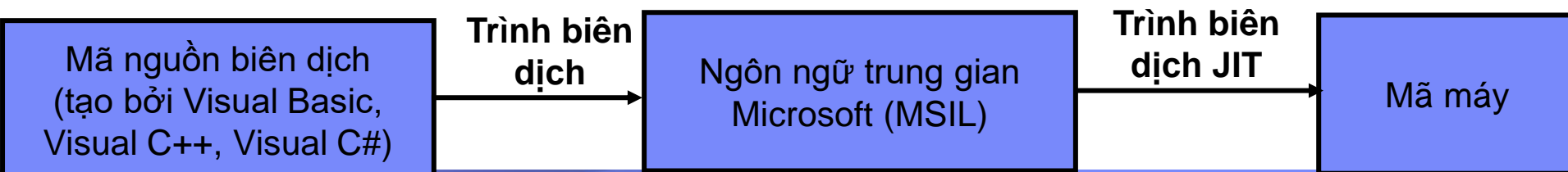
Common Language Runtime (CLR)

■ Bằng cách nào?

- 1. Khi biên dịch một ứng dụng .Net
- 2. Môi trường phát triển dịch mã nguồn thành mã tự quản (managed code) dưới dạng ngôn ngữ trung gian MSIL (hay IL) (Microsoft Intermediate Language)
- 3. Trình biên dịch phát sinh siêu dữ kiện megadata (thành viên đối tượng , kiểu , tham chiếu ,.....)
- MSIL và megadata → EXE file

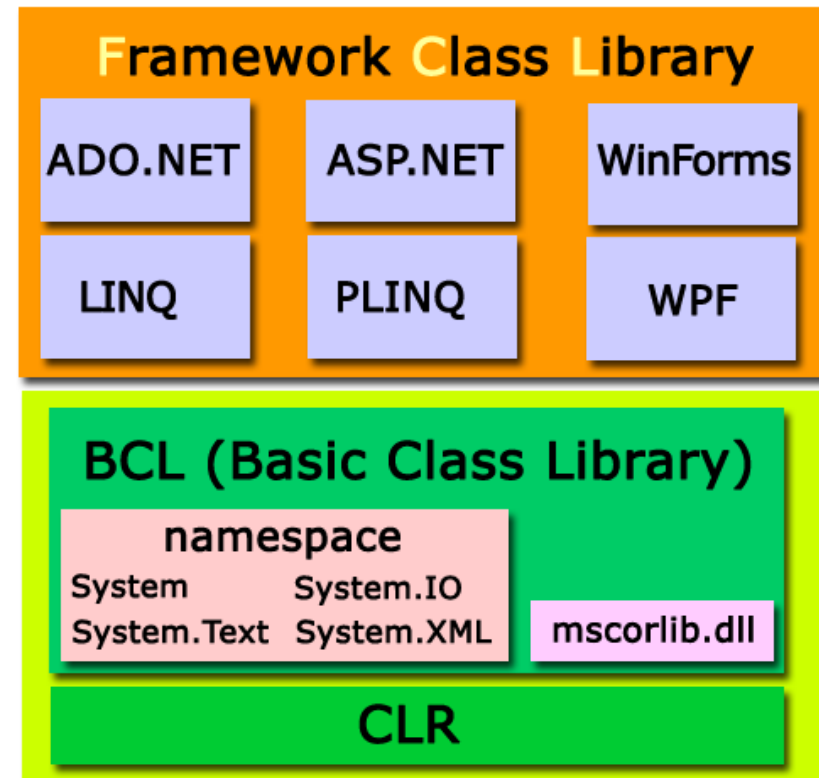
Common Language Runtime (CLR)

- **Khi thực thi mã ứng dụng (EXE file)**
 - 1. Tùy thuộc vào chỉ thị người dùng
 - 2. Trình biên dịch tức thời JIT (Just In Time) của CLR cung cấp sẽ :
 - Chuyển đổi mã quản lý (managed code) thành mã máy (native code)
 - JIT cất giữ mã máy phát sinh vào bộ nhớ đệm cho các thao tác tương tự



Framework Class Library - FCL

- Là một phần trong bộ khung .Net
- Đóng gói những lớp Hướng Đối Tượng được cung cấp
- Giúp bạn dễ dàng tạo các ứng dụng trên nền Window, trên nền Web



Framework Class Library - FCL

- Thư viện lớp của .Net Framework được tổ chức theo cấu trúc cây dựa trên khái niệm namespace (cấu trúc cây)
- Có thể có nhiều lớp trùng tên nhau nhưng buộc phải nằm ở các namespace khác nhau
- Namespace : Microsoft.Csharp, Microsoft.VisualBasic, Microsoft.Win32, System
- Class: System.Array, System.Console, System.String, System.Math v.v...

Framework (Base) Class Library

- Thư viện nền tảng cho .NET Framework
- Các namespace cơ bản của FCL/BCL

Namespace	Description
System	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

FCL
BCL

Các loại ứng dụng C#

☐ **Chương trình Console (TUI)**

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

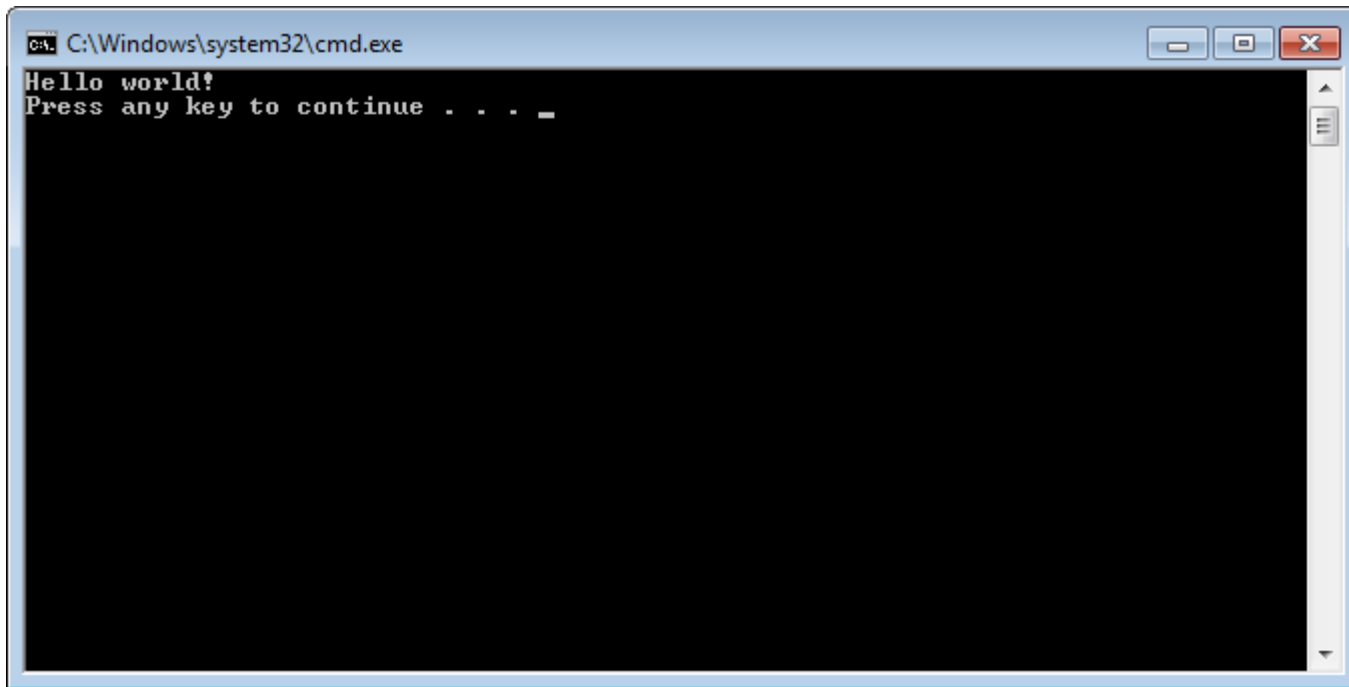
☐ **Chương trình Windows Form**

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

☐ **Chương trình Web Form**

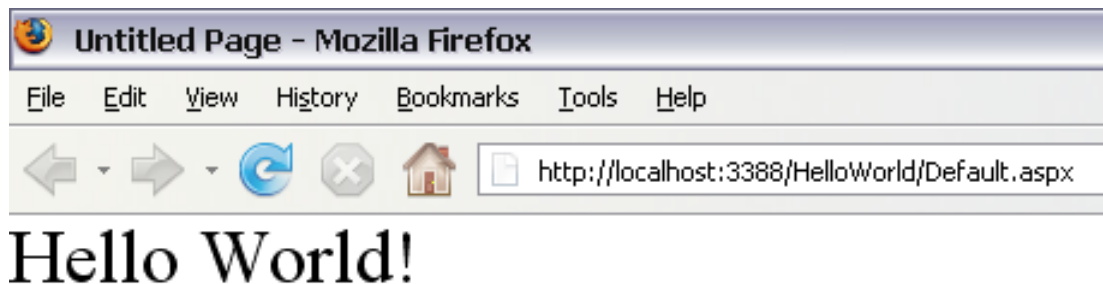
- ☐ Kết hợp với ASP.NET, C# đóng vai trò xử lý bên dưới (underlying code)
- ☐ Có giao diện đồ họa và xử lý sự kiện

Ứng dụng Console

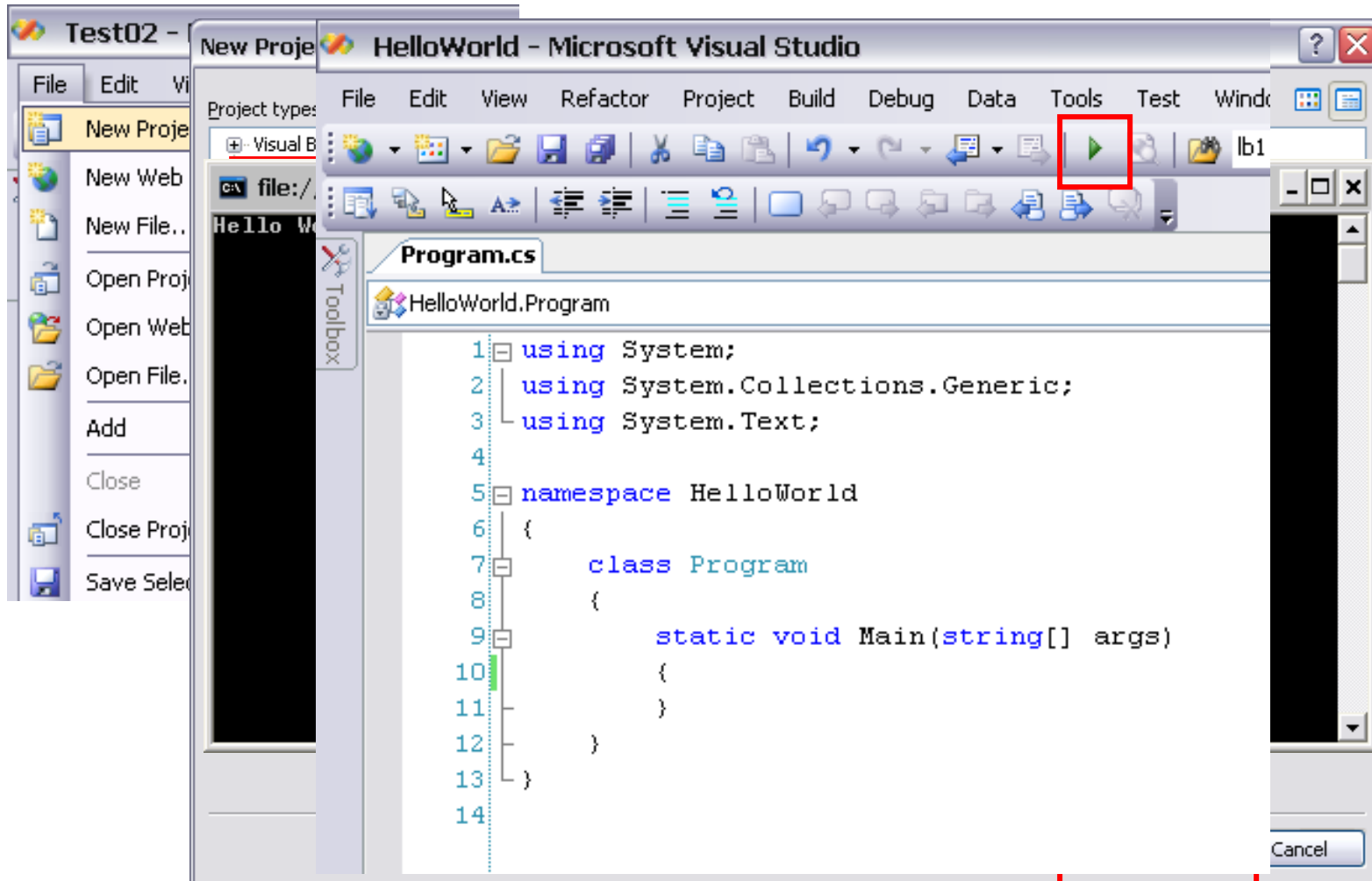


A screenshot of a Windows Command Prompt window. The title bar shows the path `C:\Windows\system32\cmd.exe`. The window contains the text `Hello world!` and `Press any key to continue . . . _` on two lines. The background is black, and the text is white. The window has standard Windows XP-style window controls (minimize, maximize, close) in the top right corner.

Ứng dụng WinForm và Web Form



Tạo ứng dụng Console



Ứng dụng C# đầu tiên

```
// Chương trình C# đầu tiên

using System;
using System.Collections.Generic;
using System.Text;

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

Cấu trúc chương trình C#

- Phần chú thích (option)

```
// Chương trình C# đầu tiên
```

- Phần khai báo dùng namespace (option)

```
using System;  
using System.Collections.Generic;  
using System.Text;
```

- Phần định nghĩa namespace và lớp

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

Câu lệnh

- 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 (;)

Phương thức Main

Các câu lệnh

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

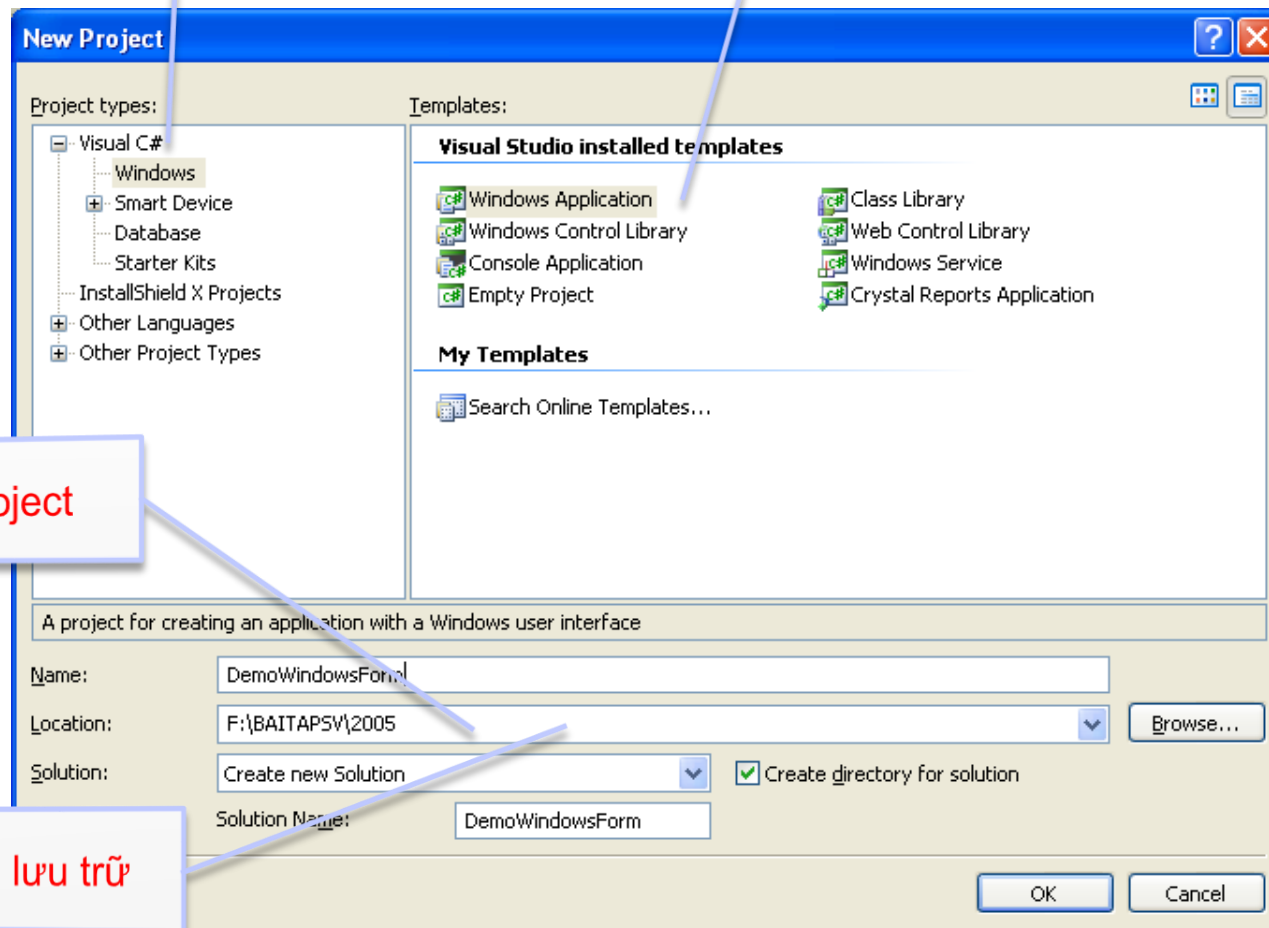
Tạo ứng dụng mẫu

1. Chọn loại ngôn ngữ

2. Chọn loại ứng dụng

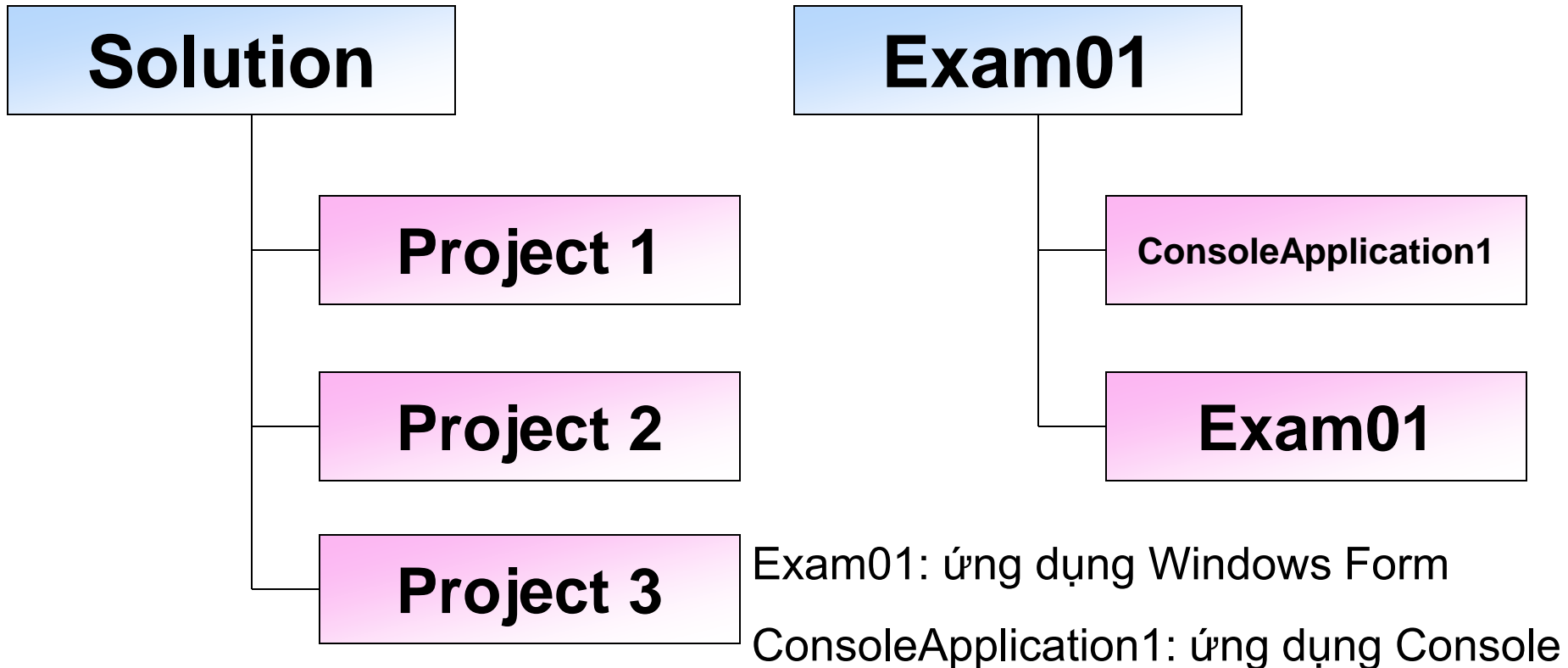
3. Đặt tên Project

4. Chọn nơi lưu trữ



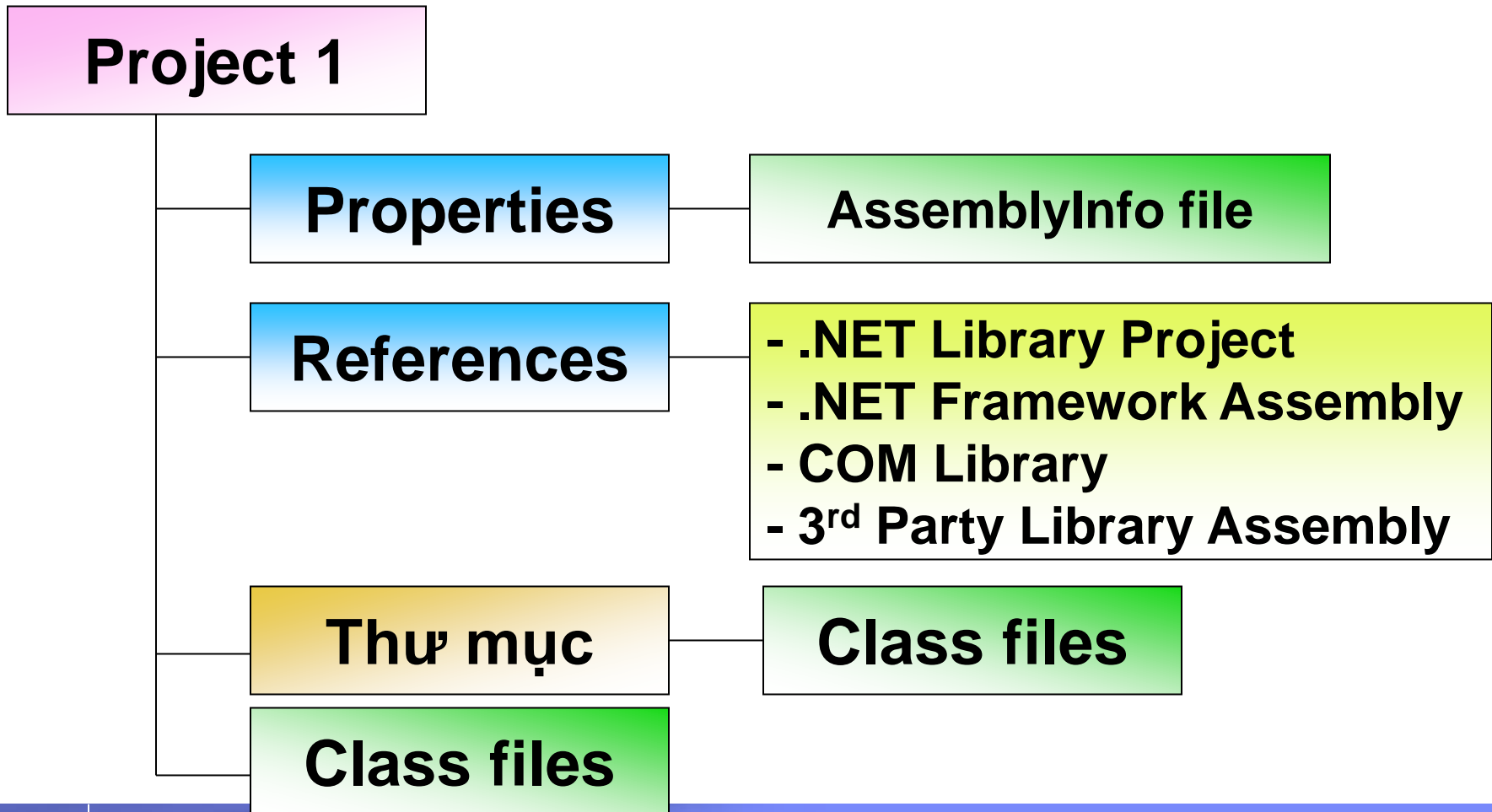
Tạo ứng dụng mẫu

❖ Cấu trúc 1 ứng dụng (Solution)



Tạo ứng dụng mẫu

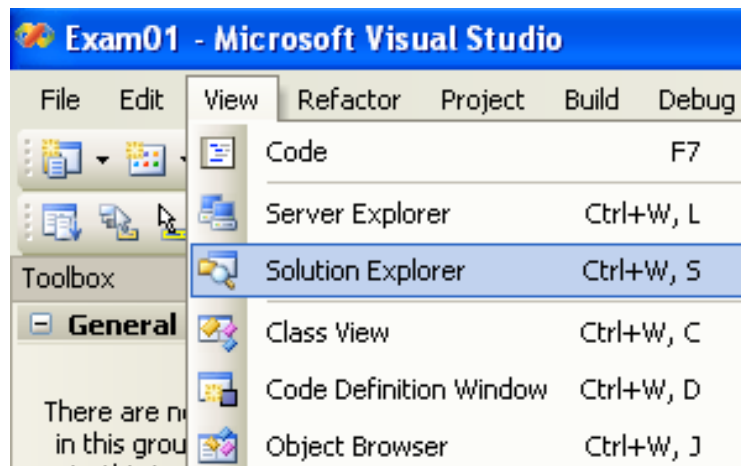
❖ Cấu trúc 1 Project



Tạo ứng dụng mẫu

- **Solution View: Xem cấu trúc ứng dụng**

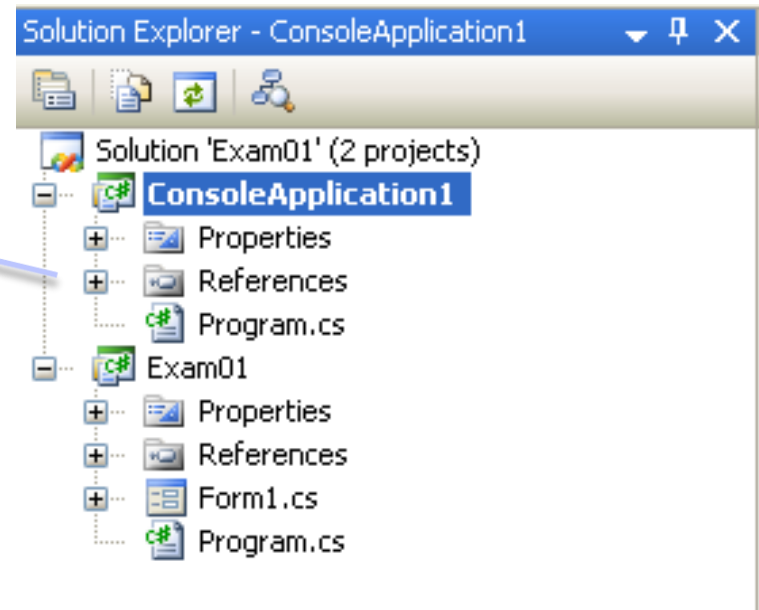
Click vào đây



Tạo ứng dụng mẫu

■ Solution view

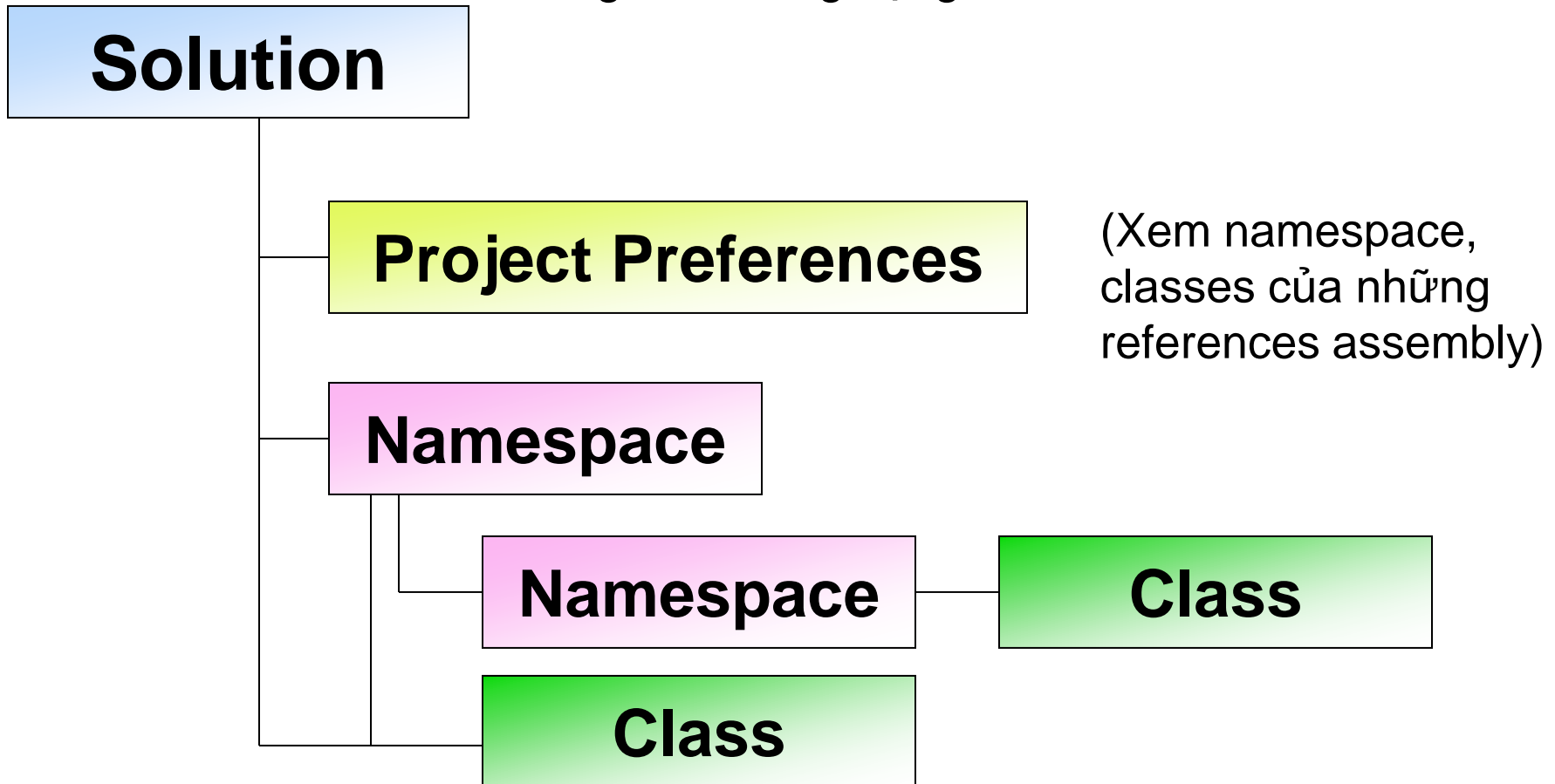
Cửa sổ Solution View



Tạo ứng dụng mẫu

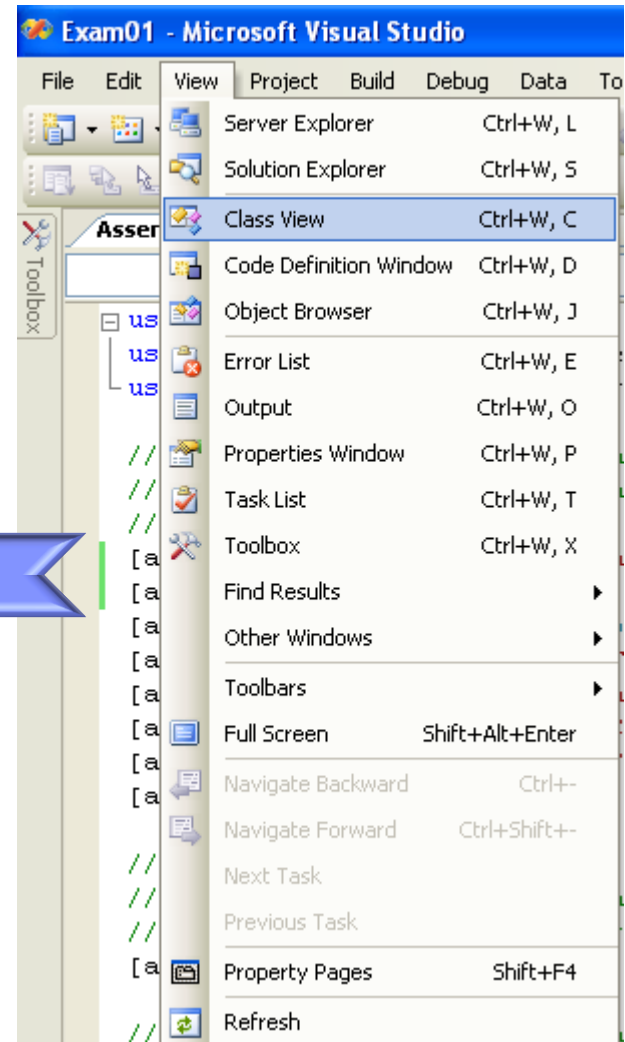
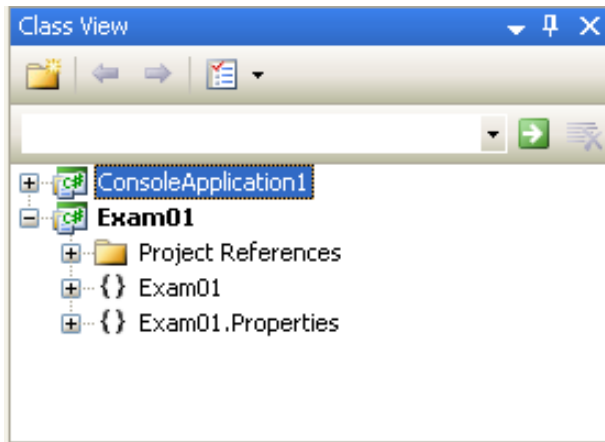
❖ Class view :

Xem cách tổ chức bên trong của 1 ứng dụng

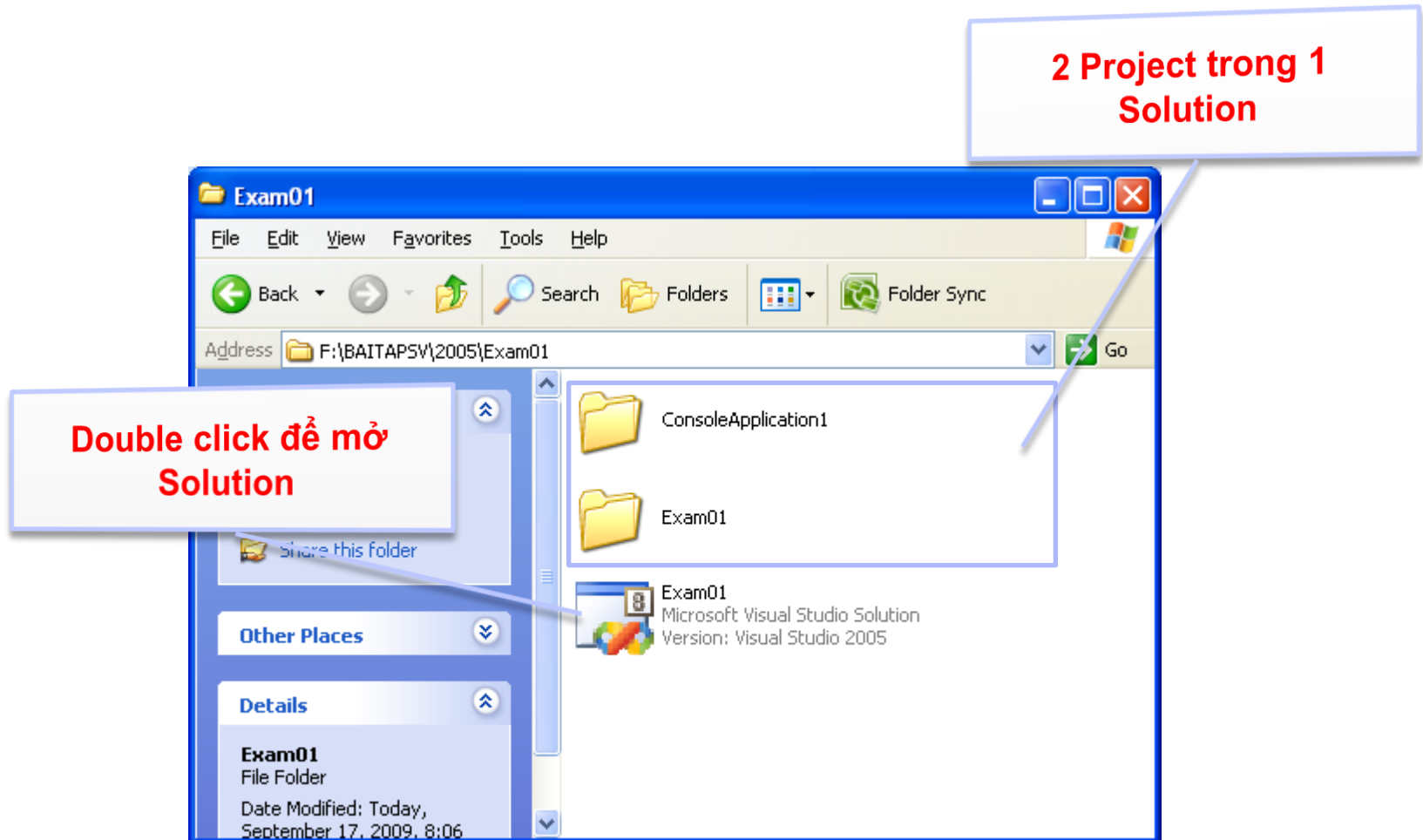


Tạo ứng dụng mẫu

- **Class View:**
 - View → Class View



Tạo ứng dụng mẫu



Thay đổi thông tin ứng dụng

- **Mở file AssemblyInfo.cs**
- **Thay đổi các thông tin cơ bản**
 - Assembly title
 - AssemblyDescription
 - AssemblyCompany
 - AssemblyProduct
 - AssemblyCopyright

Thay đổi thông tin ứng dụng

Exam01 - Microsoft Visual Studio

File Edit View Project Build Debug Data Tools Window Community Help

Debug Any CPU

AssemblyInfo.cs* Program.cs

2. Hiệu chỉnh các thông tin cơ bản cho ứng dụng

```
using System.Reflection;
using System.Runtime.InteropServices;
using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the
// set of attributes. Change these attribute values to modify the
// associated with an assembly.

[assembly: AssemblyTitle("Tieu de")]
[assembly: AssemblyDescription("Day la chương trình Demo")]
[assembly: AssemblyConfiguration("Cau hinh")]
[assembly: AssemblyCompany("FIT, HCMUP")]
[assembly: AssemblyProduct("ConsoleApplication1")]
[assembly: AssemblyCopyright("Copyright © FIT, HCMUP 2009")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not
// to COM components. If you need to access a type in this assembly
// from COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]
```

1. Mở file AssemblyInfo.cs

Solution Explorer - Solution Explorer

Solution 'Exam01' (2 projects)

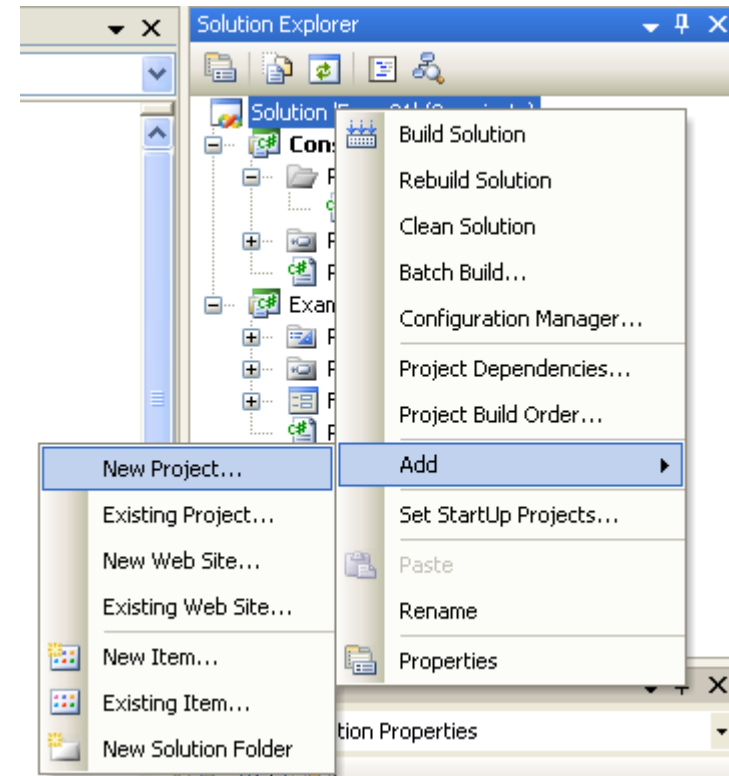
- ConsoleApplication1
 - Properties
 - AssemblyInfo.cs
 - References
 - Program.cs
- Exam01
 - Properties
 - References
 - Form1.cs
 - Program.cs

Properties

Ready

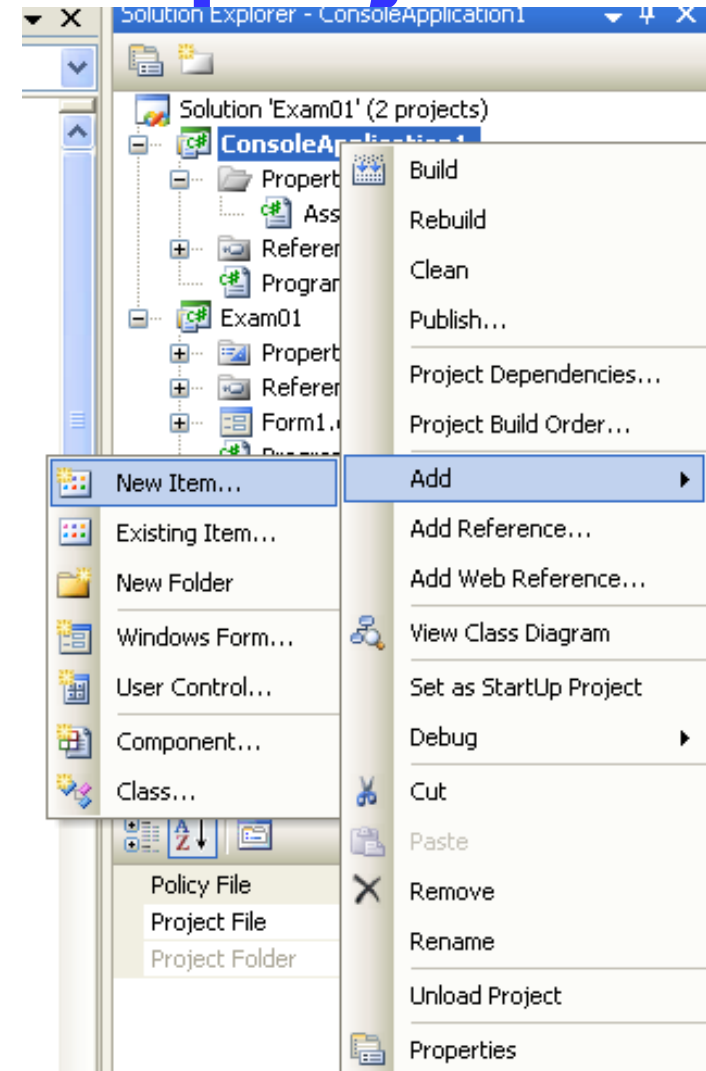
Thêm 1 project vào trong solution

- Mở view Solution
- Nhấn chuột phải lên tên solution
- Chọn **Add \ New Project**



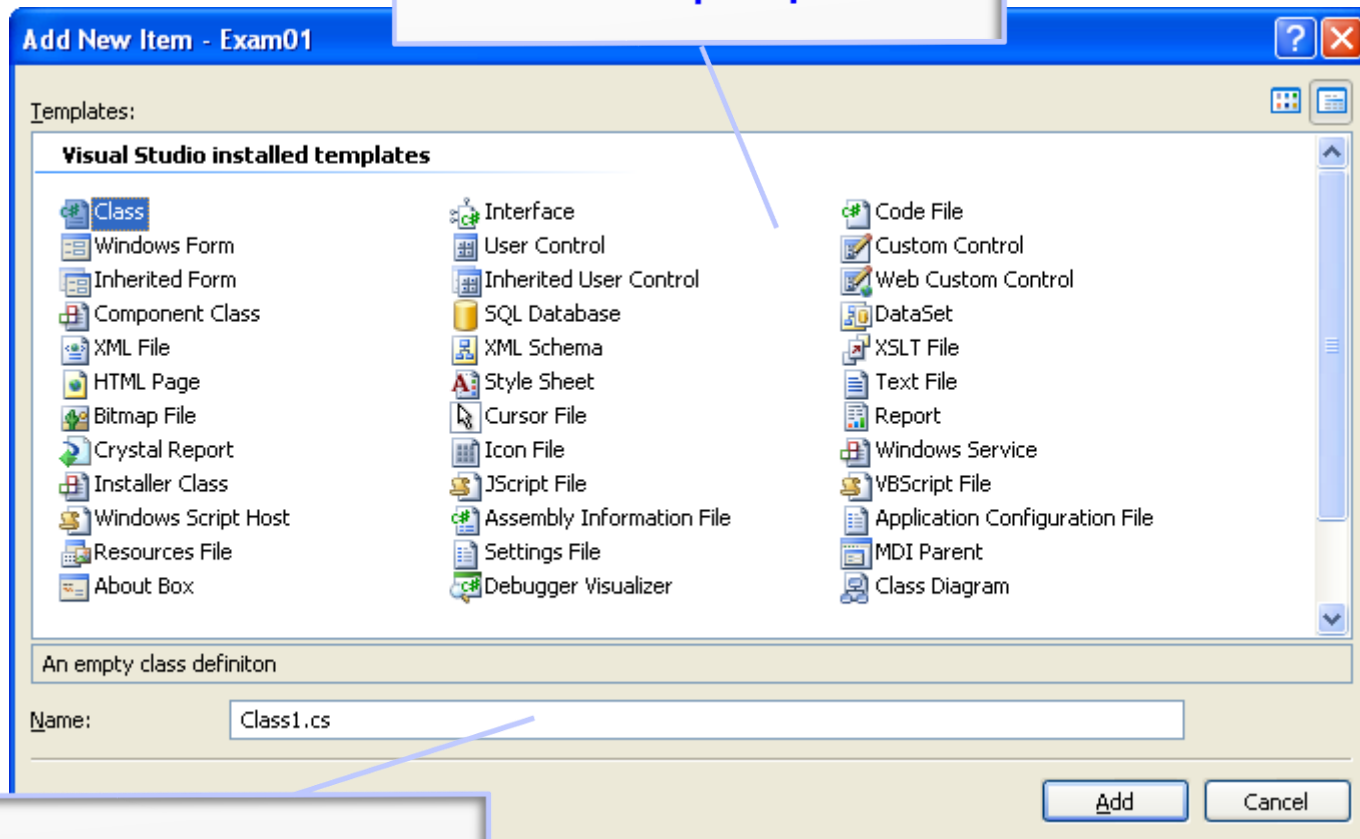
Thêm 1 item bất kì vào project

- Mở view solution
- Nhấn chuột phải project cần thêm 1 item
- Chọn **Add \ New Item**



Thêm 1 item bất kì vào project

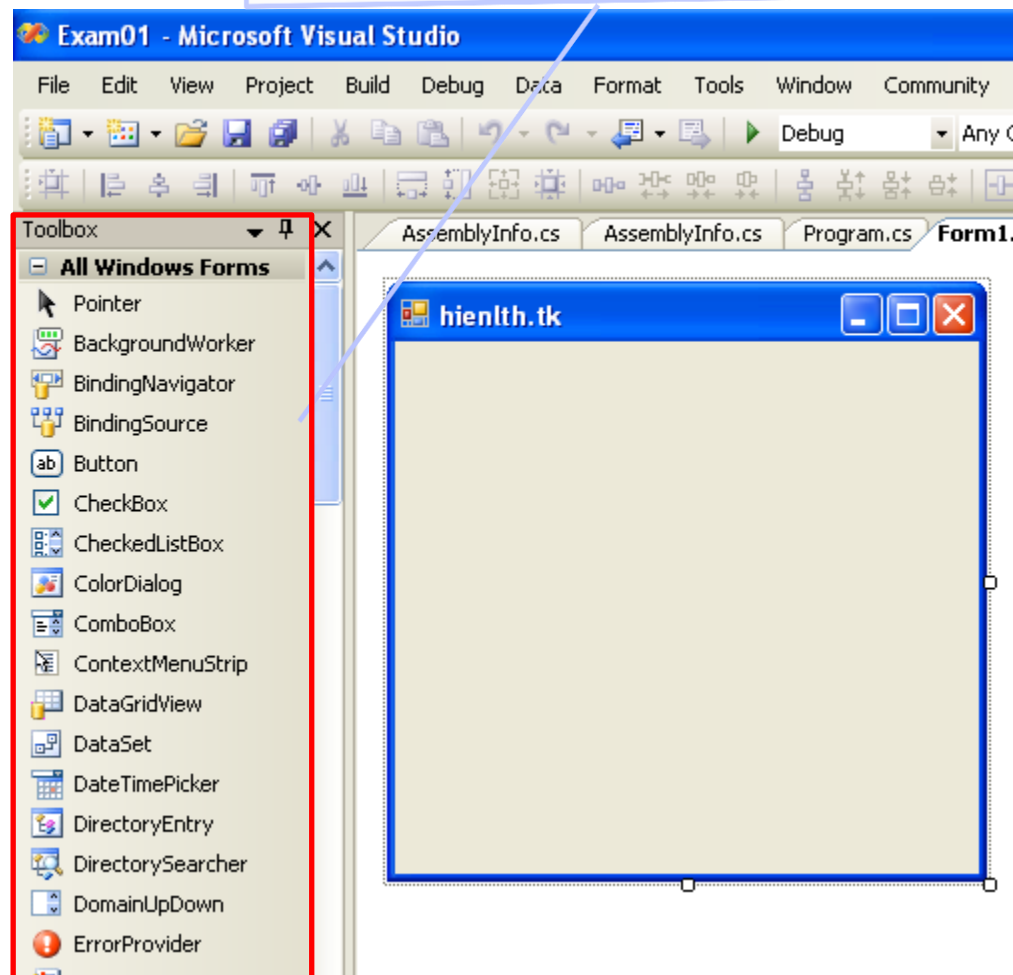
Bước 1: Chọn loại Item



Bước 2: Đặt tên file

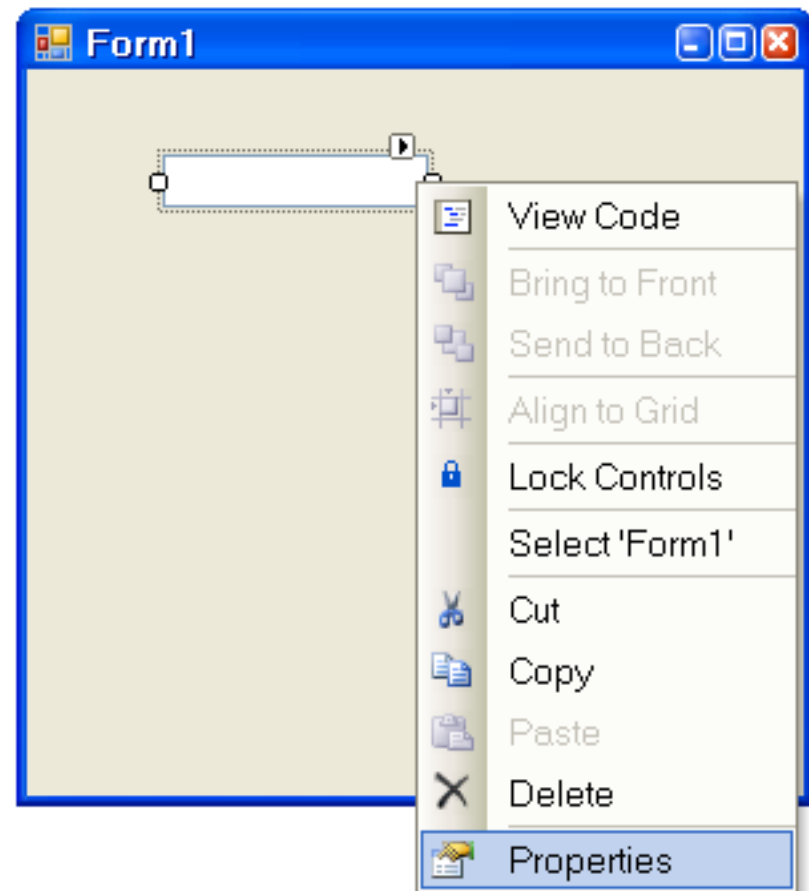
Window Application Project

Drop & Drap các Control lên Form để thiết kế giao diện



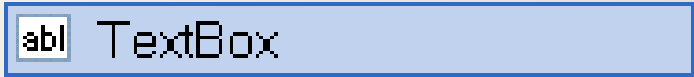

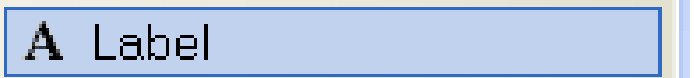
Window Application Project

- Để xem thuộc tính của của control chọn (Mở view Properties bằng cách Menu **View \ Properties Window**)
- Nhấn chuột phải lên control (như Textbox, button, ..) và chọn **Properties**, khi đó màn hình Properties của control này sẽ hiện ra.



Window Application Project

- Mỗi control sẽ được xác định bằng thuộc tính **Name**. Thuộc tính này dùng để xác định control khi lập trình
- Lưu ý: Cần đặt tên Name sao cho gợi nhớ

Control	Tiếp đầu ngữ	Biểu tượng
Textbox	txt Ví dụ: txtHoTen, txtTuoi	
Button	btn Ví dụ: btnOK, btnCancel	
Label	lbl Ví dụ: lblThongBao	

Window Application Project

Các thuộc tính chung:

- Name: Định danh cho control
- Text: Xuất dạng text thông tin control cho người dùng

Sự kiện:

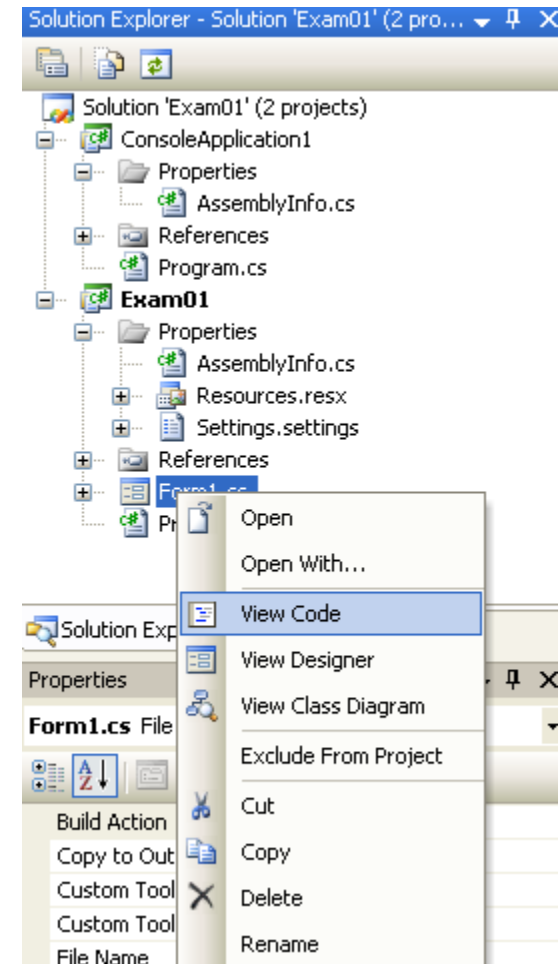
Button có sự kiện Click, phát sinh khi người dùng click vào button

Window Application Project

- **Một window form có 2 view:**
 - **Design View:** Dùng để thiết kế giao diện
 - **Code View:** Dùng để lập trình trên form (Xác định các control trên design view thông qua thuộc tính Name của từng control)

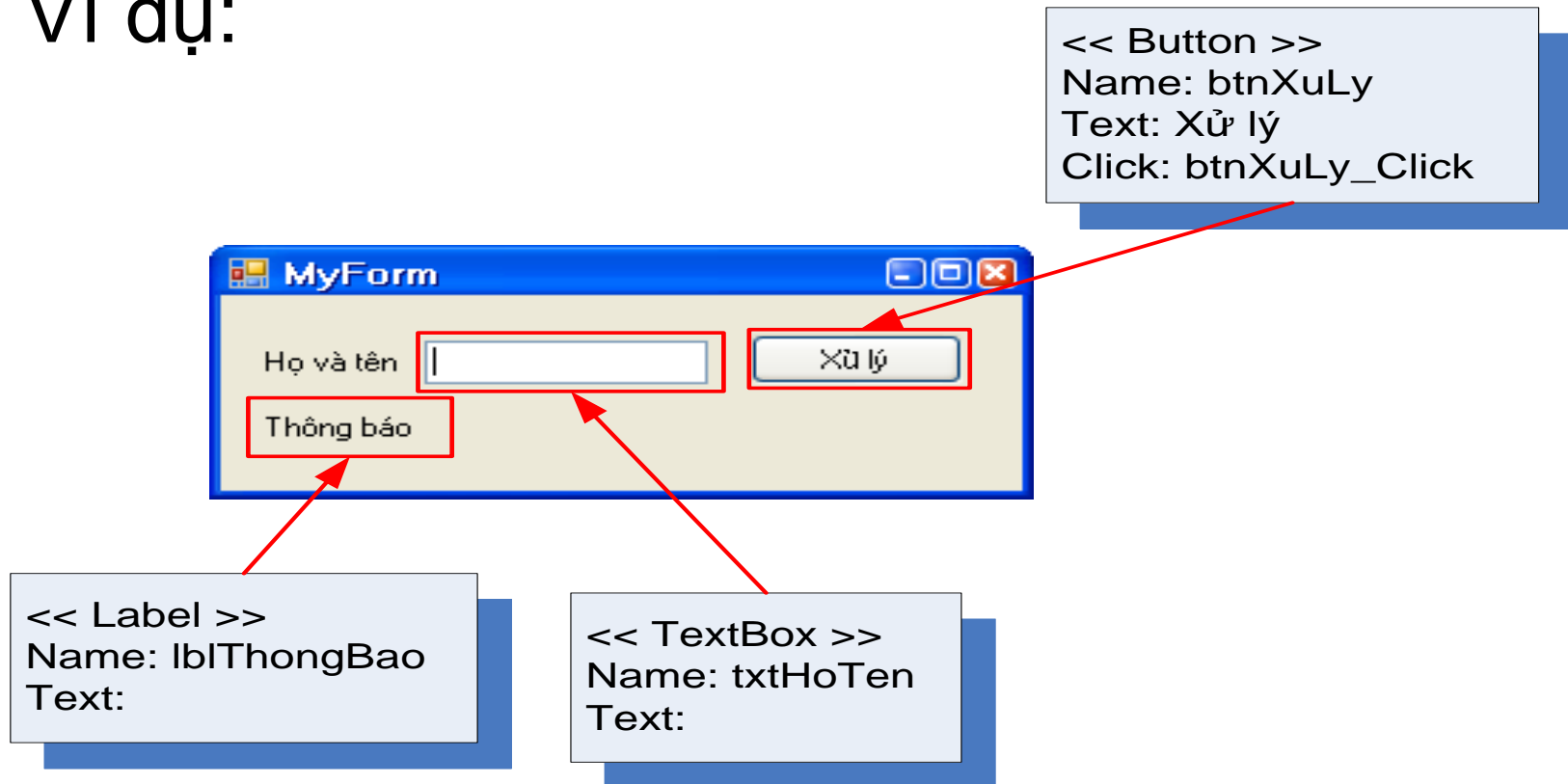
Window Application Project

- Để xem view của Form: Mở view solution, nhấn chuột phải lên Form, chọn **View Code** hoặc **View Designer**



Window Application Project

Ví dụ:



Window Application Project

■ **Bắt sự kiện Click của nút button XuLy**

- Nhấn chuột phải vào nút button
- Chọn Properties
- Nhấn vào nút biểu tượng sấm sét
- Chọn dòng sự kiện Click
- Double click vào dòng này

(Khi đó VS sẽ phát sinh hàm bắt sự kiện Click của nút button)

Window Application Project

■ Hàm bắt sự kiện Click của button XuLy

```
namespace WinApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnXuLy_Click(object sender, EventArgs e)
        {
            lblThongBao.Text = "Tên của bạn là: " + txtHoTen.Text;
        }
    }
}
```

Window Application Project

