



# **LẬP TRÌNH .NET 1 + BTL**

**(WINDOWS FORMS PROGRAMMING IN .NET)**

**Gv: Đặng Hữu Nghị**

**Sđt: 0989640319**

**Email: [nghidanghuu@gmail.com](mailto:nghidanghuu@gmail.com)**

# NỘI DUNG

Chương 1: Giới thiệu về .Net Framework

Chương 2: Giao diện người dùng (User Interface)

Chương 3: Các kiểu dữ liệu & Bẫy lỗi trong C#

Chương 4: Lập Trình Hướng Đối Tượng với C#

Chương 5: Cơ sở dữ liệu & C#

## TÀI LIỆU THAM KHẢO

- Phương Lan, Lập trình Windows với C#.net, Lao động - Xã hội, 2002
- Phương Lan, Phạm Hữu Khang, Hoàng Đức Hải, Kỹ Thuật Lập trình ứng dụng C#.net toàn tập, Lao động xã hội, 2005
- Đinh Xuân Lâm, Những bài thực hành Visual Basic.net căn bản, Đại học Mở - Địa Chất, 2004
- Phạm Hữu Khang, Ví dụ & bài tập Visual Basic.NET.Lập trình hướng đối tượng

# CHƯƠNG 1 GIỚI THIỆU VỀ .NET FRAMEWORK

1.1. Giới thiệu về .NET

1.2. Ngôn ngữ C#

## 1.1. GIỚI THIỆU VỀ .NET

- Framework là một tập hợp các thư viện để hỗ trợ cho người lập trình.
- Mỗi Framework được tạo ra có một kiến trúc khác nhau  $\Rightarrow$  Lập trình viên phải tuân theo kiến trúc đó
- .NET Framework là thư viện tài nguyên của Microsoft, hỗ trợ cho các lập trình viên trong nhiều yêu cầu khác nhau.

# 1.1. GIỚI THIỆU VỀ .NET

## .NET Framework

Phiên bản	Số hiệu phiên bản	Ngày phát hành	Visual Studio	Được phát hành kèm theo
1.0	1.0.3705.0	13 tháng 2 năm 2002	Visual Studio.NET	Windows XP Tablet and Media Center Editions <sup>[1]</sup>
1.1	1.1.4322.573	24 tháng 4 năm 2003	Visual Studio.NET 2003	Windows Server 2003
2.0	2.0.50727.42	7 tháng 11 năm 2005	Visual Studio 2005	Windows Server 2003 R2
3.0	3.0.4506.30	6 tháng 11 năm 2006		Windows Vista, Windows Server 2008
3.5	3.5.21022.8	19 tháng 11 năm 2007	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0	4.0.30319.1	12 tháng 4 năm 2010	Visual Studio 2010	
4.5	4.5.50709	15 tháng 8 năm 2012	Visual Studio 2012	Windows 8, Windows Server 2012
4.6		20 tháng 7 năm 2015	Visual Studio 2015	Windows 10

## 1.1. GIỚI THIỆU VỀ .NET

- .NET Framework là nền tảng phát triển hoàn hảo của Microsoft, cung cấp cho lập trình viên các thư viện dùng chung hỗ trợ cho việc phát triển các kiểu ứng dụng khác nhau bao gồm:
  - Ứng dụng ASP.NET
  - Ứng dụng Windows Form
  - Web Services
  - Windows Services
  - Ứng dụng mạng và các ứng dụng điều khiển truy cập từ xa

## 1.1. GIỚI THIỆU VỀ .NET

### **Các thành phần của .NET Framework**

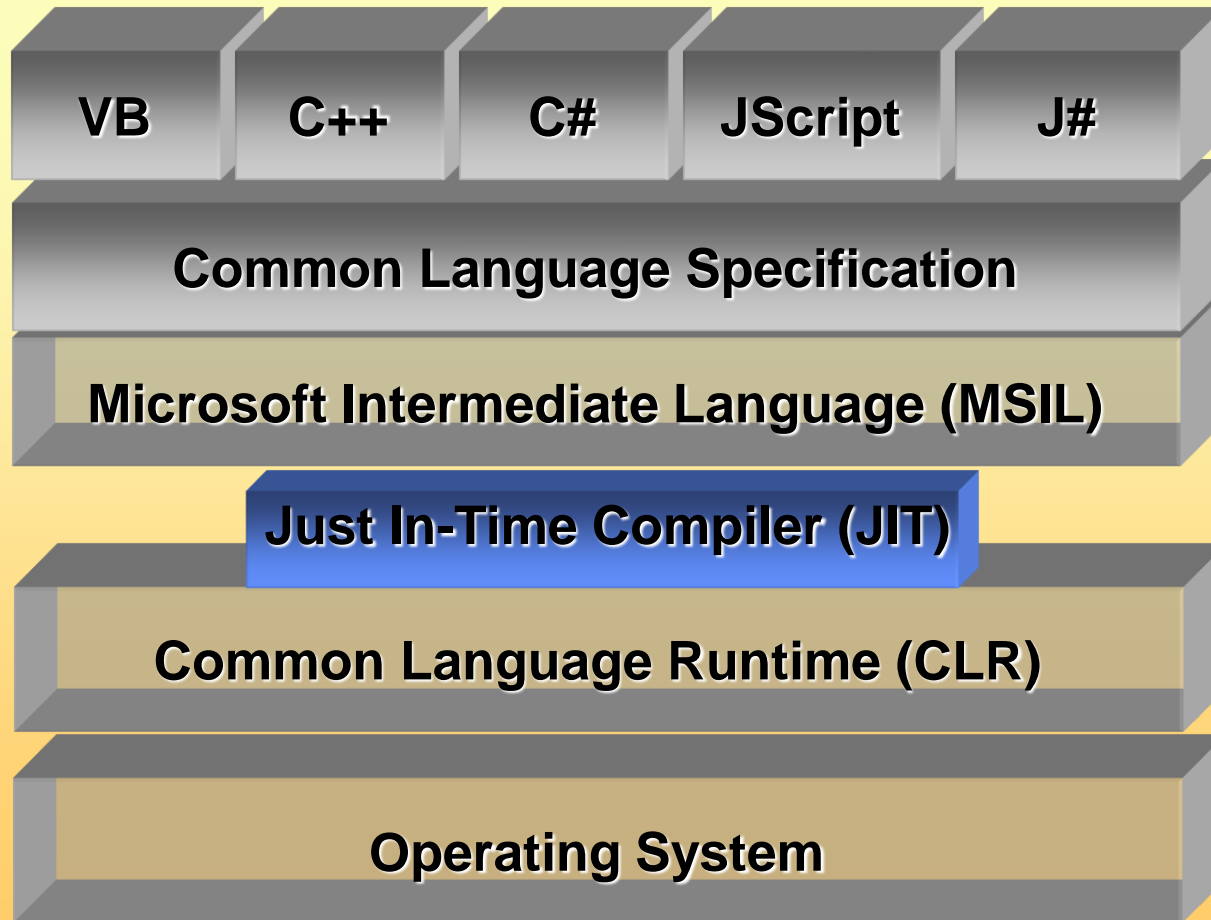
- .NET Framework gồm hai thành phần chính là:
  - Common Language Runtime (CLR)
  - và thư viện lớp (Framework Class Library – FCL).
- CLR là nền tảng của .NET Framework, giúp Microsoft có thể tích hợp nhiều ngôn ngữ lập trình khác nhau như VB.NET, C#, J#, ASP.NET,...vào bộ công cụ lập trình Visual Studio.NET
- CLR Giúp các ứng dụng viết trên các ngôn ngữ này có thể chạy được chung trên nền tảng hệ điều hành Windows



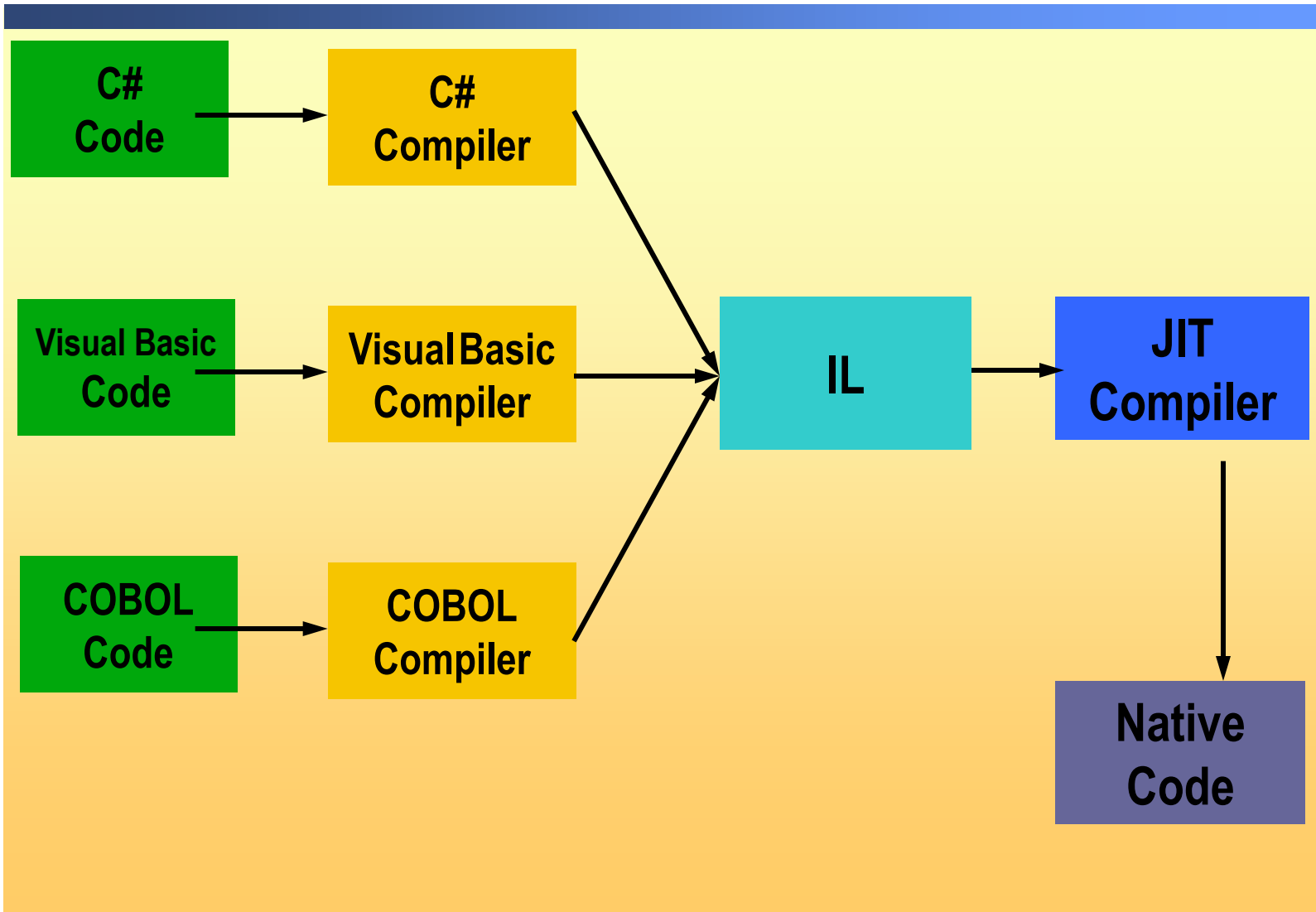
## 1.1. GIỚI THIỆU VỀ .NET

- Sở dĩ Microsoft có thể làm được điều này, bởi vì các ngôn ngữ lập trình đều được biên dịch ra một ngôn ngữ trung gian (Intermediate Language – IL) và sử dụng chung kiểu dữ liệu hệ thống (Common Type System).
- Sau đó CLR sử dụng một trình biên dịch gọi là Just-in-Time (JIT) Compiler chuyển các đoạn mã IL thành mã máy và thực thi.

# MSIL, JIT và CLR



# Thực thi một chương trình .Net



## 1.1. GIỚI THIỆU VỀ .NET

- Ngoài ra CLR còn làm các thành phần khác như:
  - *Garbage Collection (GC)*: Gọi là bộ phận thu gom rác; có chức năng tự động quản lí bộ nhớ.
  - *Code Access Security (CAS)*: Cung cấp quyền hạn cho các chương trình, tùy thuộc vào các thiết lập bảo mật của máy.
  - *Code Verification*: Bộ phận này đảm bảo cho việc chạy các đoạn mã là đúng đắn, và đảm bảo an toàn kiểu dữ liệu ngăn chặn các đoạn mã hoạt động vượt quyền như truy nhập vào các vùng nhớ không được phép

## 1.1. GIỚI THIỆU VỀ .NET

### ○ Thư viện lớp (Framework Class Library-FCL):

- Là một tập hợp lớn các lớp được viết bởi Microsoft, những lớp này được xây dựng một cách trực quan và dễ sử dụng;
- Cho phép lập trình viên thao tác rất nhiều các tác vụ sẵn có trong Windows.

## 1.1. GIỚI THIỆU VỀ .NET

### ➤ Base class library:

- ✓ Đây là thư viện các lớp cơ bản nhất, được dùng trong khi lập trình hay bản thân những người xây dựng .NET Framework cũng phải dùng nó để xây dựng các lớp cao hơn.
- ✓ Một số thư viện lớp base class library như: String, Integer, Exception, ...

## 1.1. GIỚI THIỆU VỀ .NET

### ➤ ADO.NET và XML:

- ✓ Bộ thư viện này gồm các lớp dùng để xử lý dữ liệu. ADO.NET thay thế ADO để trong việc thao tác với các dữ liệu thông thường.
- ✓ Các lớp đối tượng XML được cung cấp để xử lý các dữ liệu theo định dạng mới XML.
- ✓ Một số thư viện trong ADO.NET và XML như: `sqlDataAdapter`, `SqlCommand`, `DataSet`, ...

## 1.1. GIỚI THIỆU VỀ .NET

### ➤ Windows Forms:

- ✓ Bộ thư viện về lập trình Windows Forms gồm các lớp đối tượng dành cho việc xây dựng các ứng dụng Windows cơ bản.
- ✓ Một số thư viện thường dùng như: Form, UserControl, TextBox, Label, Button, ComboBox, ListBox, ListView, TreeView, ...



## 1.1. GIỚI THIỆU VỀ .NET

### ➤ Web Services:

- ✓ Là các dịch vụ được cung cấp qua Web (hay Internet).
- ✓ Dịch vụ được coi là Web Service không nhằm vào người dùng mà nhằm vào người xây dựng phần mềm.
- ✓ Web Services có thể dùng để cung cấp các dữ liệu hay một chức năng tính toán.

## 1.1. GIỚI THIỆU VỀ .NET

### ➤ ASP.NET:

- ✓ Ứng dụng Web xây dựng bằng ASP.NET tận dụng được toàn bộ khả năng của .NET Framework.
- ✓ ASP.Net cung cấp một bộ các Server Control để lập trình viên bắt sự kiện và xử lý dữ liệu của ứng dụng như đang làm việc với ứng dụng của Windows.
- ✓ Một số thư như: WebControl, HTML Control, ...

## 1.2. NGÔN NGỮ C#

- Ngôn ngữ lập trình được xây dựng dựa trên nền tảng những ngôn ngữ tương tự C (C, C++, Java) nhưng hoạt động trên .Net Framework.
- Hoạt động trên .NET Framework

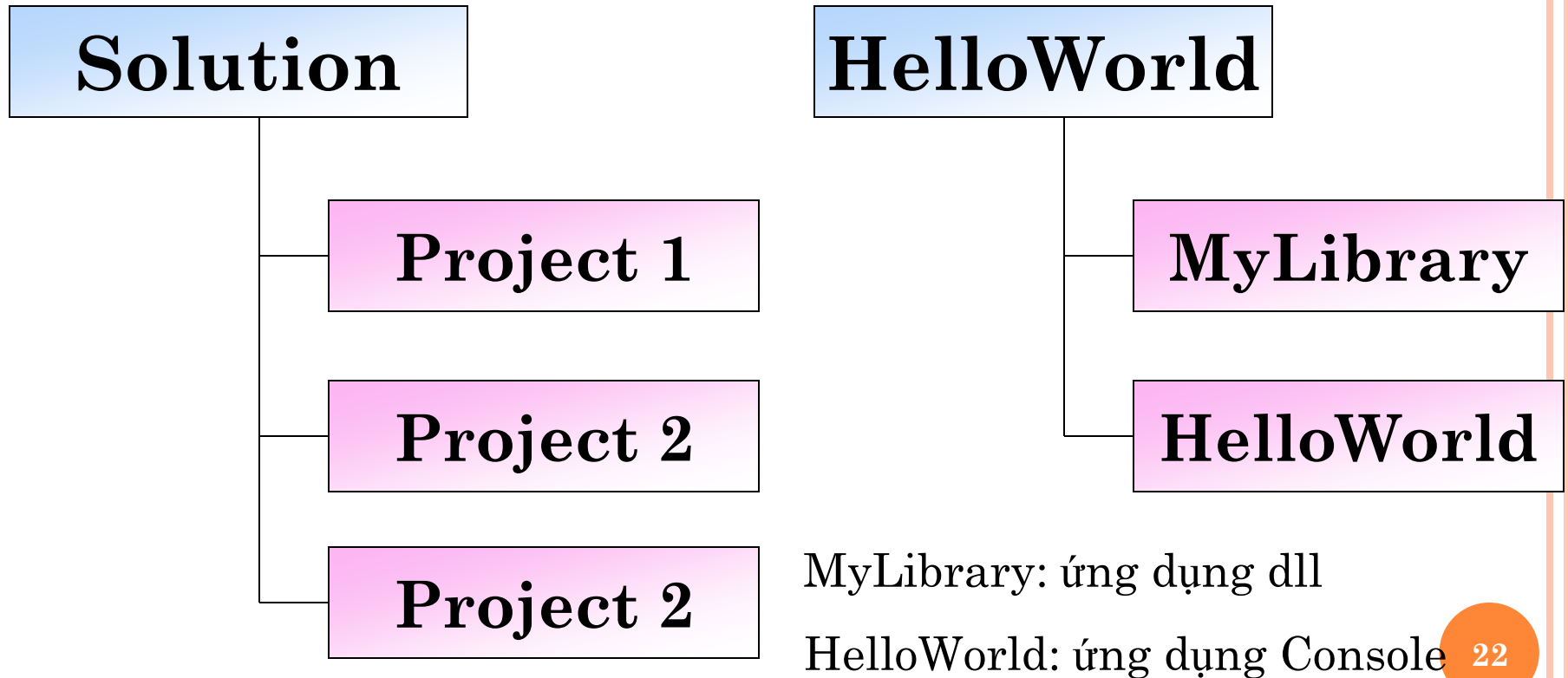
## 1.2. NGÔN NGỮ C#

- Dựa trên phương pháp thiết kế hướng đối tượng.
- Ứng dụng : Console, WinForm, WebForm.
- Có tính diễn đạt ngữ nghĩa cao.
- Phân biệt chữ hoa thường.

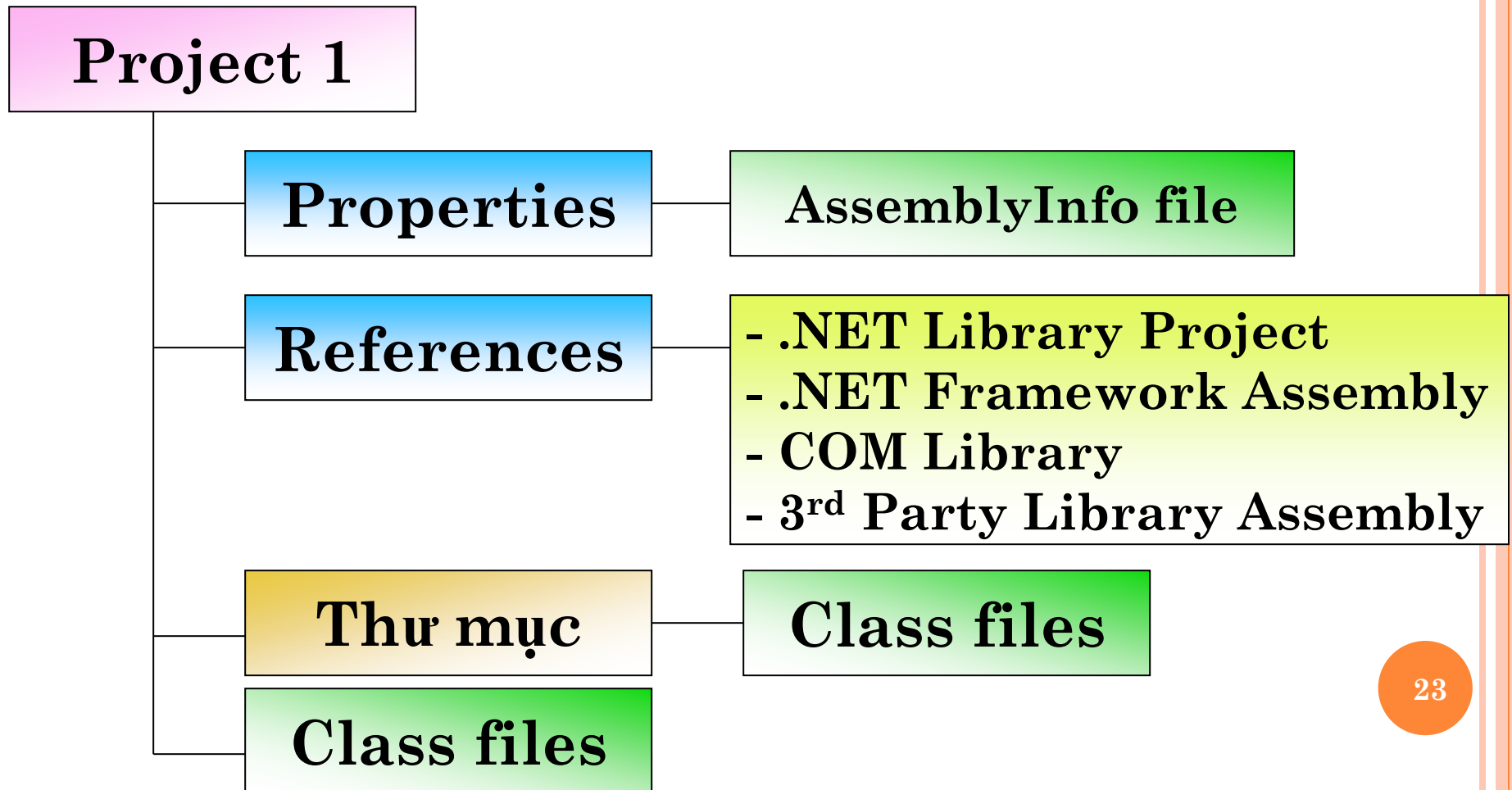
## 1.2.1. CẤU TRÚC 1 ỨNG DỤNG (SOLUTION)

- Visual Studio .NET coi một “bài toán” cần giải quyết là một solution.
- Một solution có thể bao gồm một hoặc nhiều project.
- Một Solution, nếu có nhiều Project thì nên được tạo ra trong một thư mục riêng để có thể chứa các project trong nó

## 1.2.1. CẤU TRÚC 1 ỨNG DỤNG (SOLUTION)



## 1.2.2. CẤU TRÚC 1 PROJECT



## 1.2.2. CẤU TRÚC 1 PROJECT

- Mỗi project mặc định đều chứa:
  - Các *file mã nguồn*: là các file văn bản có phần mở rộng .cs (viết tắt của C sharp);
  - Các *file cấu hình* của chương trình: là file xml có phần mở rộng .config;
  - Các *thư viện được tham chiếu* tới (References): là danh sách các file thư viện chuẩn của .NET framework, hoặc thư viện từ các hãng thứ ba, hoặc chính các project khác, chứa các class được sử dụng bởi các class trong project này.

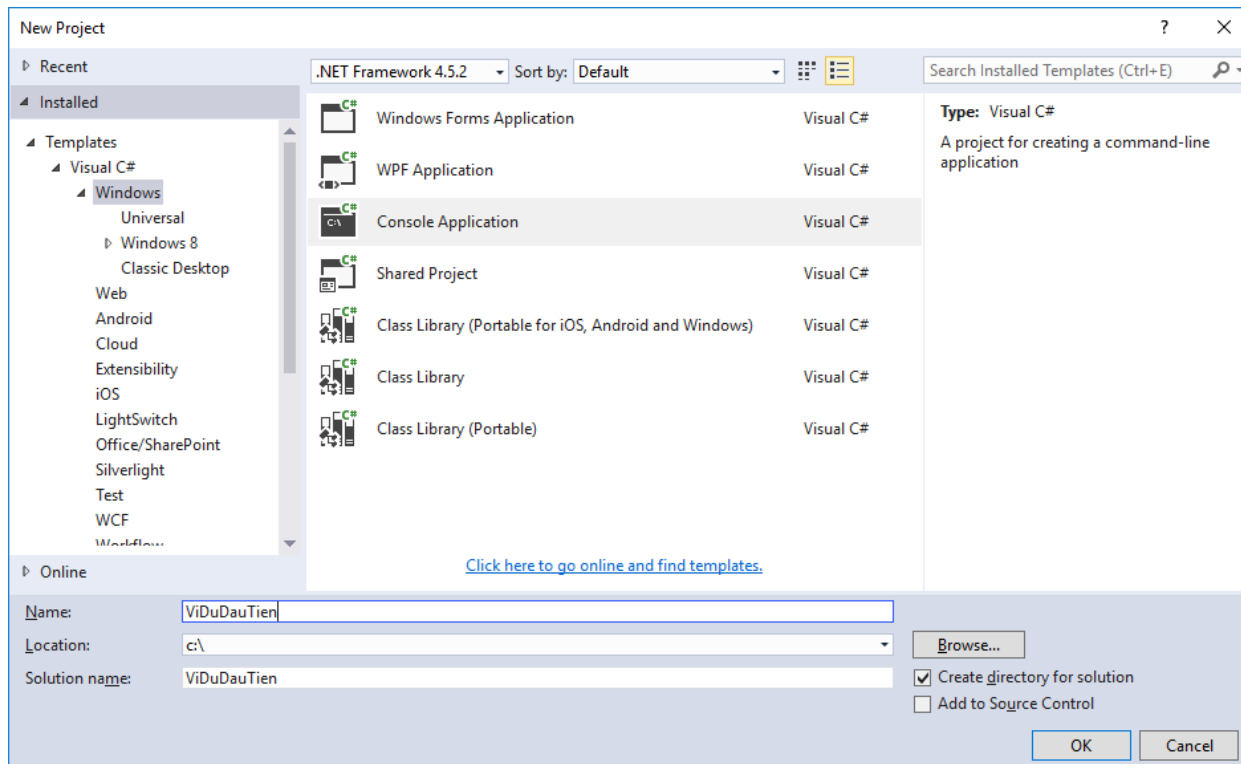


## 1.2.2. CẤU TRÚC 1 PROJECT

- Các *thuộc tính* (Properties): bao gồm nhiều loại thông tin khác nhau quyết định những tính chất quan trọng của project, như phiên bản của .NET framework được sử dụng, loại chương trình mà dự án này sẽ được dịch thành, các tài nguyên được sử dụng trong project, cấu hình của ứng dụng, v.v.. Visual Studio cung cấp giao diện đồ họa để có thể dễ dàng quản lý các thông tin này. Giao diện này mở ra khi click đúp vào mục Properties của project.

## 1.2.3. CẤU TRÚC FILE/THƯ MỤC CỦA C# PROJECT

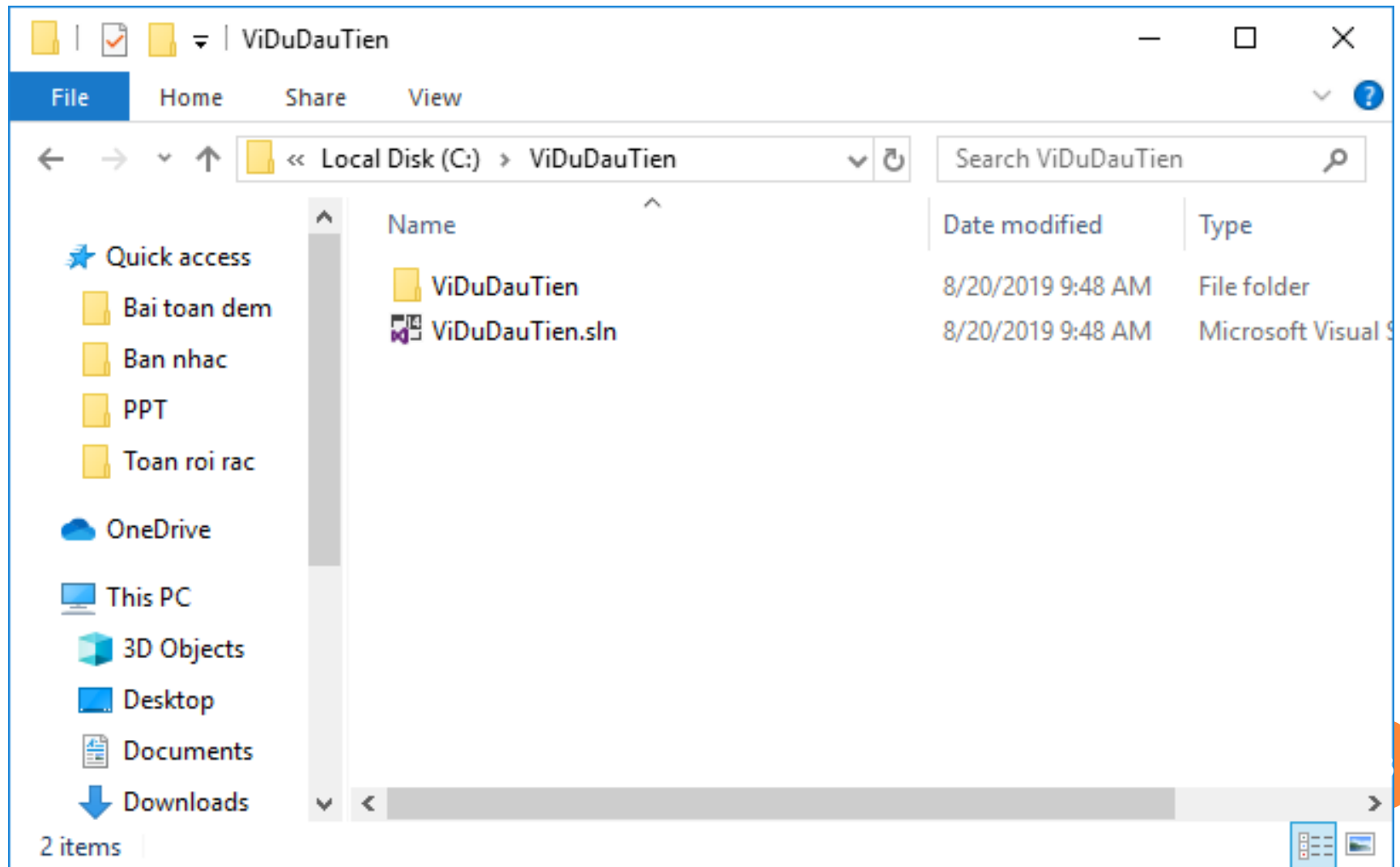
- Nếu ở giai đoạn tạo project chúng ta đánh dấu lựa chọn “Create directory for solution”, VS sẽ tạo ra một thư mục có tên được xác định trong mục “Solution name”



### 1.2.3. CẤU TRÚC FILE/THƯ MỤC CỦA C# PROJECT

- Mỗi project được tạo ra sẽ đặt trong một thư mục con của thư mục solution và có cùng tên với project
- Tất cả file của một project sẽ nằm trong thư mục này
- Toàn bộ cấu hình của solution được lưu trong một file có phần mở rộng **.sln** nằm trong thư mục chứa solution.
- Thông tin cấu hình của dự án được lưu trong file có tên trùng tên dự án và phần mở rộng **.csproj**.

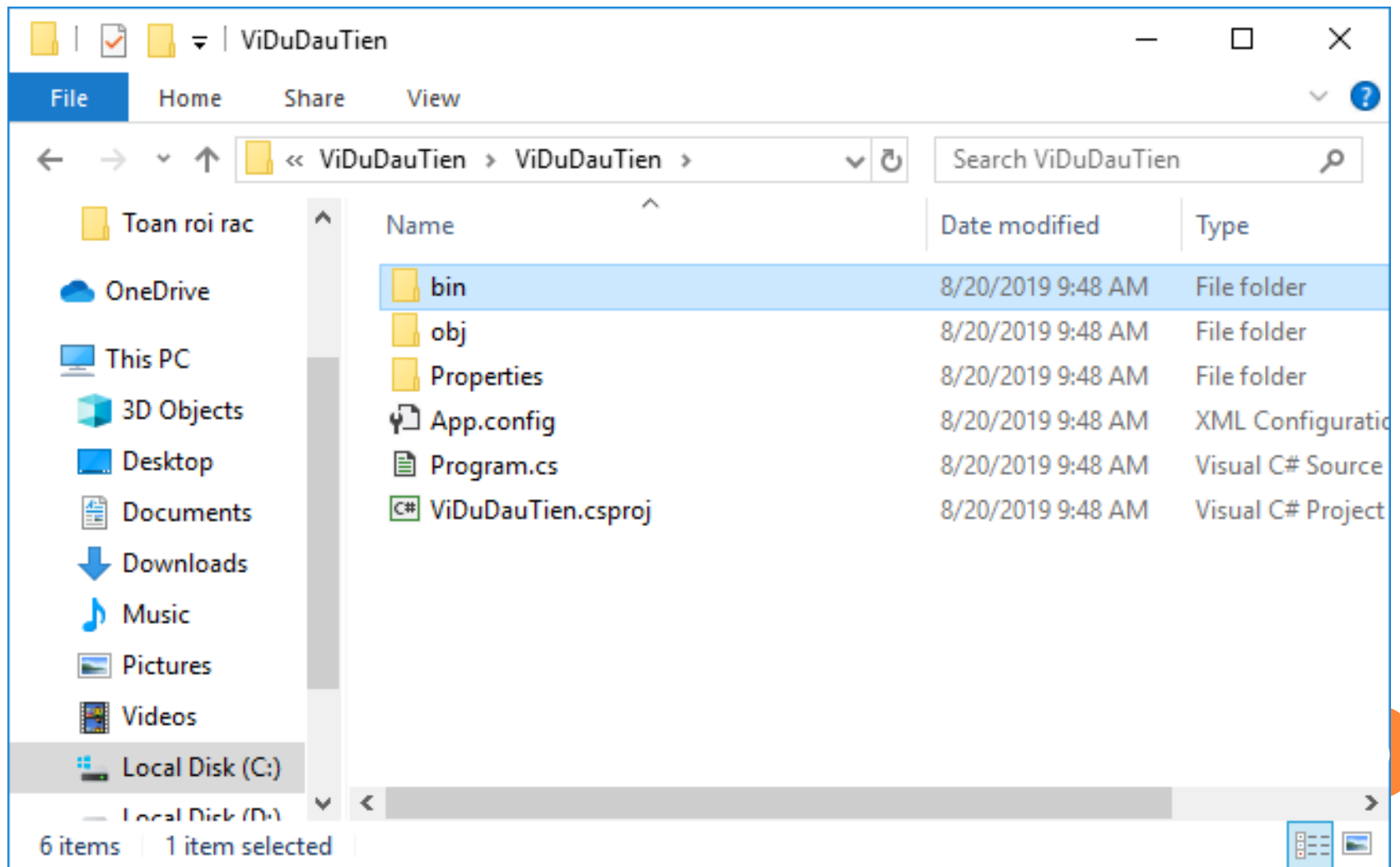
## 1.2.3. CẤU TRÚC FILE/THƯ MỤC CỦA C# PROJECT



### 1.2.3. CẤU TRÚC FILE/THƯ MỤC CỦA C# PROJECT

- Sau khi biên dịch project thành công, trong thư mục của nó sẽ xuất hiện một thư mục con có tên là *bin*. Thư mục này không xuất hiện trong cấu trúc dự án hiển thị ở Solution Explorer.

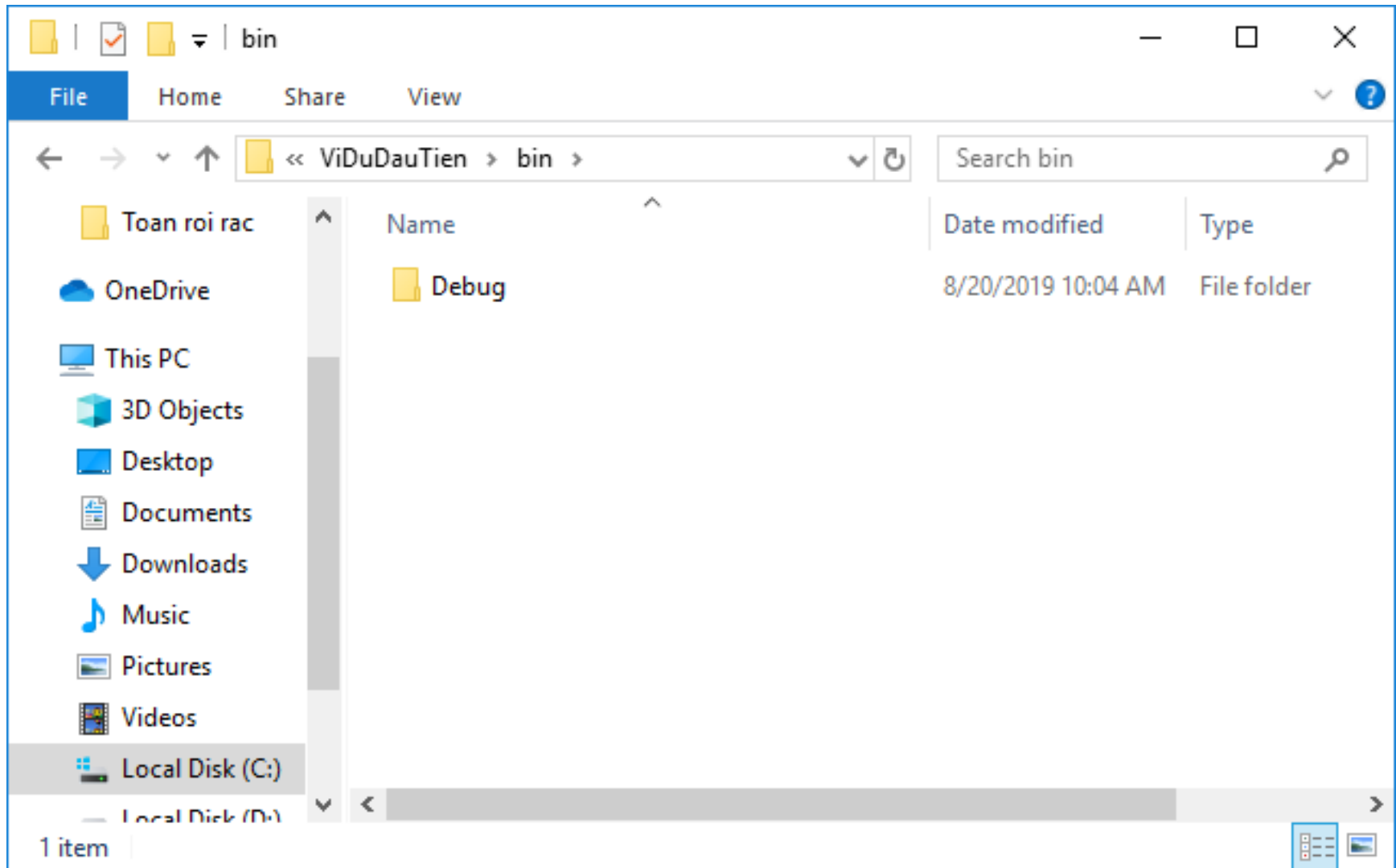
## 1.2.3. CẤU TRÚC FILE/THƯ MỤC CỦA C# PROJECT



### 1.2.3. CẤU TRÚC FILE/THƯ MỤC CỦA C# PROJECT

- Nếu biên dịch ở chế độ debug, trong thư mục bin sẽ xuất hiện thư mục con Debug. File chương trình sau khi biên dịch (ở chế độ debug) xong sẽ xuất hiện trong thư mục này.
- Nếu chạy chương trình ở chế độ debug (phím F5), thư mục Debug này sẽ trở thành thư mục hiện hành của chương trình đang chạy thử.

## 1.2.3. CẤU TRÚC FILE/THƯ MỤC CỦA C# PROJECT





## 1.2.4. TỪ KHOÁ – KEYWORDS

abstract	add*	as	base	bool	break	byte
case	catch	char	checked	class	const	continue
decimal	default	delegate	do	double	else	enum
event	explicit	extern	false	finally	fixed	float
for	foreach	get*	goto	if	implicit	in
int	interface	internal	is	lock	long	namespace
new	null	object	operator	out	override	params
partial*	private	protected	public	readonly	ref	remove
return	sbyte	sealed	set*	short	sizeof	stackalloc
static	string	struct	switch	this	throw	true
try	typeof	uint	ulong	unchecked	unsafe	ushort
using	value*	virtual	void	volatile	where*	while
yield						

## 1.2.5. NAMESPACE (KHÔNG GIAN TÊN)

Namespace là một khái niệm được sử dụng để phân nhóm các lớp đối tượng trong .Net Framework, tránh việc trùng tên giữa các lớp đối tượng

## 1.2.5. NAMESPACE (KHÔNG GIAN TÊN)

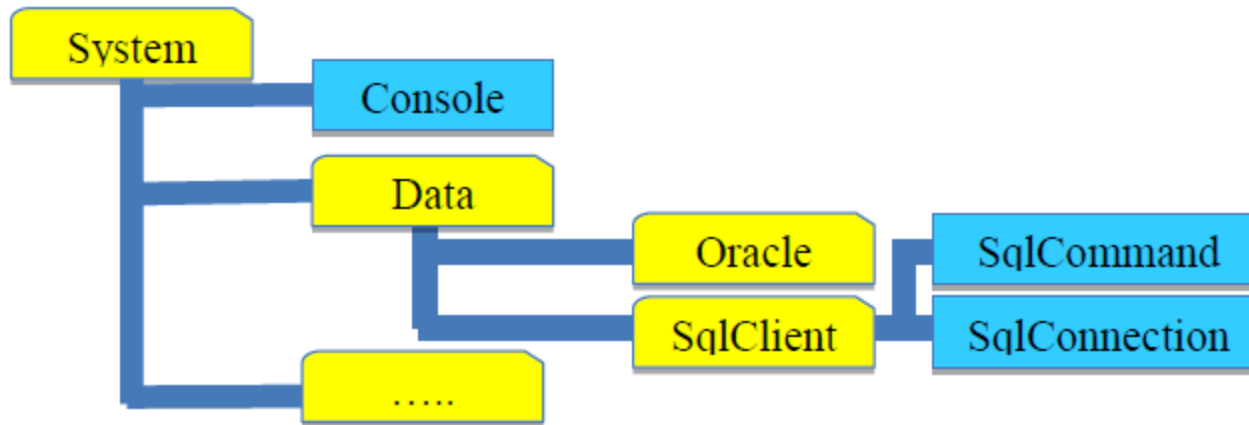
- Namespace được dùng để tổ chức quản lý code
- Namespace giúp viết code rõ ràng ràng mạch, cũng như hỗ trợ đắc lực trong việc quản lý các dự án lớn.
- Namespace có thể trực tiếp chứa các đơn vị code sau:
  - Namespace khác (namespace con)
  - Lớp (Class)
  - Giao diện (Interface)
  - Cấu trúc (Structure)
  - Kiểu ủy nhiệm (Delegate)

## 1.2.5. NAMESPACE (KHÔNG GIAN TÊN)

- Không gian tên tổ chức dạng phân cấp, y như việc phân cấp của cây thư mục
- Trong một không gian tên có thể chứa nhiều lớp, các lớp trong cùng một không gian tên không được trùng tên

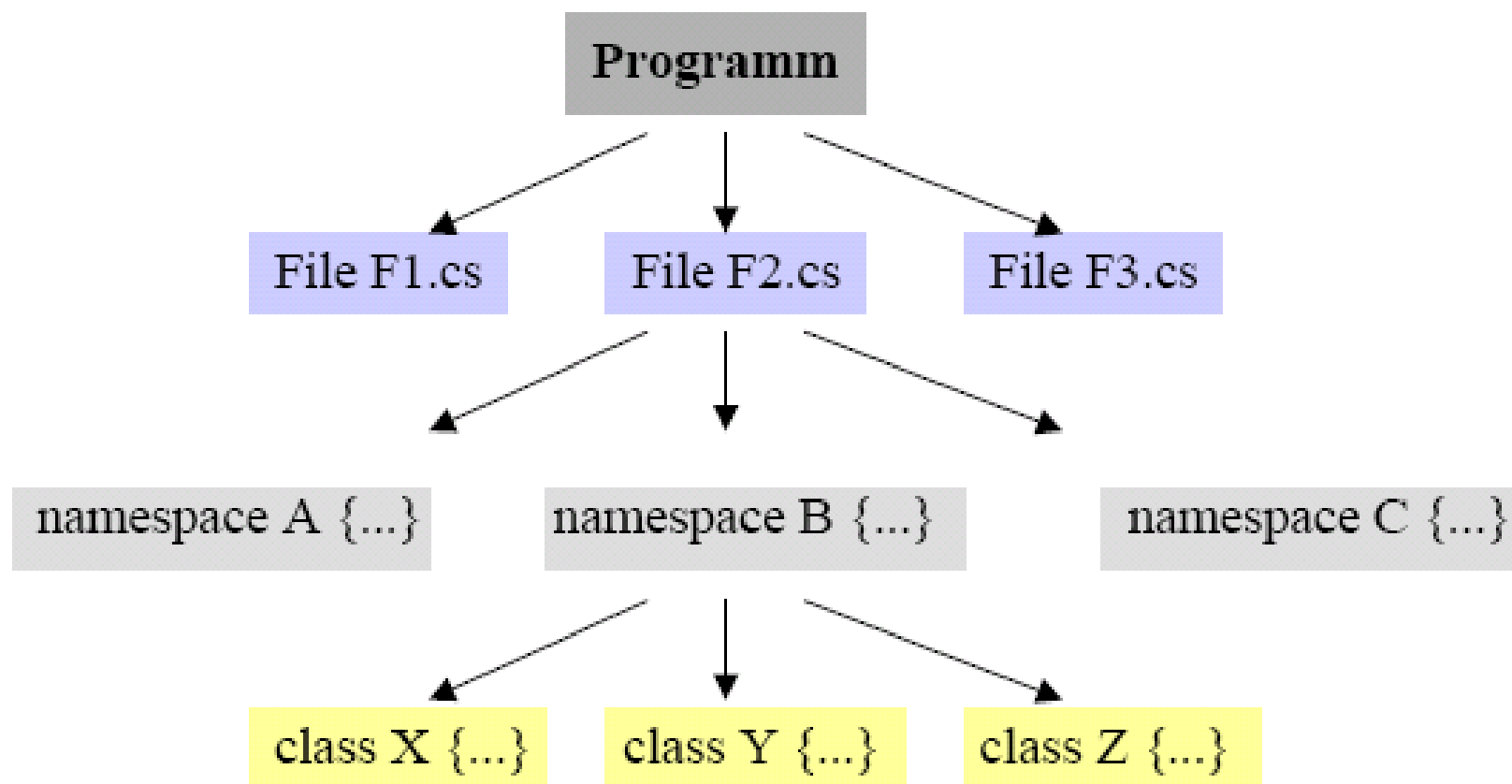
## 1.2.5. NAMESPACE (KHÔNG GIAN TÊN)

- Tổ chức phân cấp của không gian tên trong C#



- Không gian tên System chứa không gian tên Data
- Không gian tên Data chứa không gian tên Oracle và SqlClient
- Trong không gian tên SqlClient chứa nhiều lớp khác tên nhau như: lớp SqlCommand, lớp SqlConnection, ...

## 1.2.5. NAMESPACE (KHÔNG GIAN TÊN)



## 1.2.6. SỬ DỤNG NAMESPACE

- C# xây dựng sẵn nhiều lớp đối tượng đặt trong các không gian tên khác nhau phục vụ cho việc lập trình.
- Để sử dụng không gian tên trong C#, cần sử dụng từ khóa **using**.
- Ví dụ:

```
using System;
```

```
using System.Windows.Forms;
```

## 1.2.6. SỬ DỤNG NAMESPACE

- Khi tạo một dự án kiểu Windows Application, mặc định dự án sẽ sử dụng 8 không gian tên sau:

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;
```



## 1.2.6. SỬ DỤNG NAMESPACE

### ○ Ý nghĩa của các không gian tên như sau:

- Không gian tên System: Chứa các lớp và các kiểu dữ liệu cơ sở (int, double, byte, short, ...).
- Không gian tên System.Data: Chứa các lớp của ADO.NET, các lớp giúp kết nối hoặc thao tác với cơ sở dữ liệu (SQL Server, Oracle, Microsoft Access).
- Không gian tên System.Drawing: Chứa các lớp làm việc với đồ họa.

## 1.2.6. SỬ DỤNG NAMESPACE

- Không gian tên `System.ComponentModel`: Chứa các lớp cơ sở và giao diện giúp thực thi chuyển đổi kiểu, thuộc tính, liên kết với các dữ liệu nguồn và các thành phần cấp phép
- Không gian tên `System.Collections.Generic`: Chứa các lớp hỗ trợ cho việc thiết lập các tập hợp dạng chung mẫu (template) như:
  - `ArrayList List<T>`: Tập hợp chung mẫu có thể thay đổi kích thước khi vận hành.
  - `Queue Queue<T>`: Tập hợp chung mẫu theo mô hình vào trước ra trước (FIFO)
  - `Stack Stack<T>`: Tập hợp chung mẫu theo mô hình vào sau ra trước (LIFO)

## 1.2.6. SỬ DỤNG NAMESPACE

- Không gian tên `System.Text`: Cung cấp các lớp giúp mã hóa các ký tự Ascii hoặc Unicode.
- Không gian tên `System.Linq`: Cung cấp các lớp hỗ trợ cho việc sử dụng ngôn ngữ truy vấn tích hợp (Language-Integrated Query – LINQ).
- Không gian tên `System.Windows.Forms`: Là không gian tên chính cung cấp các lớp để xây dựng ứng dụng kiểu Windows Application. Không gian tên `System.Windows.Forms` chia thành các nhóm sau:

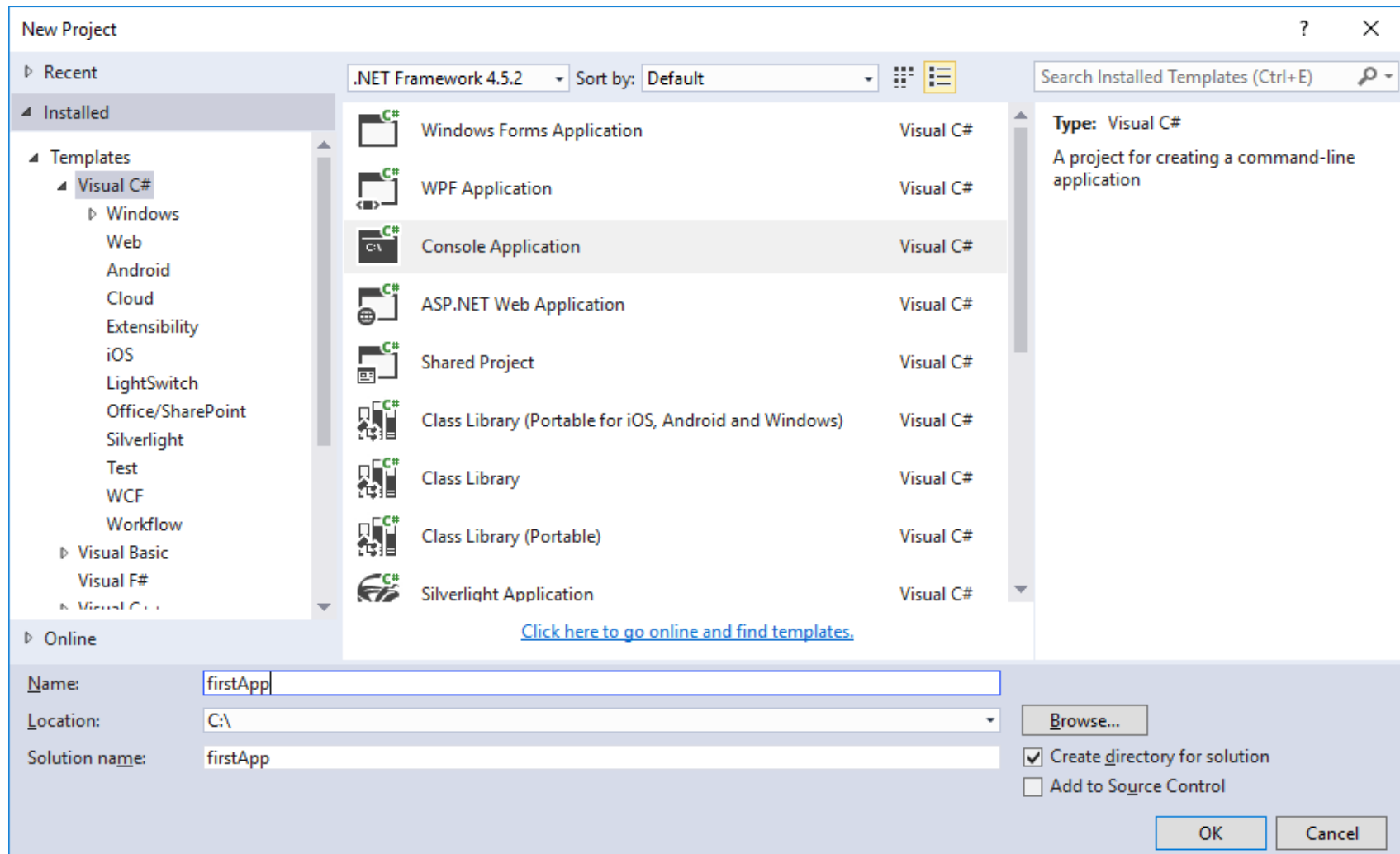
## 1.2.6. SỬ DỤNG NAMESPACE

- ✓ Control, UserControl, Form
- ✓ Menu, toolbar: ToolStrip, MenuStrip, ContextMenuStrip, StatusStrip
- ✓ Controls: TextBox, ComboBox, ListBox, ListView, WebBrowser, Label, ...
- ✓ Layout: FlowLayoutPanel, TableLayoutPanel, ... Giúp trình bày các điều khiển trên form
- ✓ Data và DataBinding: Gồm các công cụ BindingNavigator, BindingSource, ... giúp liên kết dữ liệu từ cơ sở dữ liệu đến các điều khiển Dataset hoặc DataGridView
- ✓ Command Dialog Boxes: Gồm các điều khiển OpenFileDialog, SaveFileDialog, ColorDialog, FontDialog, PrintDialog làm việc với tập tin, màu sắc, phong chữ, in ấn

## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN

- Ứng dụng này cho phép người sử dụng nhập vào 2 số, sau đó in ra màn hình tổng, tích và thương của hai số vừa nhập
- Khởi động Microsoft Visual Studio 20xx. Nhấn Ctrl + Shift + N hoặc chọn menu tương ứng là File ->New -> Project để tạo mới một project

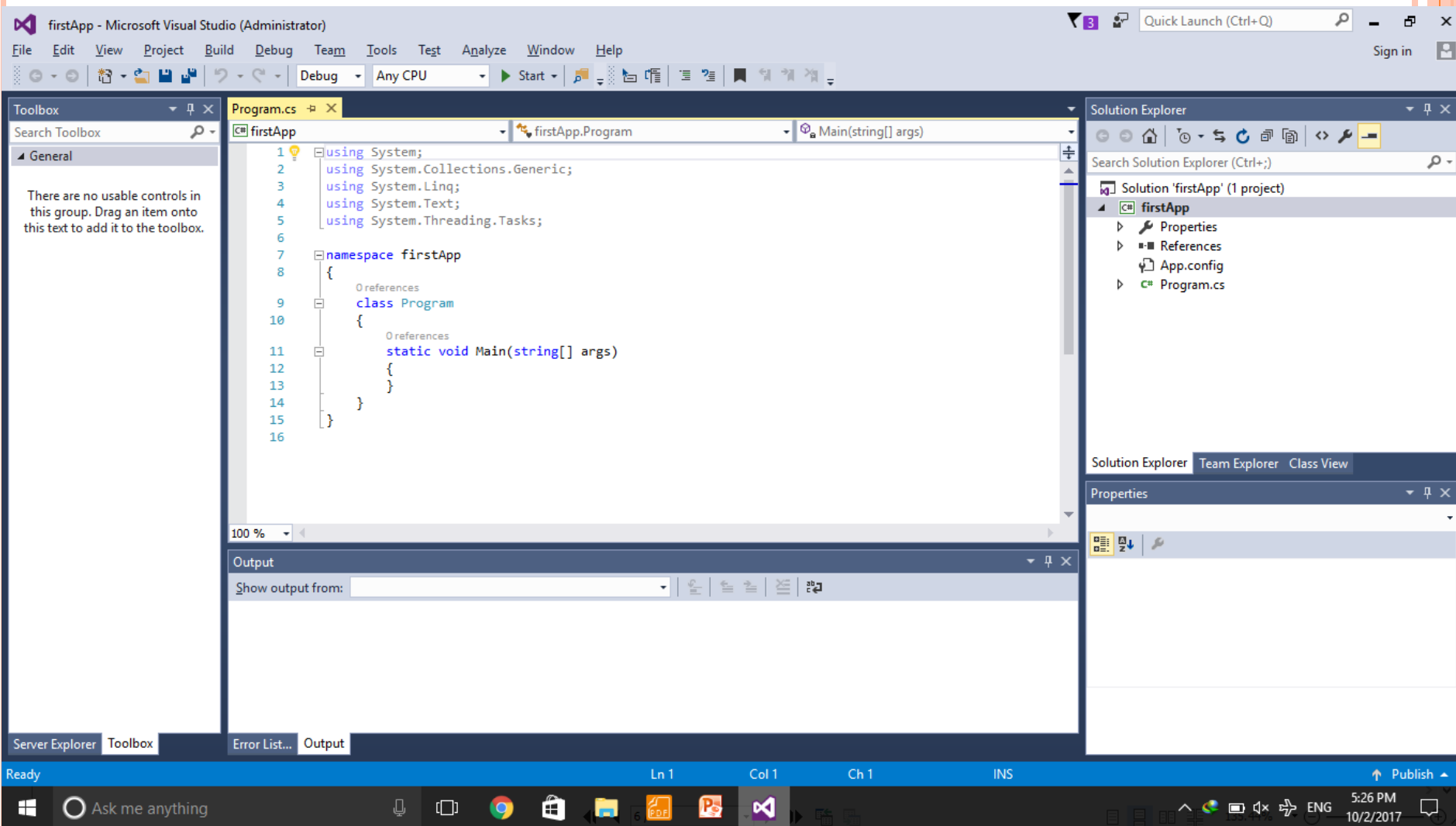
## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN



## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN

- Chọn loại ứng dụng cần phát triển là Visual C# -> Windows -> Console Application. Sau đó, chọn thư mục chứa project
- Chú ý, ở đây, chúng ta bỏ chọn ở hộp kiểm “Create directory for solution”.
- Sau khi nhấn nút OK

## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN

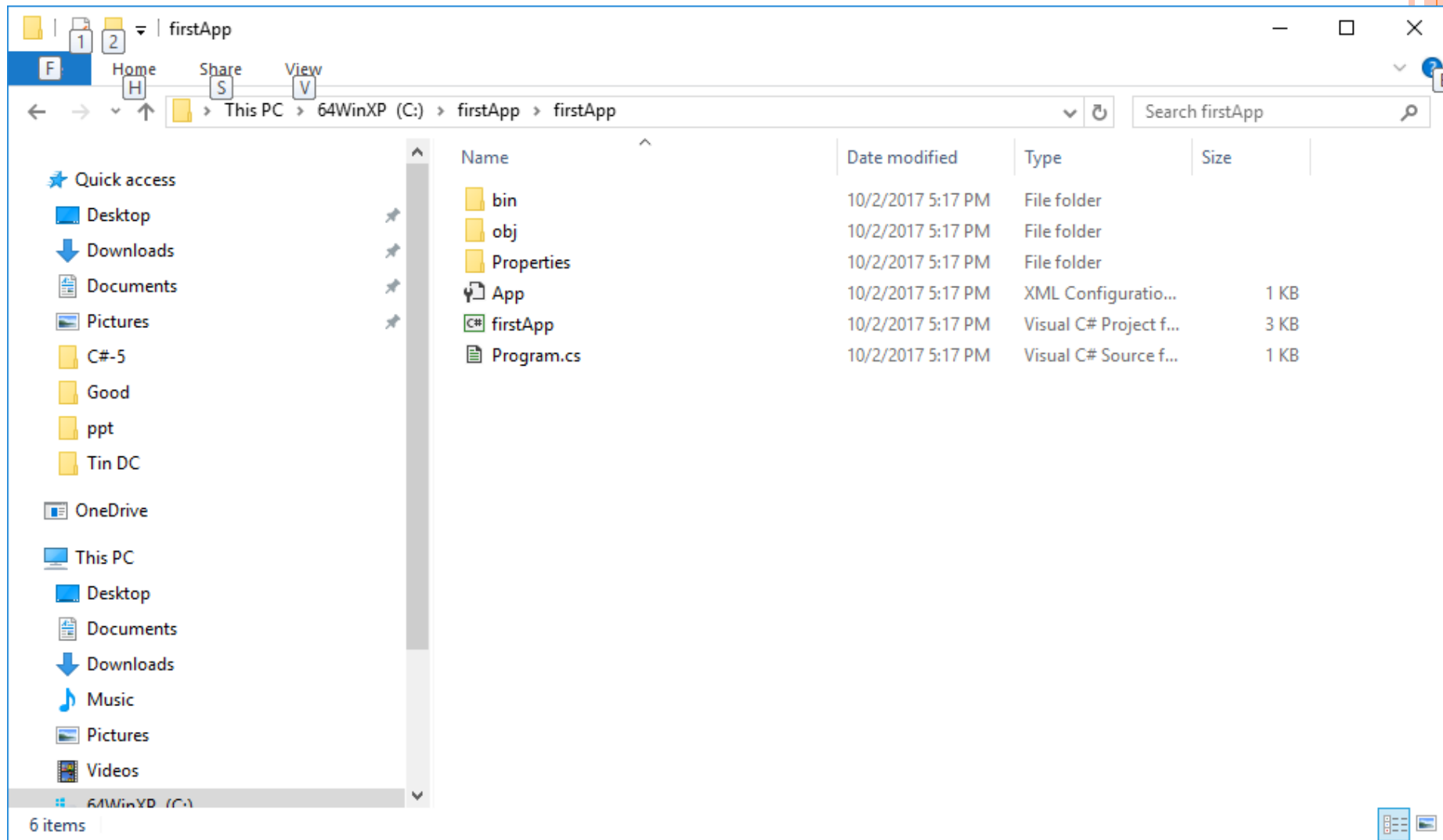




## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN

- Khảo sát xem cấu trúc của thư mục chứa solution của chúng ta.
- Bạn phải luôn nắm chắc về sự tồn tại, ý nghĩa của các tập tin, thư mục được tạo ra trong quá trình làm việc

## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN



## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN

Gõ mã lệnh như minh họa vào trong phần mã nguồn của tập tin Program.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace firstApp
6 {
7     class Program
8     {
9         static void Main(string[] args)
10         {
11             int x, y;
12             Console.Write("Số thứ nhất: "); //1. in câu thông báo "Số thứ nhất: " ra màn hình
13             string st = Console.ReadLine(); //2. lấy nội dung người sử dụng nhập vào gán cho st
14             x = int.Parse(st); //3. chuyển đổi st thành số nguyên và gán cho x
15
16             //Bước 2 và 3 ở trên có thể rút gọn như thế này
17             Console.Write("Số thứ hai: ");
18             y = int.Parse(Console.ReadLine());
19
20             //in kết quả
21             float tong = x + y;
22             Console.WriteLine("Tổng {0} và {1} là {2}", x, y, tong);
23
24             Console.WriteLine("Tích {0} và {1} là {2}", x, y, x * y);
25             float thuong = (float)x / y;
26             Console.WriteLine("Thương {0} và {1} là {2:0.000}", x, y, thuong);
27         }
28     }
29 }
```

**Phần gõ thêm vào**

**Biểu diễn với 3 chữ số thập phân**

## 1.2.7. CHƯƠNG TRÌNH ĐẦU TIÊN

### Compile & chạy chương trình

- Trình biên dịch (compiler) sẽ biên dịch các tập tin chứa ngôn ngữ C# thường là các file .cs trong project thành một tập tin chạy chương trình .exe
- Có 2 cách biên dịch:
  - Tại cửa sổ cmd, gõ : `csc.exe tenfile.cs`
  - Nhấn Build / Compile (hoặc Build / Build Solution)  $\Rightarrow$  Biên dịch cả project.
- Chạy chương trình
  - Sử dụng file `tenfile.exe` trong thư mục `Bin\Debug`
  - Hoặc click Debug\ Start (Ctrl + F5)