



LẬP TRÌNH C# 1

BÀI 1: LÀM QUEN VỚI C#

- ⊙ Tìm Hiểu Chương trình C#
- ⊙ Biến, kiểu, toán tử, biểu thức



Phần I: Tìm Hiểu Chương trình C#

 .Net framework

 Ngôn ngữ lập trình c#

 Chương trình c#

Phần II: Biến, kiểu, toán tử, biểu thức

 Lưu trữ dữ liệu

 Các hàm thông dụng

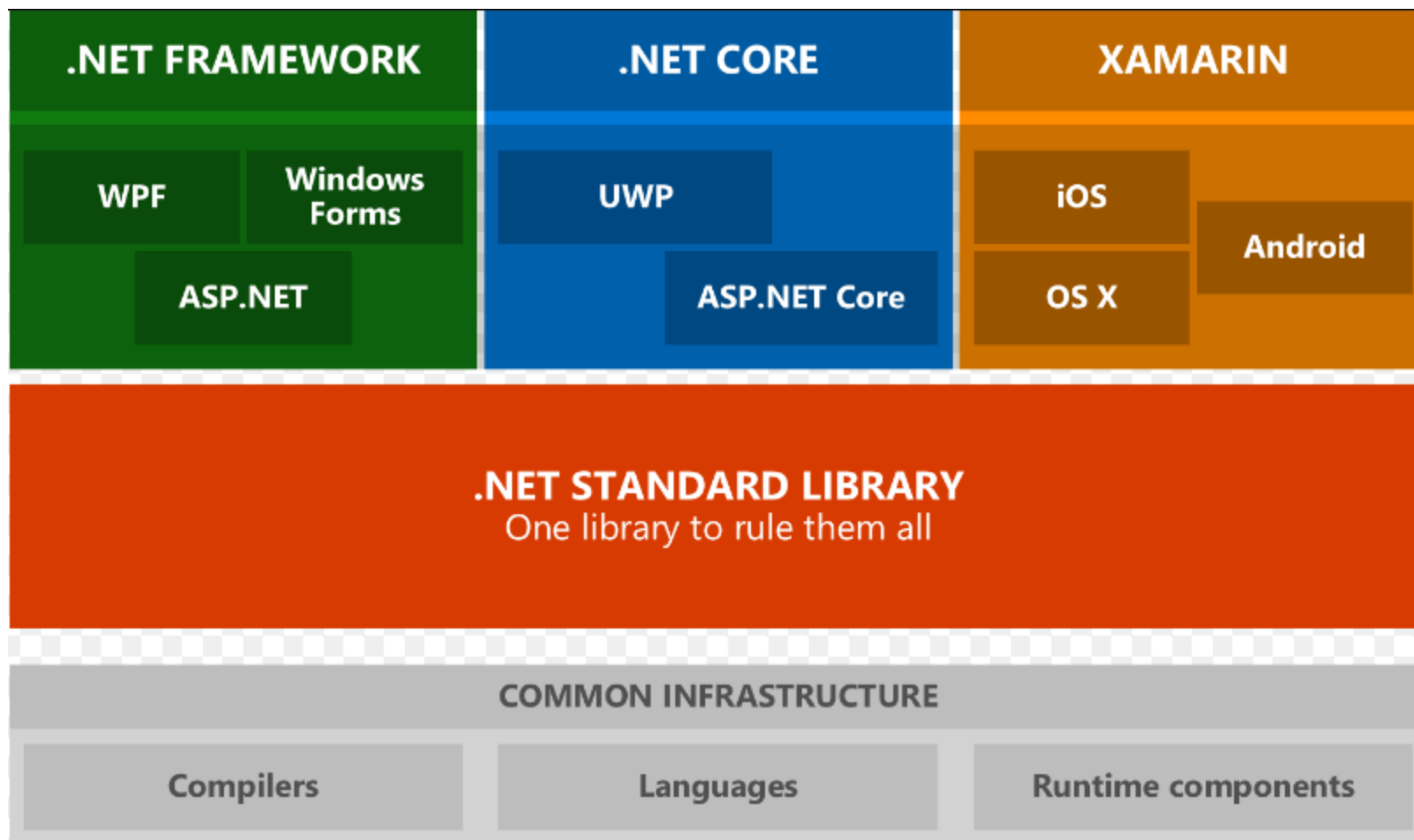


□ Giới thiệu .Net Framework

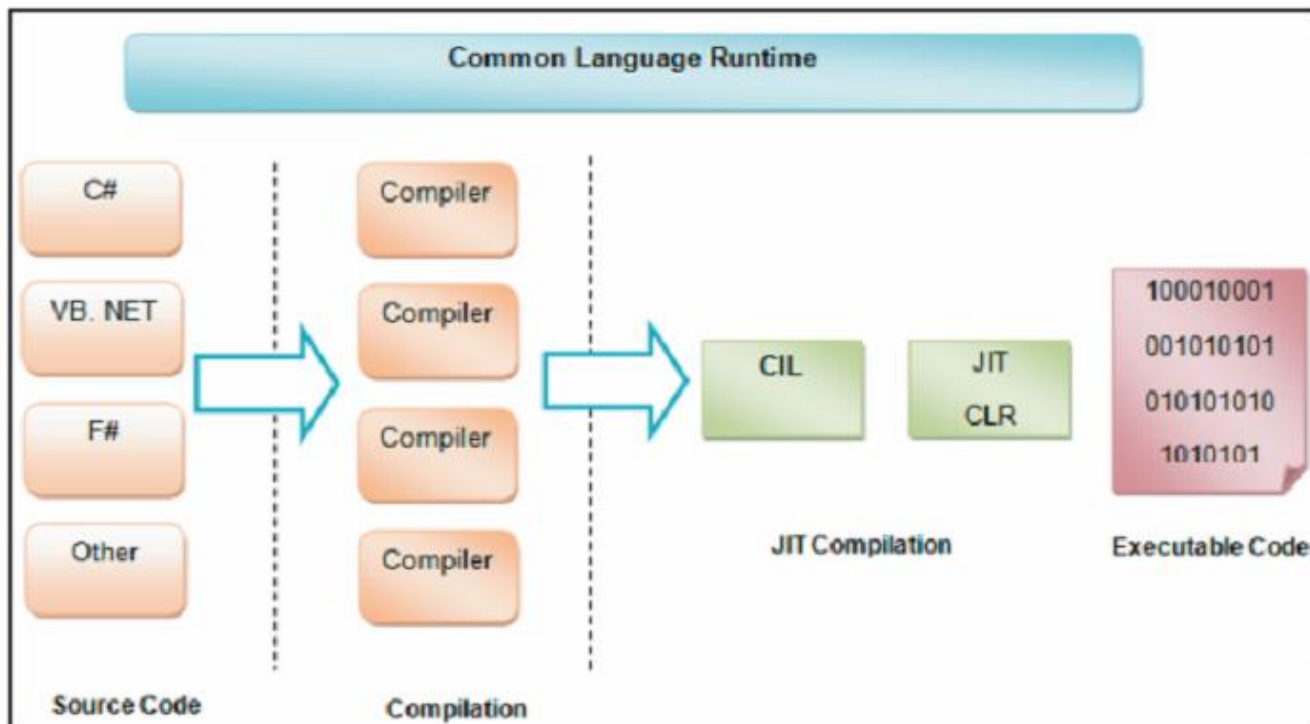
- ❖ Cho phép xây dựng, triển khai và chạy ứng dụng khác nhau và các dịch vụ sử dụng công nghệ .NET.
- ❖ Dùng .NET Framework giảm thiểu tối đa thời gian phát triển và triển khai phần mềm
- ❖ Giảm thiểu xung đột khi người sử dụng triển khai các hệ thống trên nhiều phiên bản khác nhau



- ❑ Bộ .Net Framework được tạo từ rất nhiều các thành phần khác nhau



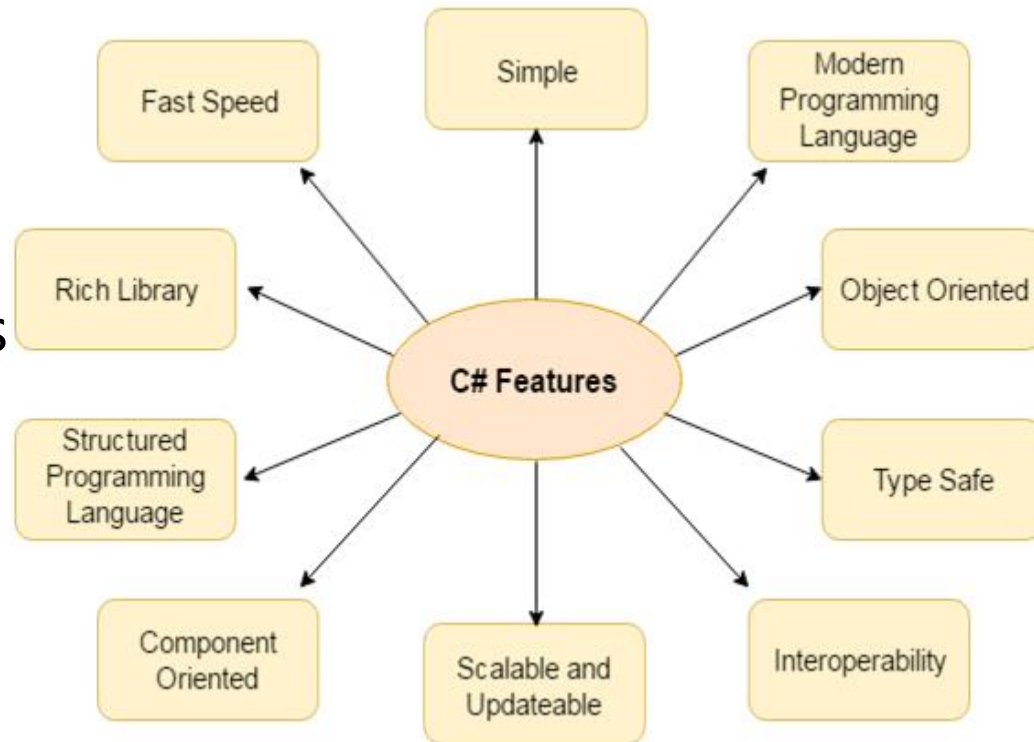
- ❑ Common Language Runtime: Là thành phần cơ bản của bộ .Net Framework.
 - ❖ Quản lý quá trình khi thực thi mã ứng dụng (.exe)
 - ❖ Quản lý bộ nhớ, quản lý phân luồng, quản lý truy cập
 - ❖ Cung cấp môi trường cho chương trình chạy



- ❑ C# hay Csharp là một trong những ngôn ngữ lập trình trong họ DotNet của Microsoft và khi lập trình C# người lập trình phải chuẩn bị và cài đặt .NET Framework.
- ❑ C# được phát triển bởi Anders Hejlsberg và team của ông trong khi phát triển .Net Framework
- ❑ C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment
- ❑ C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java

❑ Các đặc trưng cơ bản của ngôn ngữ C#

- ❖ Ngôn ngữ lập trình hướng đối tượng
- ❖ Tuân theo chuẩn ECMA
- ❖ Bộ thu gom rác tự động
- ❖ An toàn kiểu
- ❖ Hỗ trợ Generic
- ❖ Partial Classes
- ❖ Anonymous Methods
- ❖ Nullable Type
- ❖ Lập trình song song



❑ Các loại ứng dụng khác nhau của C#

- ❖ Xây dựng ứng dụng trên nền tảng web, web service
- ❖ Game
- ❖ Ứng dụng Windows Store
- ❖ Mobile
- ❖ Các ứng dụng Enterprise
- ❖ Các ứng dụng Desktop
- ❖ Add On/Add in
- ❖ Điện toán đám mây



- ❑ Cài đặt .NET Framework
- ❑ Công cụ phát triển C#: Visual Studio Code, Project Rider, MonoDevelop
- ❑ Khuyến khích dùng Visual Studio <https://visualstudio.microsoft.com/downloads/>



Version: 15.9
[Release notes](#)


[Compare editions](#)
[How to install offline](#)

Visual Studio 2017

Full-featured integrated development environment (IDE) for Android, iOS, Windows, web, and cloud


Community

Powerful IDE, free for students, open-source contributors, and individuals

Free download 


Professional

Professional IDE best suited to small teams

Free trial 

Enterprise

Scalable, end-to-end solution for teams of any size

Free trial 

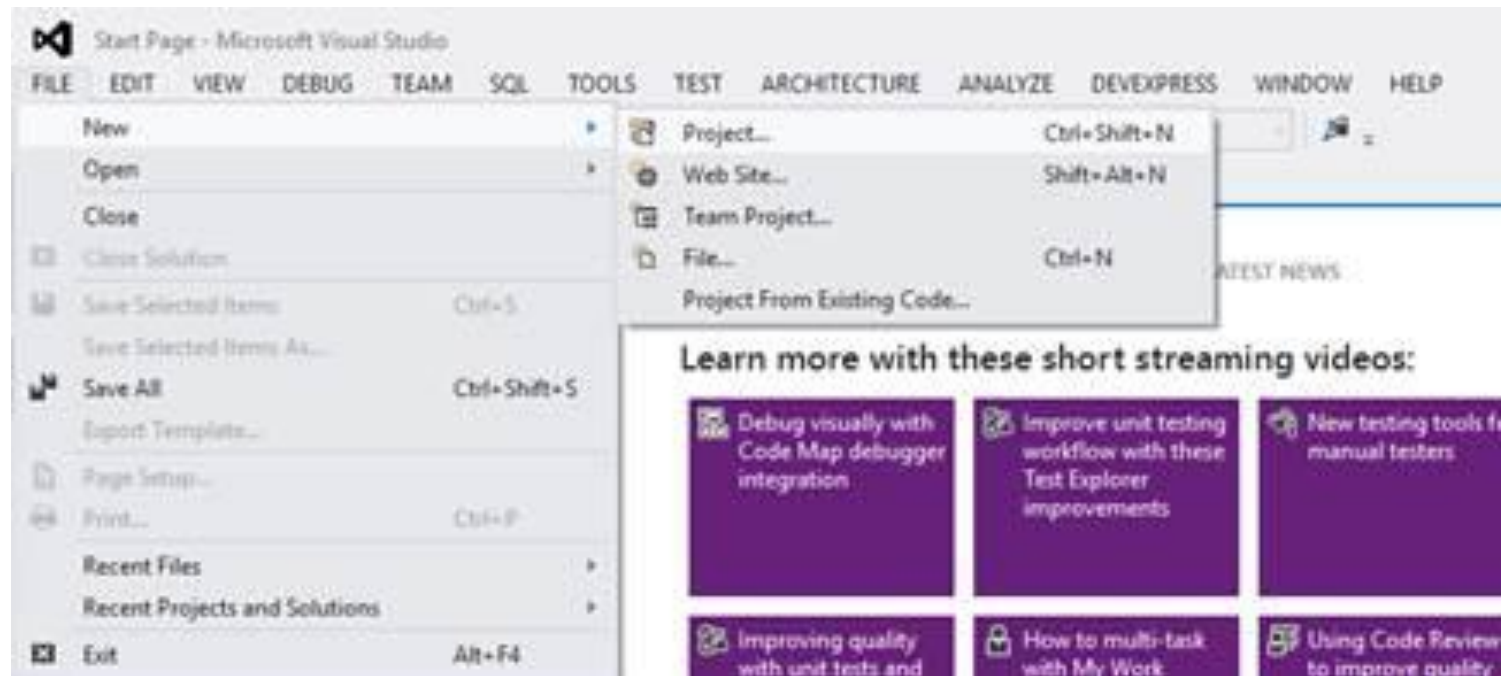
- ❑ Một chương trình sẽ gồm các phần sau: Namespace, class, methods, attributes, Main method, Statements và expressions, Comments

```
using System;

namespace Fpoly
{
    class TestCsharp
    {
        static void Main(string[] args)
        {
            /* chương trình in dòng chu Hello World trong C# */
            Console.WriteLine("Hello World");
            Console.ReadKey();
        }
    }
}
```

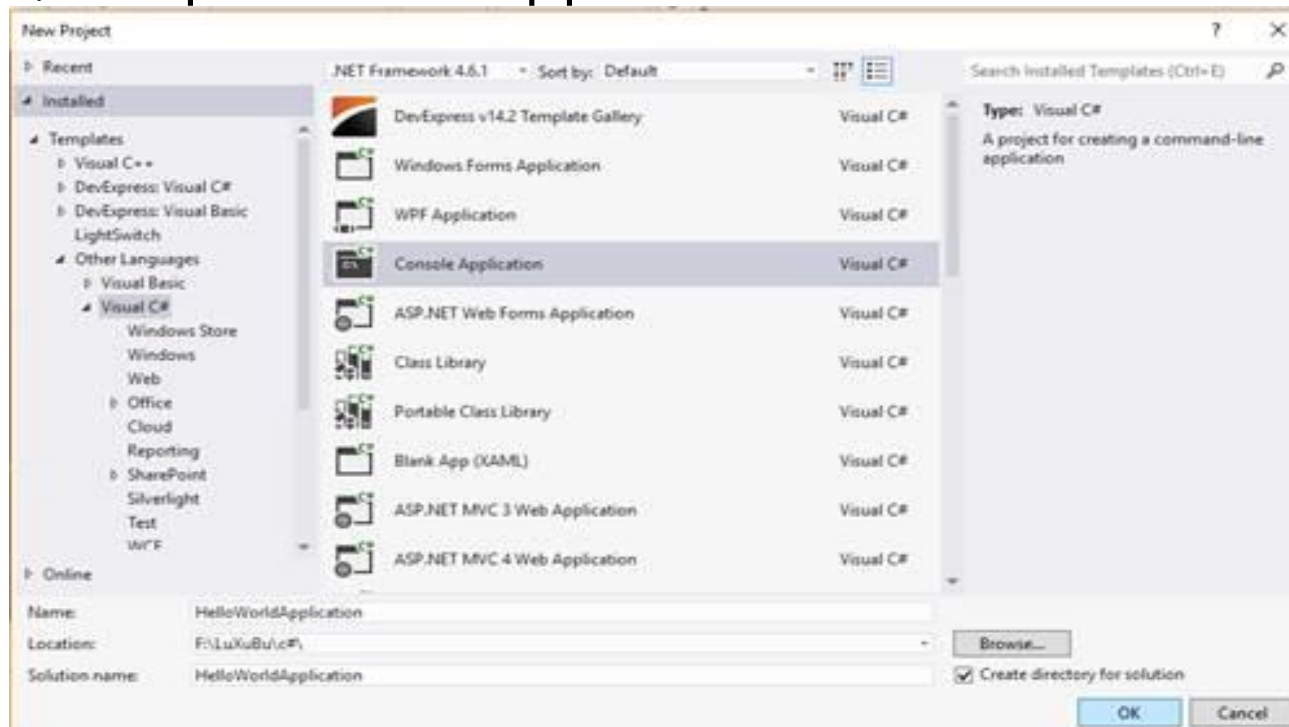
CẤU TRÚC MỘT CHƯƠNG TRÌNH C#

- ❑ Tạo ứng dụng console c# với VS: chọn File->New->Project hoặc có thể sử dụng tổ hợp phím Ctrl+Shift+N.



❑ Tạo ứng dụng console c# với VS:

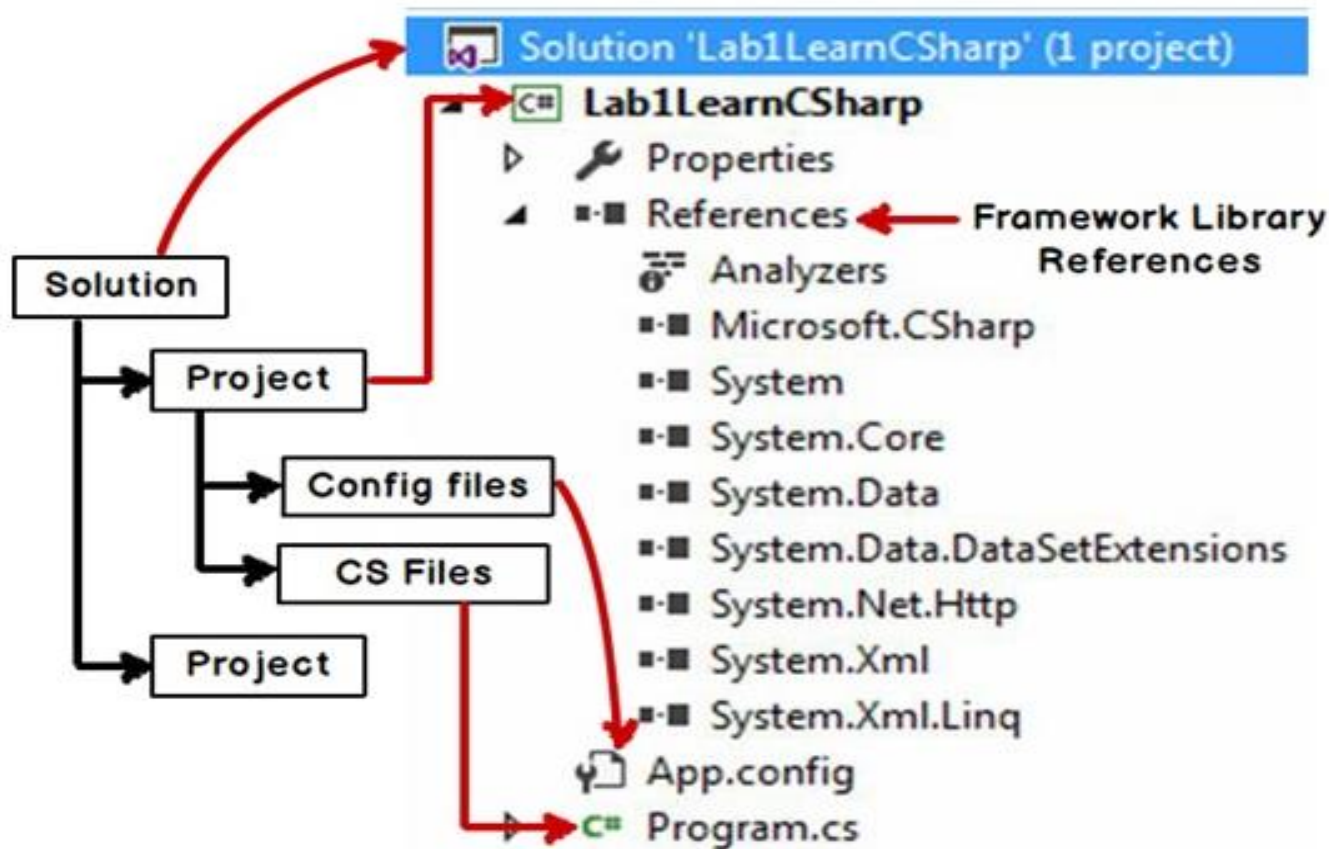
- ❖ Chọn visual C# từ các mẫu cung cấp và chọn Windows
- ❖ Chọn tiếp Console Application



- ❖ Đặt tên cho project và click OK

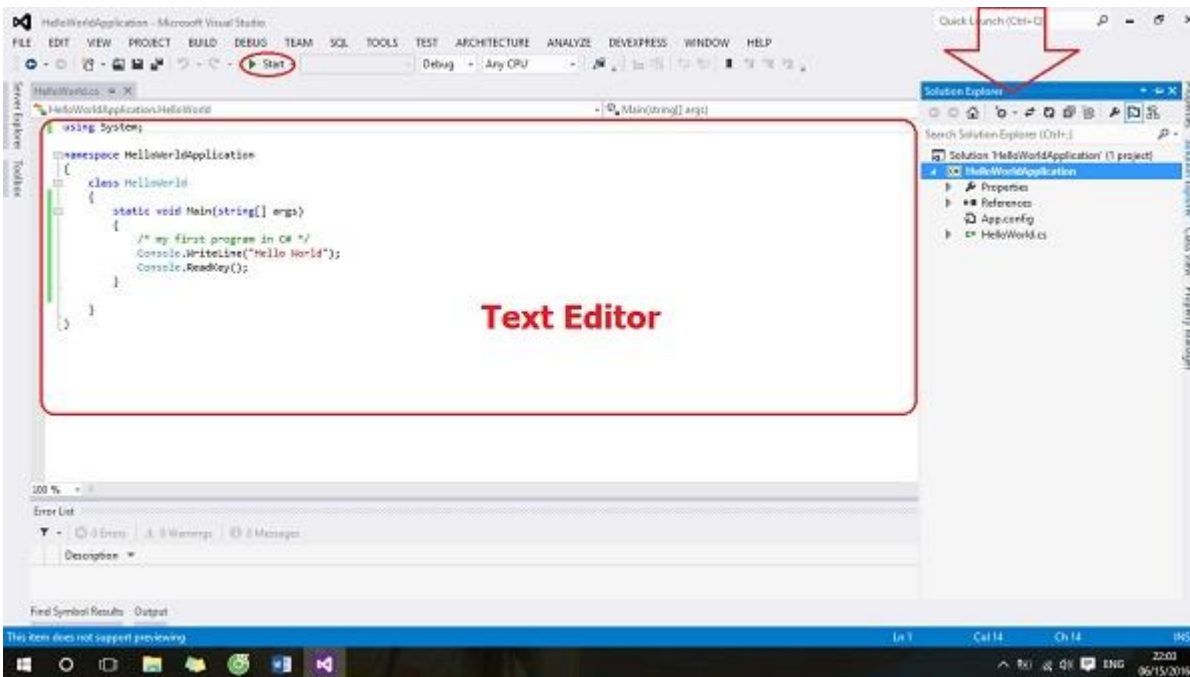
CẤU TRÚC MỘT CHƯƠNG TRÌNH C#

□ Cấu trúc thư mục, tập tin trên VS



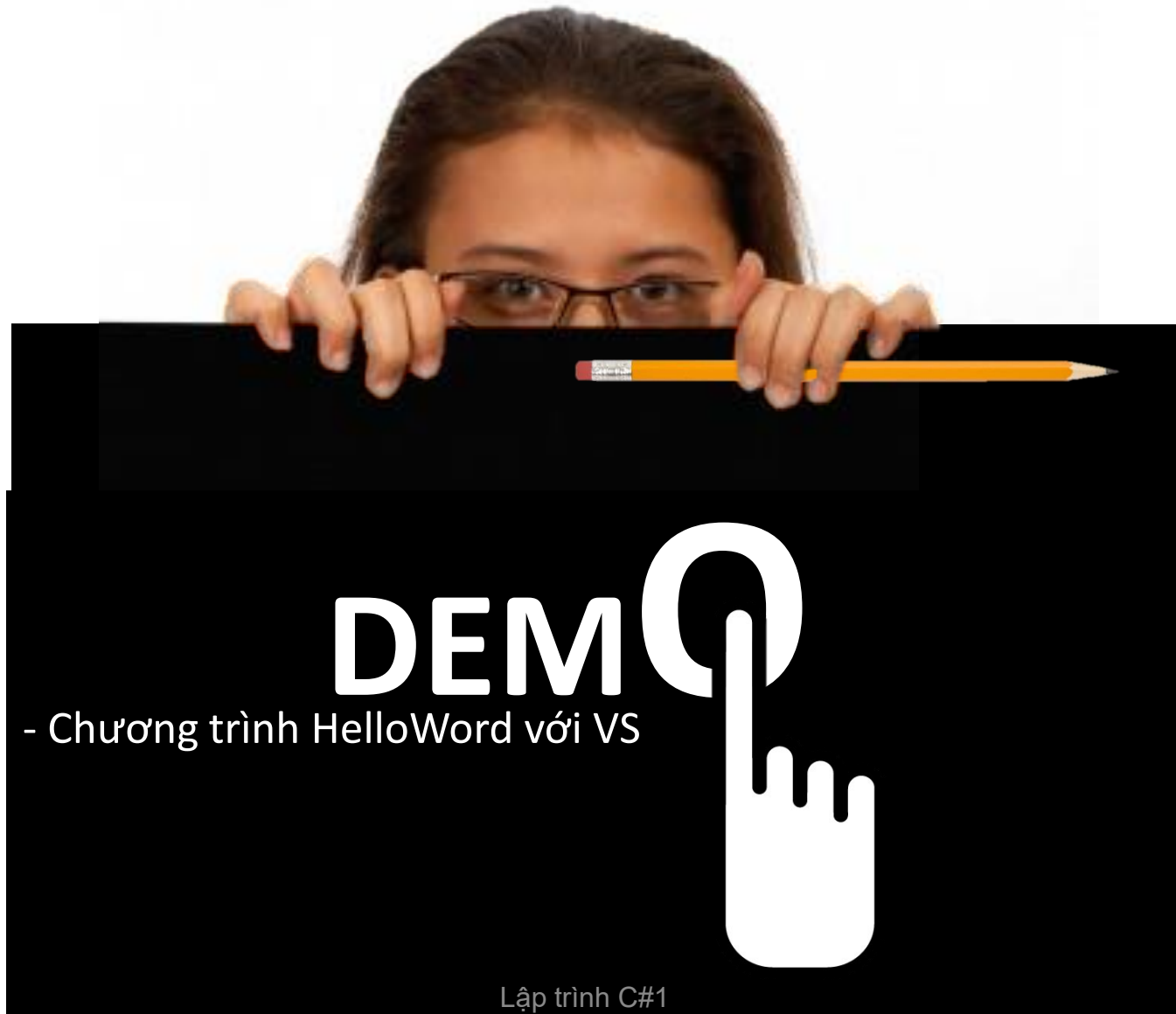
CẤU TRÚC MỘT CHƯƠNG TRÌNH C#

- ❑ Đã tạo được một project trong Solution Explorer.
- ❑ Viết code trong vùng code Editor
- ❑ Click Start hoặc F5 để thực thi project. Một màn hình console sẽ xuất hiện hiển thị kết quả



```
C:\Windows\system32\cmd.exe
Start Unit 1 !
Press any key to continue . . .
```

CẤU TRÚC MỘT CHƯƠNG TRÌNH C#



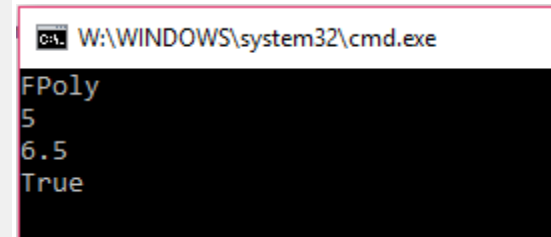


LẬP TRÌNH C# 1

BÀI 1: LÀM QUEN VỚI C# (P2)

- ❑ Xử lý output, sử dụng một trong 2 phương thức bên dưới
 - ❖ Console.**Write**(<giá trị cần in ra màn hình>): in giá trị ra màn hình nhưng không đưa con trỏ xuống dòng
 - ❖ Console.**WriteLine**(<giá trị cần in ra màn hình>): in giá trị ra màn hình và đưa con trỏ xuống dòng

```
1 static void Main(string[] args)
2 {
3     Console.Write("FPoly\n"); // Sử dụng ký tự đặc biệt để xuống dòng
4     Console.WriteLine(5); // Sử dụng lệnh in ra màn hình có xuống dòng
5     Console.Write(6.5f); // In ra giá trị nhưng không xuống dòng
6     Console.Write(Environment.NewLine); // sử dụng lệnh xuống dòng
7     Console.Write(true);
8
9     Console.ReadLine();
10 }
11
```



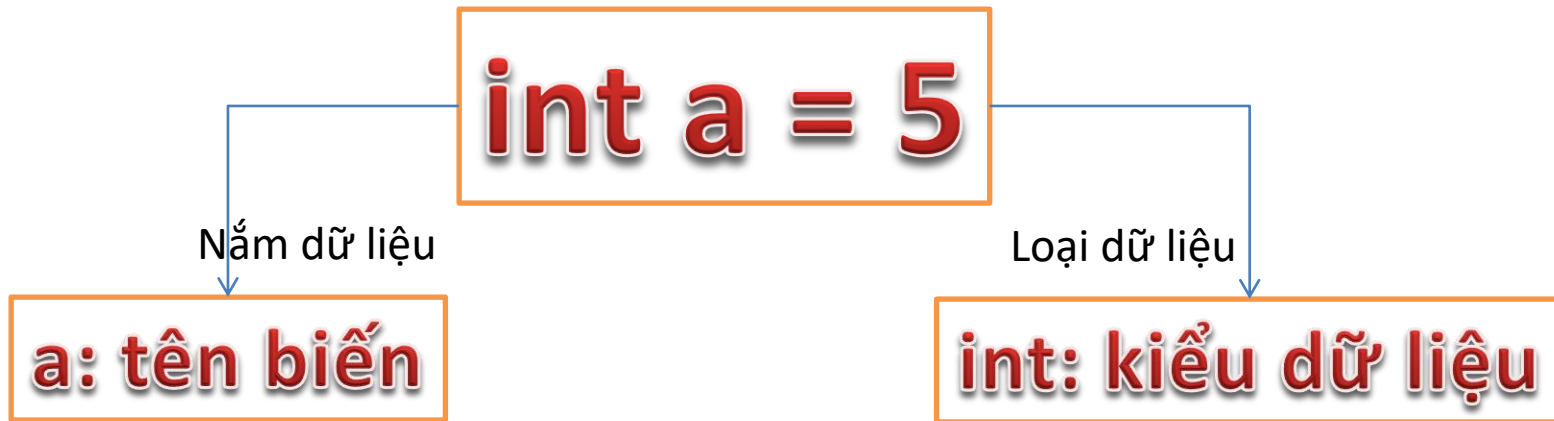
```
W:\WINDOWS\system32\cmd.exe
FPoly
5
6.5
True
```

- ❑ Xử lý input, sử dụng một trong 2 phương thức bên dưới
 - ❖ Console.Read(): Nhập một ký tự dưới dạng số nguyên
 - ❖ Console.ReadLine(): Nhập nhiều ký tự

```
class Program
{
    static void Main(string[] args)
    {
        string name;
        Console.Write("Vui long nhap vao ten cua ban: ");
        name = Console.ReadLine();
        Console.WriteLine("Ten cua ban la " + name);
    }
}
```

```
namespace Fpoly
{
    //References
    class Program
    {
        //References
        static void Main(string[] args)
        {
            int sohang1 = 8;
            int sohang2 = 4;
            int tong = sohang1 + sohang2;
            Console.WriteLine("Tổng là: " + tong);
        }
    }
}
```

- ❑ Đoạn mã trên gán các giá trị 8 cho sohang1, 4 cho sohang2 và tổng sohang1 + sohang2 cho tong sau đó xuất tổng ra màn hình
- ❑ Sohang1, sohang2 và tong gọi là biến số nguyên
- ❑ Biến là thành phần nắm giữ dữ liệu được chương trình sử dụng trong các biểu thức tính toán
- ❑ Mỗi biến có kiểu dữ liệu riêng



❑ Biến là thành phần nắm giữ dữ liệu được chương trình sử dụng trong các biểu thức tính toán
(*biến a nắm giữ số 5*)

- ❑ int: Số nguyên
- ❑ double : số thực
- ❑ String: Chuỗi
- ❑ ...

□ Cú pháp

<kiểu dữ liệu> <tên biến> [=giá trị khởi đầu];

□ Ví dụ:

`int a;` // khai báo biến không khởi đầu giá trị

`double b = 5;` // khai báo biến có khởi đầu giá trị

□ Khai báo nhiều biến cùng kiểu

`int a, b=5, c;`

□ Gán giá trị cho biến

`c = 9;`

`a = 15;`

□ Cú pháp

<kiểu dữ liệu> <tên biến> [=giá trị khởi đầu];

□ Ví dụ:

`int a;` // khai báo biến không khởi đầu giá trị

`double b = 5;` // khai báo biến có khởi đầu giá trị

□ Khai báo nhiều biến cùng kiểu

`int a, b=5, c;`

□ Gán giá trị cho biến

`c = 9;`

`a = 15;`



DEMO

Khai báo 2 biến số nguyên a, b và c
Thực hiện phép cộng a và b được c
Xuất kết quả c



❑ Danh sách một số kiểu dữ liệu trong C#

Kiểu	Biểu diễn	Dãy giá trị	Giá trị mặc định
bool	Giá trị Boolean	True hoặc False	False
byte	Kiểu unsigned integer (8 bit)	0 tới 255	0
char	Kiểu Unicode character (16 bit)	U +0000 tới U +ffff	'\0'
decimal	Kiểu thập phân (128 bit)	$(-7.9 \times 10^{28}$ tới $7.9 \times 10^{28}) / 10^0$ to 28	0.0M
double	Kiểu double (64 bit)	$(+/-)5.0 \times 10^{-324}$ tới $(+/-)1.7 \times 10^{308}$	0.0D
float	Kiểu float (32 bit)	-3.4×10^{38} tới $+ 3.4 \times 10^{38}$	0.0F
int	Kiểu integer (32 bit)	-2,147,483,648 tới 2,147,483,647	0
long	Kiểu signed integer (64 bit)	-9,223,372,036,854,775,808 tới 9,223,372,036,854,775,807	0L
sbyte	Kiểu signed integer (8 bit)	-128 tới 127	0

❑ Chuyển đổi kiểu ngầm định (implicit)

- ❖ Việc chuyển đổi được thực hiện bởi trình biên dịch theo một phương thức an toàn kiểu (type-safe)
- ❖ Từ miền giá trị nhỏ sang miền giá trị lớn
- ❖ Từ lớp dẫn xuất sang lớp cơ sở

```
// 1. Ép kiểu từ kiểu có miền giá trị nhỏ qua kiểu có miền giá trị lớn
byte b = 7;
int i = b;          /* Kiểu int có miền giá trị lớn hơn kiểu byte */

float flt = 13.3f;
double dbl = flt;   /* Kiểu double có miền giá trị lớn hơn kiểu float */

// 2. Ép kiểu từ lớp dẫn xuất(string) qua lớp cơ sở(object)
string s = "Hello World";
object o = s;
```

❑ Chuyển đổi kiểu tường minh (explicit)

- ❖ Được thực hiện một cách rõ ràng bởi người dùng bằng việc sử dụng các hàm được định nghĩa trước. Các chuyển đổi kiểu tường minh cần một toán tử cast.
- ❖ Cú pháp: (<kiểu dữ liệu>) <biến cần ép kiểu>:

```
static void Main(string[] args)
{
    Console.WriteLine("Chuyen doi kieu du lieu trong C#");
    Console.WriteLine("-----");

    double d = 5678.74;
    int i;

    // cast kieu du lieu double thanh kieu du lieu int.
    i = (int)d;
    Console.WriteLine("Gia tri cua i = " +i);
    Console.ReadKey();
}
```

❑ Chuyển đổi kiểu tường minh (explicit)

❖ Sử dụng phương thức Parse()

```
string stringValue = "123";  
int a = int.Parse( stringValue );    // a sẽ mang giá trị 123  
  
float b = float.Parse( "20.7" );    // b sẽ mang giá trị 20.7  
bool c = bool.Parse( "true" );      // c sẽ mang giá trị True
```

❖ Sử dụng lớp hỗ trợ Convert

```
int a = Convert.ToInt32( "123" );  
Console.WriteLine(a);                // a = 123  
  
double b = Convert.ToInt32( 789 );  
Console.WriteLine(b);                // b = 789  
  
bool c = Convert.ToBoolean( 123 );  
Console.WriteLine(c);                // c = True  
  
bool d = Convert.ToBoolean(null);     // Tham số truyền vào là null -> sẽ output giá trị  
Console.WriteLine(d);                // d = False
```

- ❑ Sử dụng ký tự **alphabet**, **số**, **\$** hoặc gạch dưới (**_**).
Tên có phân biệt **HOA/thường**
- ❑ Không ~~bắt đầu bởi số~~, không dùng ~~từ khóa~~

** Từ khóa là các từ được sử dụng để xây dựng ra ngôn ngữ lập trình C#*

abstract	assert	boolean	break	byte	case
catch	char	class	const	continue	default
do	double	else	enum	extends	final
finally	float	for	goto	if	implements
import	instanceof	int	interface	long	native
new	package	private	protected	public	return
short	static	strictfp	super	switch	synchronized
this	throw	throws	transient	try	void
volatile	while				

TÊN BIẾN NÀO SAU ĐÂY KHÔNG HỢP LỆ

A •abc

B •1abc *Sai vì bắt đầu bởi số*

D •ab_c

E •_abc

F •\$abc

G •ab-c *Sai vì chứa dấu gạch ngang*

H •_123

I •\$123

J •if *Sai vì sử dụng từ khóa*

K •\$if

Toán tử	Diễn giải	Ví dụ
+	Phép cộng	<code>int a = 5 + 7</code>
-	Phép trừ	<code>int b = 9 - 6</code>
*	Phép nhân	<code>double c = 9.5 * 2</code>
/	Phép chia	<code>double d = 3.5 / 5</code>

❑ Toán tử số học được sử dụng để thực hiện các phép toán số học

❑ Thứ tự ưu tiên

1. Nhân và chia
2. Cộng và trừ
3. Trái sang phải

$5 + 7 * 2 - 4 / 2$

???

Hàm	Diễn giải	Ví dụ
Math.min(a, b)	Lấy số nhỏ nhất của 2 số a và b	$x = \text{Math.min}(5, 3.5) \Rightarrow x = 3.5$
Math.max(a, b)	Lấy số lớn nhất của 2 số a và b	$x = \text{Math.max}(5, 3.5) \Rightarrow x = 5$
Math.pow(a, n)	Tính a^n (a lũy thừa n)	$x = \text{Math.pow}(5, 3) \Rightarrow x = 75$
Math.sqrt(a)	Tính \sqrt{a} (căn bậc 2 của a)	$x = \text{Math.sqrt}(16) \Rightarrow x = 4$
Math.abs(a)	Lấy giá trị tuyệt đối của a	$x = \text{Math.abs}(-5) \Rightarrow x = 5$
Math.ceil(a)	Lấy số nguyên trên của a	$x = \text{Math.ceil}(3.5) \Rightarrow x = 4$
Math.floor(a)	Lấy số nguyên dưới của a	$x = \text{Math.floor}(3.5) \Rightarrow x = 3$
Math.round(a)	Làm tròn số của a	$x = \text{Math.round}(3.5) \Rightarrow x = 4$
Math.random()	Sinh số ngẫu nhiên từ 0 đến 1	$x = \text{Math.random()} \Rightarrow x = 0..1$



DEMO


1. Sinh số ngẫu nhiên từ 5 đến 12
Xuất số đó và căn bậc 2 của nó ra màn hình
2. Nhập 2 số thực a và b từ bàn phím
Tính và xuất a lũy b, giá trị nhỏ nhất của 2 số



Tổng kết bài học

Phần I: Tìm Hiểu Chương trình C#

 .Net framework

 Ngôn ngữ lập trình c#

 Chương trình c#



Phần II: Biến, kiểu, toán tử, biểu thức

 Lưu trữ dữ liệu

 Các hàm thông dụng





KẾT THÚC