

TỔNG QUAN

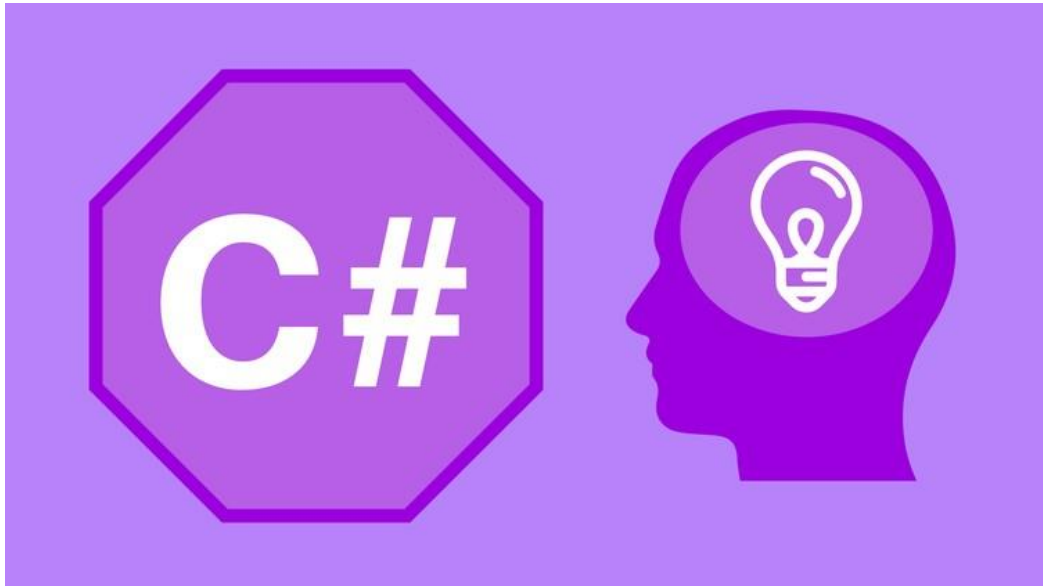
Biên soạn: **ThS. Nguyễn Thị Anh Thư**

1

NỘI DUNG

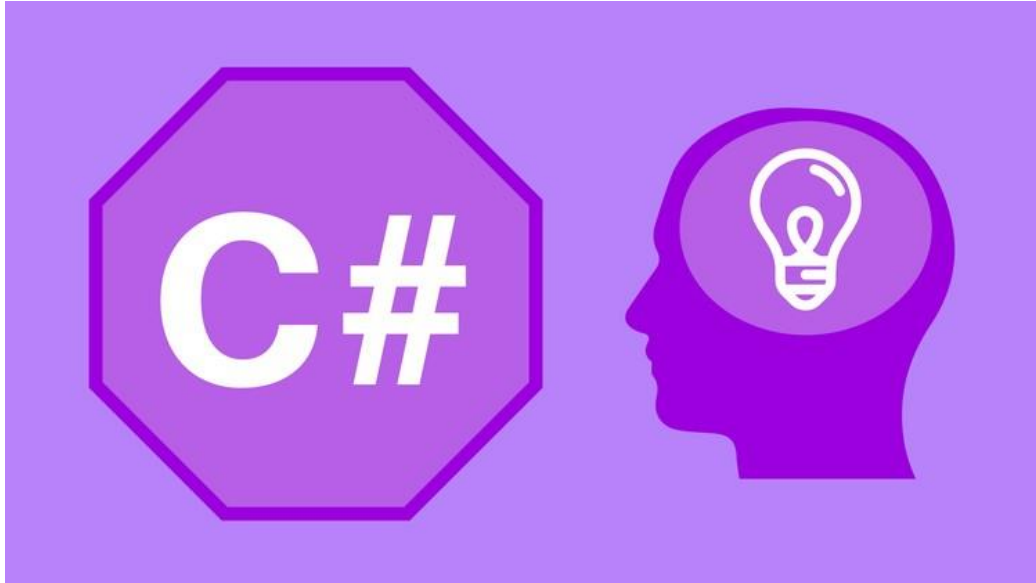
1. Giới thiệu
2. Cài đặt môi trường
3. Cấu trúc chương trình
4. Kiểu dữ liệu
5. Chuyển đổi kiểu
6. Biến
7. Hằng
8. Toán tử
9. Lệnh If, Else, Switch
10. Vòng lặp

1. GIỚI THIỆU



- C# là một ngôn ngữ lập trình đơn giản, hiện đại, mục đích tổng quát và hướng đối tượng.
- Được phát triển bởi Microsoft.
- **Ngôn ngữ lập trình C# dựa trên các ngôn ngữ lập trình C và C++.**

1. GIỚI THIỆU



- **C# là ngôn ngữ lập trình chuyên nghiệp được sử dụng rộng rãi:**
 - Là một ngôn ngữ lập trình hiện đại, mục đích tổng quát.
 - Hướng đối tượng.
 - Dễ dàng để học.
 - Một ngôn ngữ được cấu trúc.
 - Tạo các chương trình hiệu quả.
 - Có thể được biên dịch trên nhiều nền tảng máy tính khác nhau.
 - Là một phần của *.Net framework*.

2. CÀI ĐẶT MÔI TRƯỜNG

- **C# là một phần của .Net Framework và được sử dụng để viết các ứng dụng .Net.**
- **.NET Framework là một nền tảng lập trình và cũng là một nền tảng thực thi ứng dụng chủ yếu trên hệ điều hành Microsoft Windows được phát triển bởi Microsoft.**
- Trong thế giới .NET:
 - Môi trường chạy ứng dụng được đặt tên là **Common Language Runtime (CLR)**. Khi người dùng chạy một ứng dụng, code cho ứng dụng đó thực sự được biên dịch thành ngôn ngữ máy tại thời gian chạy và sau đó được thực thi.
 - CLR cũng cung cấp một số dịch vụ khác, chẳng hạn như *quản lý bộ nhớ và các bộ vi xử lý, xử lý các trường hợp ngoại lệ của chương trình và quản lý bảo mật.*

2. CÀI ĐẶT MÔI TRƯỜNG

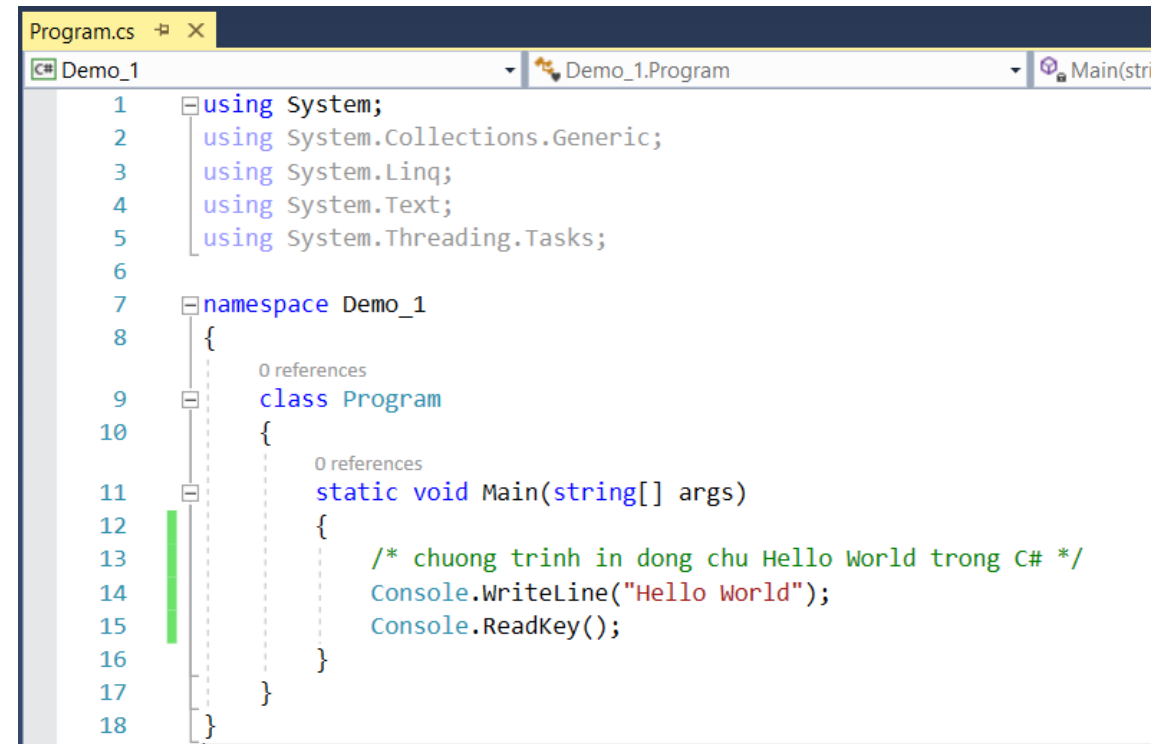
- Phiên bản Windows hiện nay đã có sẵn .NET Framework trong cài đặt mặc định.
- Microsoft cung cấp các công cụ phát triển sau cho lập trình C#:
 - Visual Studio (VS)
 - Visual C# Express (VCE)
 - Visual Web Developer



3. CẤU TRÚC CHƯƠNG TRÌNH

▪ Biên dịch và thực thi chương trình C#:

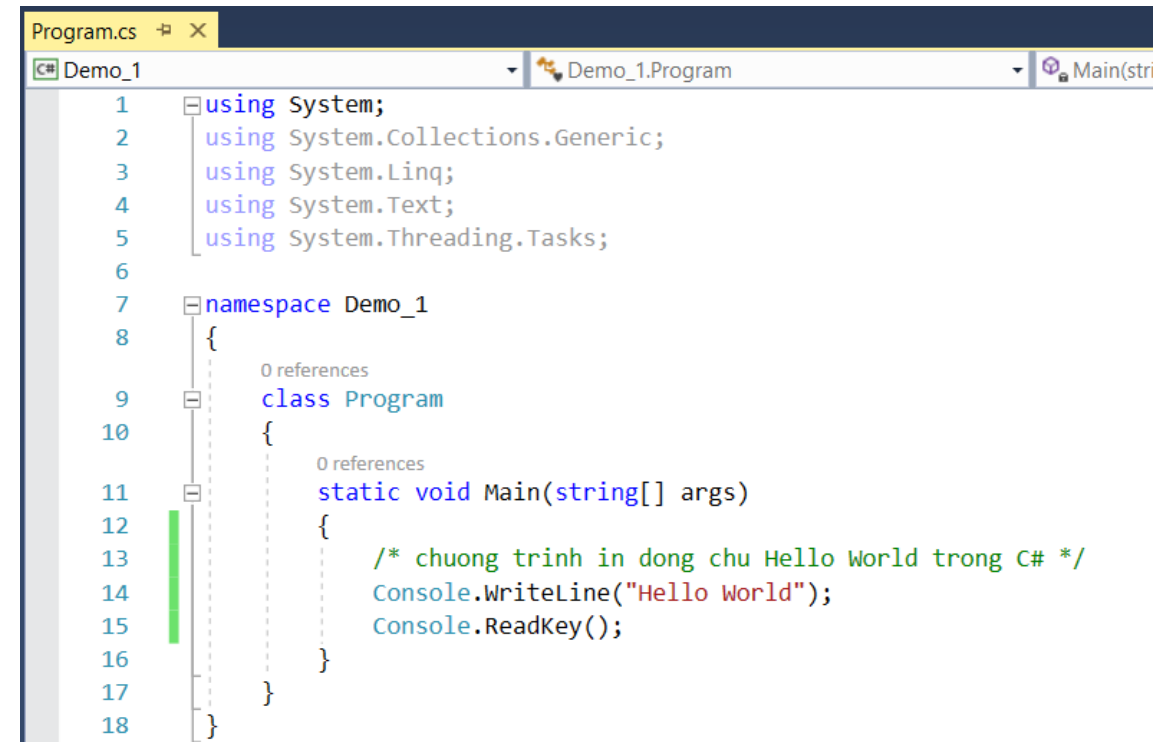
1. Bắt đầu *Visual Studio*.
2. Trên thanh menu, chọn *File* → *New* → *Project*.
3. Chọn Visual C# từ các *Template* và sau đó chọn *Windows*.
4. Chọn *Console Application*.
5. *Xác định tên cho project* và nhấn nút OK.
6. Điều này tạo New Project trong Solution Explorer.
7. Viết code trong Code Editor.
8. Nhấn nút *Run* hoặc nhấn phím *F5* để thực thi project. *Một cửa sổ Command Prompt xuất hiện mà chứa dòng Hello World.*



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Demo_1
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            /* chương trình in dòng chu Hello World trong C# */
16            Console.WriteLine("Hello World");
17            Console.ReadKey();
18        }
19    }
```

3. CẤU TRÚC CHƯƠNG TRÌNH

- Một chương trình C# gồm các phần sau:
 - Dòng đầu tiên **using System** (*một chương trình có nhiều lệnh using*)
 - Khai báo **Namespace** (*một namespace là một tập hợp các lớp*)
 - Một **class** (*chứa các định nghĩa dữ liệu và phương thức*)
 - Phương thức của lớp (*định nghĩa hành vi của lớp*)
 - Thuộc tính của lớp (*là các thành phần dữ liệu*)
 - Một phương thức **Main**
 - Lệnh và biểu thức
 - Comment



```
Program.cs
C# Demo_1 Demo_1.Program Main(strin
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Demo_1
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            /* chương trình in dòng chu Hello World trong C# */
16            Console.WriteLine("Hello World");
17            Console.ReadKey();
18        }
19    }
20 }
```


3. CẤU TRÚC CHƯƠNG TRÌNH

▪ Lưu ý:

- C# là phân biệt kiểu chữ (case sensitive).
- Tất cả lệnh và biểu thức phải kết thúc với một dấu chấm phẩy (;).
- Sự thực thi chương trình bắt đầu tại phương thức **Main**.
- Không giống Java, tên file chương trình có thể khác tên lớp.

4. KIỂU DỮ LIỆU

- **Các biến trong C# được phân chia thành các kiểu sau:**
 - **Kiểu giá trị (Value type)**: Các biến kiểu giá trị có thể được gán một giá trị một cách trực tiếp.
 - VD: `int a=100;`
 - **Kiểu tham chiếu (Reference type)**: Kiểu tham chiếu không chứa dữ liệu thực sự được lưu giữ trong một biến, nhưng chúng chứa một tham chiếu tới các biến.
 - VD: Các kiểu tham chiếu có sẵn trong C# là: **object**, **dynamic** và **string**.
 - `object obj;`
 - `obj = 100;`
 - **Kiểu con trỏ (Pointer type)**: Các biến kiểu con trỏ lưu giữ địa chỉ bộ nhớ của kiểu khác. Các con trỏ trong C# có khả năng như con trỏ trong C hoặc C++.
 - VD: `char* cptr;`

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
short	Kiểu signed integer (16 bit)	-32,768 tới 32,767	0
uint	Kiểu unsigned integer (32 bit)	0 tới 4,294,967,295	0
ulong	Kiểu unsigned integer (64 bit)	0 tới 18,446,744,073,709,551,615	0
ushort	Kiểu unsigned integer (16 bit)	0 tới 65,535	0

5. CHUYỂN ĐỔI KIỂU

- Chuyển đổi kiểu dữ liệu trong C# (*ép kiểu*) là biến đổi một kiểu dữ liệu này thành kiểu dữ liệu khác.
- Trong C#, có 2 cách ép kiểu:
 - **Chuyển đổi kiểu ngầm định (implicit)** - Việc chuyển đổi này được thực hiện bởi C# theo một phương thức an toàn kiểu (type-safe). Ví dụ: việc chuyển đổi từ các lớp kế thừa thành các lớp cơ sở.
 - **Chuyển đổi kiểu tường minh (explicit)** - Việc chuyển đổi này được thực hiện một cách rõ ràng bởi người dùng bằng việc sử dụng các hàm được định nghĩa trước.

5. CHUYỂN ĐỔI KIỂU

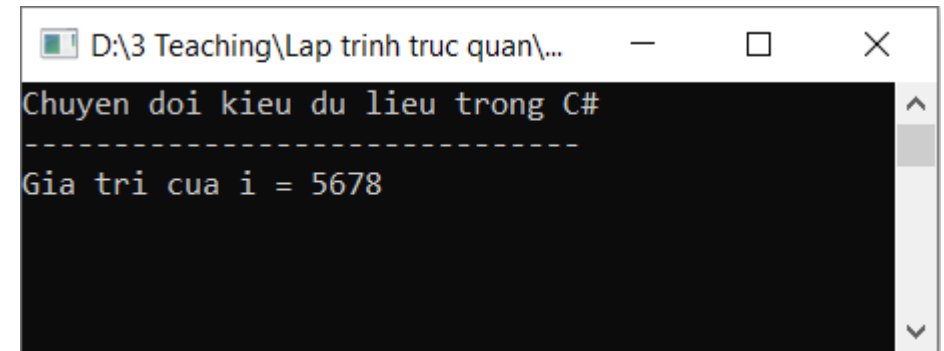
▪ Ví dụ:

0 references

```
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();
}
```

A screenshot of a Windows console window. The title bar shows the path "D:\3 Teaching\Lap trinh truc quan\...". The console output is as follows:
Chuyen doi kieu du lieu trong C#

Gia tri cua i = 5678
The text is displayed in a monospaced font on a black background.

STT	Phương thức & Miêu tả
1	ToBoolean Chuyển đổi một kiểu thành một giá trị Boolean, nếu có thể
2	ToByte Chuyển đổi một kiểu thành một byte
3	ToChar Chuyển đổi một kiểu thành một Unicode character, nếu có thể
4	ToDateTime Chuyển đổi một kiểu (kiểu integer hoặc string) thành các cấu trúc date-time
5	ToDecimal Chuyển đổi một kiểu số thực hoặc số nguyên thành một kiểu thập phân
6	ToDouble Chuyển đổi một kiểu thành một kiểu double
7	ToInt16 Chuyển đổi một kiểu thành một 16-bit integer

8	ToInt32 Chuyển đổi một kiểu thành một 32-bit integer
9	ToInt64 Chuyển đổi một kiểu thành một 64-bit integer
10	ToSbyte Chuyển đổi một kiểu thành một kiểu signed byte
11	ToSingle Chuyển đổi một kiểu thành một số small floating point
12	ToString Chuyển đổi một kiểu thành một string
13	ToType Chuyển đổi một kiểu thành một kiểu đã xác định
14	ToUInt16 Chuyển đổi một kiểu thành một kiểu unsigned int
15	ToUInt32 Chuyển đổi một kiểu thành một kiểu unsigned long
16	ToUInt64 Chuyển đổi một kiểu thành một unsigned big integer

5. CHUYỂN ĐỔI KIỂU

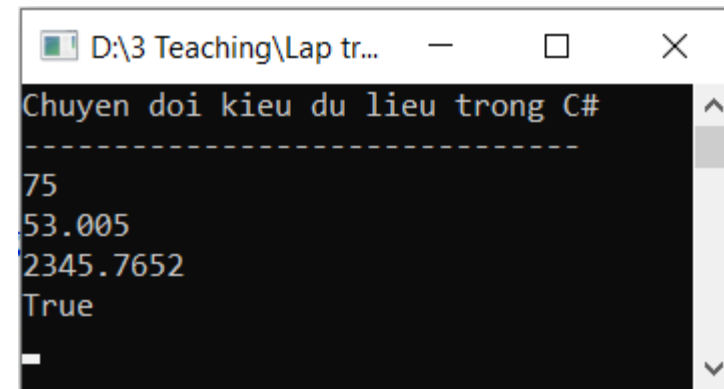
▪ Ví dụ:

0 references

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

    int i = 75;
    float f = 53.005f;
    double d = 2345.7652;
    bool b = true;
    //su dung phuong thuc ToString()
    Console.WriteLine(i.ToString());
    Console.WriteLine(f.ToString());
    Console.WriteLine(d.ToString());
    Console.WriteLine(b.ToString());

    Console.ReadKey();
}
```



6. BIẾN

- **Một biến là một tên được cung cấp cho khu vực lưu giữ.**
- **Mỗi biến trong C# có một kiểu cụ thể quyết định:**
 - Kích cỡ và cách bố trí bộ nhớ của biến đó.
 - Cách dãy giá trị có thể được lưu giữ trong bộ nhớ đó.
 - Cách tập hợp các hoạt động có thể được áp dụng tới biến đó.

Kiểu dữ liệu	Ví dụ
Kiểu số nguyên	sbyte, byte, short, ushort, int, uint, long, ulong, và char
Kiểu số thực dấu chấm động	float và double
Kiểu thập phân	decimal
Kiểu Boolean	true hoặc false
Kiểu Nullable	Kiểu dữ liệu Nullable

6. BIẾN

- Cú pháp để **định nghĩa biến** trong C# là:

<kiểu_dữ_liệu> <danh_sách_biến>;

- Ở đây:
 - **kiểu_dữ_liệu** phải là một kiểu dữ liệu hợp lệ trong C#.
 - **danh_sách_biến** có thể chứa một hoặc nhiều tên định danh được phân biệt bởi dấu phẩy.
- **Biến được khởi tạo** (*được gán một giá trị*) với một dấu bằng được theo sau bởi một biểu thức hằng.

tên_biến = giá_trị;

6. BIẾN

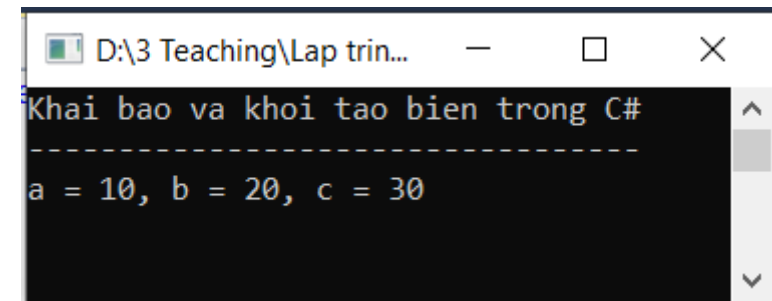
▪ Ví dụ:

0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Khai bao va khoi tao bien trong C#");
    Console.WriteLine("-----");

    //khai bao bien
    short a;
    int b;
    double c;

    /* khoi tao bien */
    a = 10;
    b = 20;
    c = a + b;
    Console.WriteLine("a = {0}, b = {1}, c = {2}", a, b, c);
    Console.ReadLine();
}
```

A screenshot of a Windows console window titled "D:\3 Teaching\Lap trin...". The window has a black background with white text. The output of the program is displayed: "Khai bao va khoi tao bien trong C#" followed by a dashed line "-----" and then "a = 10, b = 20, c = 30". The window includes standard Windows window controls (minimize, maximize, close) in the title bar.

7. HẰNG

- Constant liên quan tới các giá trị cố định mà chương trình không thể thay đổi trong khi thực thi.
- Constant có thể là bất kỳ kiểu dữ liệu cơ bản nào trong C#.
- Ngoài ra, constant được đối xử giống như các biến thông thường, ngoại trừ việc giá trị của chúng là không thể thay đổi sau khi định nghĩa.
- Hằng trong C# được định nghĩa bởi sử dụng từ khóa **const**.
- Cú pháp để định nghĩa một hằng là:

const <kiểu_dữ_liệu> <tên_hằng> = giá_trị;

7. HẰNG

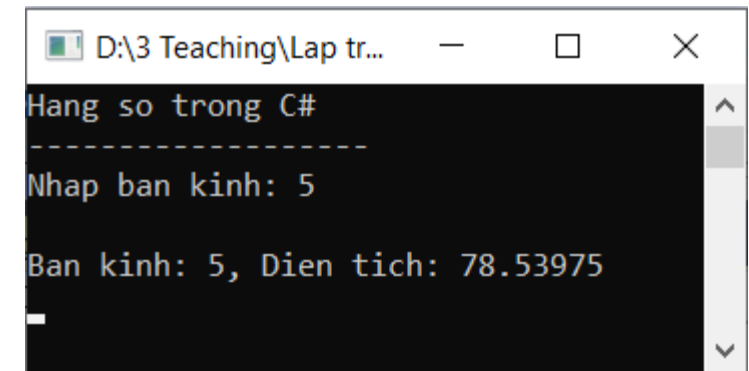
▪ Ví dụ:

0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Hang so trong C#");
    Console.WriteLine("-----");

    //khai bao hang so pi
    const double pi = 3.14159;

    double r;
    Console.Write("Nhap ban kinh: ");
    r = Convert.ToDouble(Console.ReadLine());
    double dien_tich = pi * r * r;
    Console.WriteLine("\nBan kinh: {0}, Dien tich: {1}", r, dien_tich);
    Console.ReadLine();
}
```



The screenshot shows a console window titled "D:\3 Teaching\Lap tr...". The output of the program is as follows:

```
Hang so trong C#
-----
Nhap ban kinh: 5

Ban kinh: 5, Dien tich: 78.53975
```

8. TOÁN TỬ

- **Một toán tử là một biểu tượng nói cho compiler thực hiện các thao tác toán học và logic cụ thể.**
- C# cung cấp nhiều toán tử có sẵn, đó là:
 - Toán tử số học
 - Toán tử quan hệ
 - Toán tử logic
 - Toán tử so sánh bit
 - Toán tử gán
 - Toán tử hỗn hợp

8. TOÁN TỬ

▪ Toán tử số học

Toán tử	Miêu tả
+	Thêm hai toán hạng
-	Trừ giá trị toán hạng hai từ toán hạng đầu
*	Nhân hai toán hạng
/	Chia lấy phần nguyên hai toán hạng
%	Chia lấy phần dư
++	Lượng gia giá trị toán hạng thêm 1 đơn vị
--	Lượng giảm giá trị toán hạng một đơn vị

8. TOÁN TỬ

▪ Toán tử quan hệ

Toán tử	Miêu tả
==	Kiểm tra nếu 2 toán hạng bằng nhau hay không. Nếu bằng thì điều kiện là true.
!=	Kiểm tra 2 toán hạng có giá trị khác nhau hay không. Nếu không bằng thì điều kiện là true.
>	Kiểm tra nếu toán hạng bên trái có giá trị lớn hơn toán hạng bên phải hay không. Nếu lớn hơn thì điều kiện là true.
<	Kiểm tra nếu toán hạng bên trái nhỏ hơn toán hạng bên phải hay không. Nếu nhỏ hơn thì là true.
>=	Kiểm tra nếu toán hạng bên trái có giá trị lớn hơn hoặc bằng giá trị của toán hạng bên phải hay không. Nếu đúng là true.
<=	Kiểm tra nếu toán hạng bên trái có giá trị nhỏ hơn hoặc bằng toán hạng bên phải hay không. Nếu đúng là true.

8. TOÁN TỬ

▪ Toán tử logic

Toán tử	Miêu tả
&&	Được gọi là toán tử logic AND (và). Nếu cả hai toán tử đều có giá trị khác 0 thì điều kiện trở lên true.
	Được gọi là toán tử logic OR (hoặc). Nếu một trong hai toán tử khác 0, thì điều kiện là true.
!	Được gọi là toán tử NOT (phủ định). Sử dụng để đảo ngược lại trạng thái logic của toán hạng đó. Nếu điều kiện toán hạng là true thì phủ định nó sẽ là false.

8. TOÁN TỬ

▪ Toán tử gán

Toán tử	Miêu tả
=	Toán tử gán đơn giản. Gán giá trị toán hạng bên phải cho toán hạng trái.
+=	Thêm giá trị toán hạng phải tới toán hạng trái và gán giá trị đó cho toán hạng trái.
-=	Trừ đi giá trị toán hạng phải từ toán hạng trái và gán giá trị này cho toán hạng trái.
*=	Nhân giá trị toán hạng phải với toán hạng trái và gán giá trị này cho toán hạng trái.
/=	Chia toán hạng trái cho toán hạng phải và gán giá trị này cho toán hạng trái.
%=	Lấy phần dư của phép chia toán hạng trái cho toán hạng phải và gán cho toán hạng trái.
<<=	Dịch trái toán hạng trái sang số vị trí là giá trị toán hạng phải.
>>=	Dịch phải toán hạng trái sang số vị trí là giá trị toán hạng phải.
&=	Phép AND bit
^=	Phép OR loại trừ bit
=	Phép OR bit.

9. LỆNH IF, ELSE, SWITCH

- C# cung cấp các loại điều khiển luồng sau:

Lệnh	Miêu tả
Lệnh if trong C#	Một lệnh if bao gồm một biểu thức logic theo sau bởi một hoặc nhiều lệnh khác.
Lệnh if...else trong C#	Một lệnh if có thể theo sau bởi một lệnh else (tùy ý: có hoặc không), mà có thể được thực hiện khi biểu thức logic có giá trị false.
Lồng các lệnh if trong C#	Bạn có thể sử dụng lệnh if hoặc lệnh else if bên trong lệnh if hoặc else if khác
Lệnh switch trong C#	Lệnh switch cho phép kiểm tra điều kiện của một biến trước khi thực thi các lệnh
Lồng các lệnh switch trong C#	Bạn có thể sử dụng một lệnh switch bên trong một lệnh switch khác

9. LỆNH IF, ELSE, SWITCH

- **Lệnh if trong C#**
- Nếu biểu thức logic được ước lượng là **true**, thì sau đó khối code bên trong lệnh if sẽ được thực thi. Nếu biểu thức logic được ước lượng là **false**, thì khi đó, lệnh ngay sau lệnh if sẽ được thực thi.

```
if(biểu_thức)
{
    /* các lệnh được thực thi nếu biểu_thức là true */
}
```

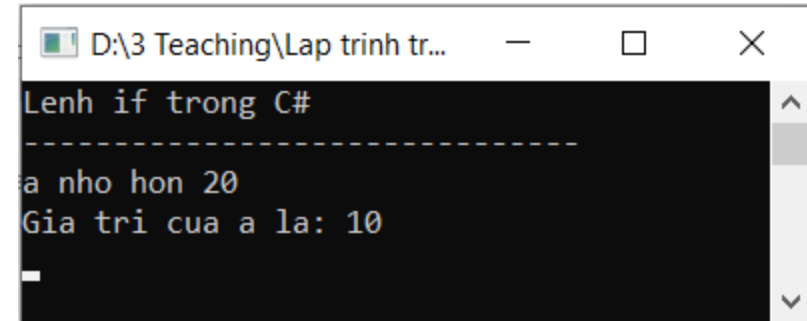
9. LỆNH IF, ELSE, SWITCH

- **Lệnh if trong C#**
- Ví dụ:

0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Lệnh if trong C#");
    Console.WriteLine("-----");

    /* phân định nghĩa biến cục bộ */
    int a = 10;
    /* kiểm tra điều kiện của biểu thức trong lệnh if*/
    if (a < 20)
    {
        /* nếu điều kiện là true thì sẽ in ra dòng sau: */
        Console.WriteLine("a nhỏ hơn 20");
    }
    Console.WriteLine("Giá trị của a là: {0}", a);
    Console.ReadLine();
}
```



The screenshot shows a console window titled "D:\3 Teaching\Lap trình tr...". The output of the program is displayed as follows:

```
Lệnh if trong C#
-----
a nhỏ hơn 20
Giá trị của a là: 10
```

9. LỆNH IF, ELSE, SWITCH

- **Lệnh if...else trong C#**
- Nếu biểu thức logic được ước lượng là **true**, thì khi đó **khối if** sẽ được thực thi, nếu không thì **khối else** sẽ được thực thi.

```
if(biểu_thức)
{
    /* các lệnh được thực thi nếu biểu thức là true */
}
else
{
    /* các lệnh được thực thi nếu biểu thức là false */
}
```

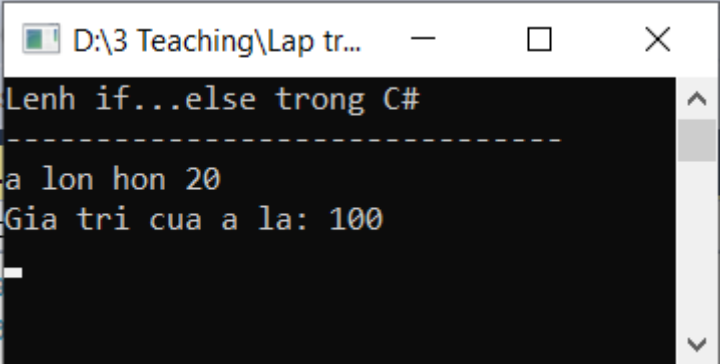
9. LỆNH IF, ELSE, SWITCH

- Lệnh if...else trong C#
- Ví dụ:

```
0 references
static void Main(string[] args)
{
    Console.WriteLine("Lệnh if...else trong C#");
    Console.WriteLine("-----");

    /* phan dinh nghia bien cuc bo */
    int a = 100;

    /* kiem tra dieu kien */
    if (a > 20)
    {
        /* neu dieu kien la true thi in dong sau: */
        Console.WriteLine("a lon hon 20");
    }
    else
    {
        /* neu dieu kien la false thi in dong sau: */
        Console.WriteLine("a khong lon hon 20");
    }
    Console.WriteLine("Gia tri cua a la: {0}", a);
    Console.ReadLine();
}
```



The screenshot shows a console window titled "D:\3 Teaching\Lap tr...". The output text is as follows:

```
Lệnh if...else trong C#
-----
a lon hon 20
Gia tri cua a la: 100
```

9. LỆNH IF, ELSE, SWITCH

- Lồng các lệnh if trong C#

```
if( biểu_thức 1)
{
    /* các lệnh được thực thi nếu biểu thức 1 là true */
    if(biểu_thức 2)
    {
        /* các lệnh được thực thi nếu biểu thức 2 là true */
    }
}
```


9. LỆNH IF, ELSE, SWITCH

▪ Lòng các lệnh if trong C#

▪ Ví dụ:

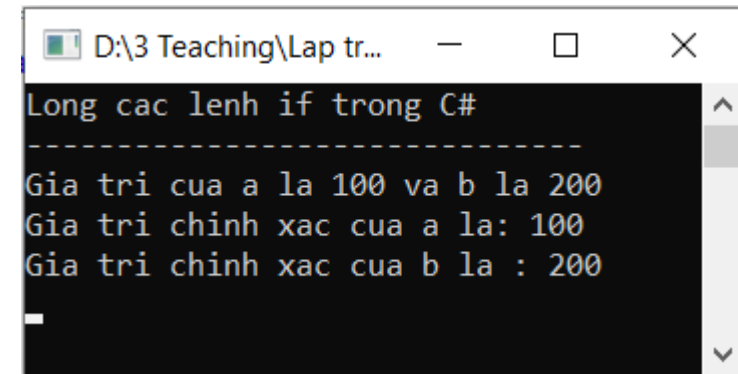
0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Long cac lenh if trong C#");
    Console.WriteLine("-----");

    /* phan dinh nghĩa biến cục bộ */
    int a = 100;
    int b = 200;

    /* kiểm tra điều kiện */
    if (a == 100)
    {
        /* nếu điều kiện là true, tiếp tục kiểm tra: */
        if (b == 200)
        {
            /* nếu điều kiện là true thì in dòng sau: */
            Console.WriteLine("Gia tri cua a la 100 va b la 200");
        }
    }

    Console.WriteLine("Gia tri chinh xac cua a la: {0}", a);
    Console.WriteLine("Gia tri chinh xac cua b la : {0}", b);
    Console.ReadLine();
}
```



The screenshot shows a console window titled "D:\3 Teaching\Lap tr...". The output text is as follows:

```
Long cac lenh if trong C#
-----
Gia tri cua a la 100 va b la 200
Gia tri chinh xac cua a la: 100
Gia tri chinh xac cua b la : 200
```

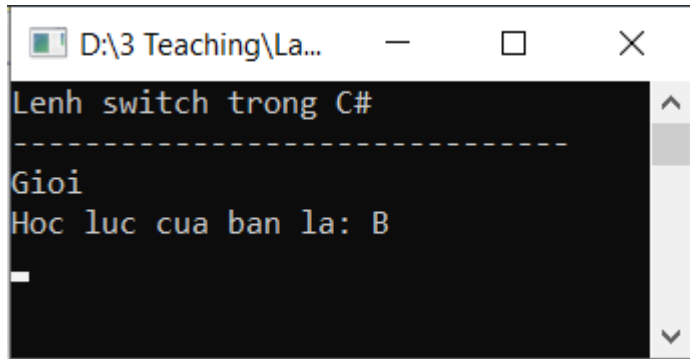
9. LỆNH IF, ELSE, SWITCH

- **Lệnh switch trong C#**
- Một lệnh **switch** trong C# cho một biến được kiểm tra một cách bình đẳng trong danh sách các giá trị. Mỗi giá trị được gọi là một **case - trường hợp** và biến được chuyển tới được kiểm tra cho mỗi trường hợp switch.

```
switch(biểu_thức) {  
    case biểu_thức_hằng :  
        các_lệnh_cần_thực_thi;  
        break; /* tùy ý */  
    case biểu_thức_hằng :  
        các_lệnh_cần_thực_thi;  
        break; /* tùy ý */  
  
    /* số lượng lệnh case là tùy theo bạn */  
    default : /* tùy ý */  
        các_lệnh_cần_thực_thi;  
}
```

9. LỆNH IF, ELSE, SWITCH

- **Lệnh switch trong C#**
- Ví dụ:



```
D:\3 Teaching\La...
Lenh switch trong C#
-----
Gioi
Hoc luc cua ban la: B
_
```

0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Lenh switch trong C#");
    Console.WriteLine("-----");

    /* phan dinh nghia bien cuc bo */
    char grade = 'B';

    switch (grade)
    {
        case 'A':
            Console.WriteLine("Xuat sac!");
            break;
        case 'B':
        case 'C':
            Console.WriteLine("Gioi");
            break;
        case 'D':
            Console.WriteLine("Trung binh");
            break;
        case 'F':
            Console.WriteLine("Hoc lai");
            break;
        default:
            Console.WriteLine("Gia tri khong hop le");
            break;
    }
    Console.WriteLine("Hoc luc cua ban la: {0}", grade);
    Console.ReadLine();
}
```

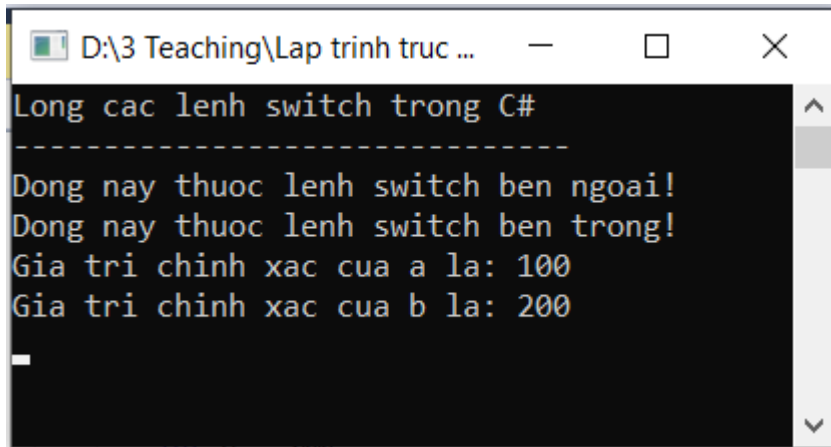
9. LỆNH IF, ELSE, SWITCH

- **Lồng các lệnh switch trong C#**
- Nó là có thể để có một lệnh switch như là một phần của dãy lệnh trong một lệnh switch ở vòng ngoài. Ngay cả khi hằng số case trong và ngoài lệnh switch chứa các giá trị bình thường, sẽ không có sự xung đột diễn ra ở đây.

```
switch(biểu_thức)
{
    case 'A':
        printf("A này là của lệnh switch bên ngoài!" );
        switch(biểu_thức)
        {
            case 'A':
                printf("A này là của lệnh switch bên trong!" );
                break;
            case 'B': /* B của lệnh switch bên trong */
            }
        break;
    case 'B': /* của lệnh switch bên ngoài */
}
```

9. LỆNH IF, ELSE, SWITCH

- Lồng các lệnh switch trong C#
- Ví dụ:



```
Long cac lenh switch trong C#
-----
Dong nay thuoc lenh switch ben ngoai!
Dong nay thuoc lenh switch ben trong!
Gia tri chinh xac cua a la: 100
Gia tri chinh xac cua b la: 200
-
```

```
0 references
static void Main(string[] args)
{
    Console.WriteLine("Long cac lenh switch trong C#");
    Console.WriteLine("-----");

    int a = 100;
    int b = 200;

    switch (a)
    {
        case 100:
            Console.WriteLine("Dong nay thuoc lenh switch ben ngoai!");
            switch (b)
            {
                case 200:
                    Console.WriteLine("Dong nay thuoc lenh switch ben trong!");
                    break;
            }
            break;
    }

    Console.WriteLine("Gia tri chinh xac cua a la: {0}", a);
    Console.WriteLine("Gia tri chinh xac cua b la: {0}", b);
    Console.ReadLine();
}
```

10. VÒNG LẶP

- **C# hỗ trợ những lệnh điều khiển sau đây:**

Kiểu vòng lặp	Miêu tả
Vòng lặp while trong C#	Lặp lại một hoặc một nhóm các lệnh trong khi điều kiện đã cho là đúng. Nó kiểm tra điều kiện trước khi thực hiện thân vòng lặp.
Vòng lặp for trong C#	Thực thi một dãy các lệnh nhiều lần và tóm tắt đoạn code mà quản lý biến vòng lặp.
Vòng lặp do...while trong C#	Giống lệnh while, ngoại trừ ở điểm là nó kiểm tra điều kiện ở cuối thân vòng lặp.
Lồng các vòng lặp trong C#	Bạn có thể sử dụng một hoặc nhiều vòng lặp trong các vòng lặp while, for hoặc do..while khác.

10. VÒNG LẶP

- **Vòng lặp while trong C#**
- Một vòng lặp **while** trong C# thực hiện lặp đi lặp lại một lệnh mục tiêu đến khi nào điều kiện đã cho còn là đúng.

```
while(điều_kiện)
{
    statement - các lệnh cần thực thi
}
```

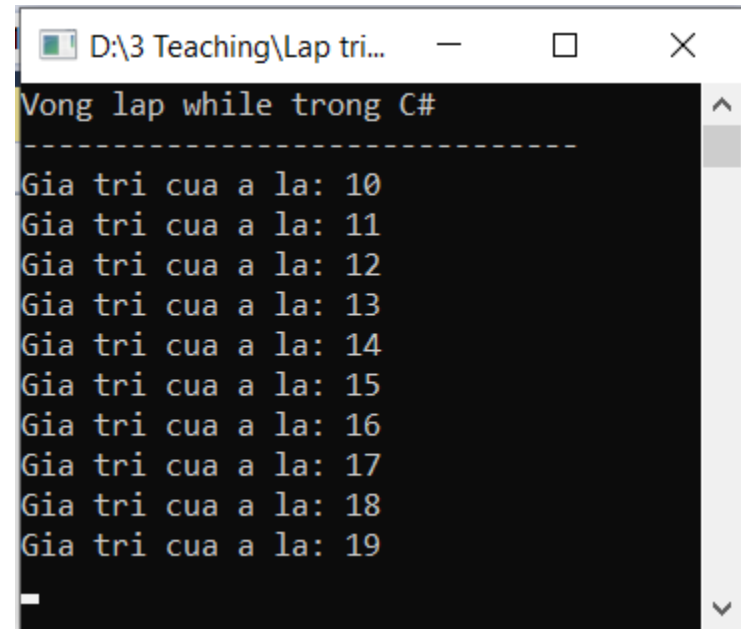
10. VÒNG LẶP

- Vòng lặp while trong C#
- Ví dụ:

```
0 references
static void Main(string[] args)
{
    Console.WriteLine("Vong lap while trong C#");
    Console.WriteLine("-----");

    /* phan dinh nghia bien cuc bo */
    int a = 10;

    /* su thuc thi cua vong lap while */
    while (a < 20)
    {
        Console.WriteLine("Gia tri cua a la: {0}", a);
        a++;
    }
    Console.ReadLine();
}
```



```
D:\3 Teaching\Lap tri...
Vong lap while trong C#
-----
Gia tri cua a la: 10
Gia tri cua a la: 11
Gia tri cua a la: 12
Gia tri cua a la: 13
Gia tri cua a la: 14
Gia tri cua a la: 15
Gia tri cua a la: 16
Gia tri cua a la: 17
Gia tri cua a la: 18
Gia tri cua a la: 19
```


10. VÒNG LẶP

- **Vòng lặp for trong C#**
- Vòng lặp **for** trong C# là một cấu trúc điều khiển lặp đi lặp lại mà cho phép bạn viết một vòng lặp một cách hiệu quả, mà cần thực hiện trong một khoảng thời gian cụ thể nào đó.

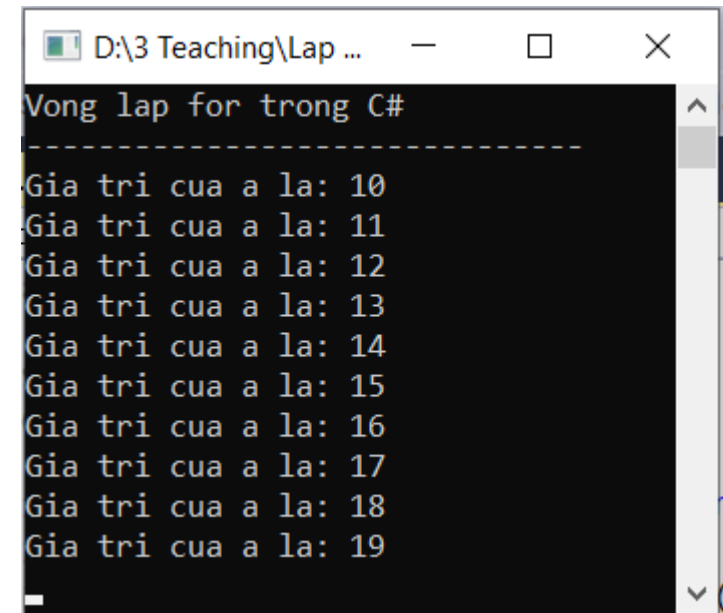
```
for ( khởi_tạo_biến_vòng_lặp; điều_kiện; tăng_giảm_biến_vòng_lặp )  
{  
    các_lệnh_được_thực_thi;  
}
```

10. VÒNG LẶP

- Vòng lặp for trong C#
- Ví dụ:

```
0 references
static void Main(string[] args)
{
    Console.WriteLine("Vong lap for trong C#");
    Console.WriteLine("-----");

    //su thuc thi vong lap for
    for (int a = 10; a < 20; a = a + 1)
    {
        Console.WriteLine("Gia tri cua a la: {0}", a);
    }
    Console.ReadLine();
}
```



```
D:\3 Teaching\Lap ...
Vong lap for trong C#
-----
Gia tri cua a la: 10
Gia tri cua a la: 11
Gia tri cua a la: 12
Gia tri cua a la: 13
Gia tri cua a la: 14
Gia tri cua a la: 15
Gia tri cua a la: 16
Gia tri cua a la: 17
Gia tri cua a la: 18
Gia tri cua a la: 19
```

10. VÒNG LẶP

- Vòng lặp **do...while** trong C#
- Không giống như các vòng lặp **for** và **while**, mà kiểm tra điều kiện vòng lặp ở ngay bước đầu tiên của vòng lặp, vòng lặp **do...while** trong Ngôn ngữ C# kiểm tra điều kiện của nó tại phần cuối của vòng lặp.
- Một vòng lặp **do...while** là tương tự như vòng lặp **while**, ngoại trừ ở điểm một vòng lặp **do...while** bảo đảm thực hiện vòng lặp ít nhất một lần.

```
do
{
    các lệnh được thực thi
}while( điều_kiện );
```

10. VÒNG LẶP

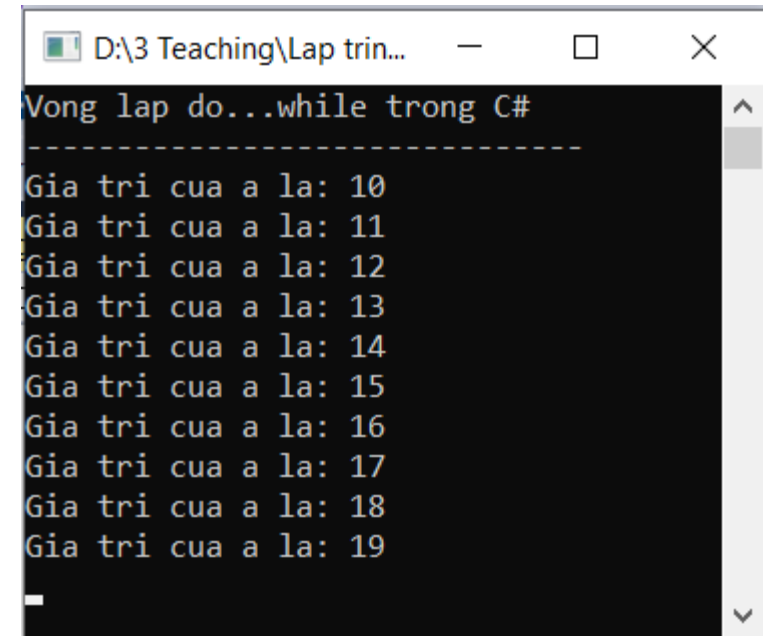
- Vòng lặp do...while trong C#
- Ví dụ:

0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Vong lap do...while trong C#");
    Console.WriteLine("-----");

    /* phan dinh nghia bien cuc bo */
    int a = 10;

    /* su thuc thi vong lap do...while */
    do
    {
        Console.WriteLine("Gia tri cua a la: {0}", a);
        a = a + 1;
    }
    while (a < 20);
    Console.ReadLine();
}
```



```
D:\3 Teaching\Lap trin...
Vong lap do...while trong C#
-----
Gia tri cua a la: 10
Gia tri cua a la: 11
Gia tri cua a la: 12
Gia tri cua a la: 13
Gia tri cua a la: 14
Gia tri cua a la: 15
Gia tri cua a la: 16
Gia tri cua a la: 17
Gia tri cua a la: 18
Gia tri cua a la: 19
```

10. VÒNG LẶP

- Lòng các vòng lặp trong C#
- C# cho phép bạn sử dụng một vòng lặp bên trong một vòng lặp.

```
for ( khởi_tạo_biến_vòng_lặp; điều_kiện; tăng_giảm_biến_vòng_lặp )  
{  
    for ( khởi_tạo_biến_vòng_lặp; điều_kiện; tăng_giảm_biến_vòng_lặp )  
    {  
        các_lệnh_được_thực_thi  
    }  
    các_lệnh_được_thực_thi  
}
```

```
do  
{  
    các_lệnh_được_thực_thi  
    do  
    {  
        các_lệnh_được_thực_thi  
    }  
    while( điều_kiện );  
}  
while( điều_kiện );
```

```
while(điều_kiện)  
{  
    while(điều_kiện)  
    {  
        các_lệnh_được_thực_thi  
    }  
    các_lệnh_được_thực_thi  
}
```

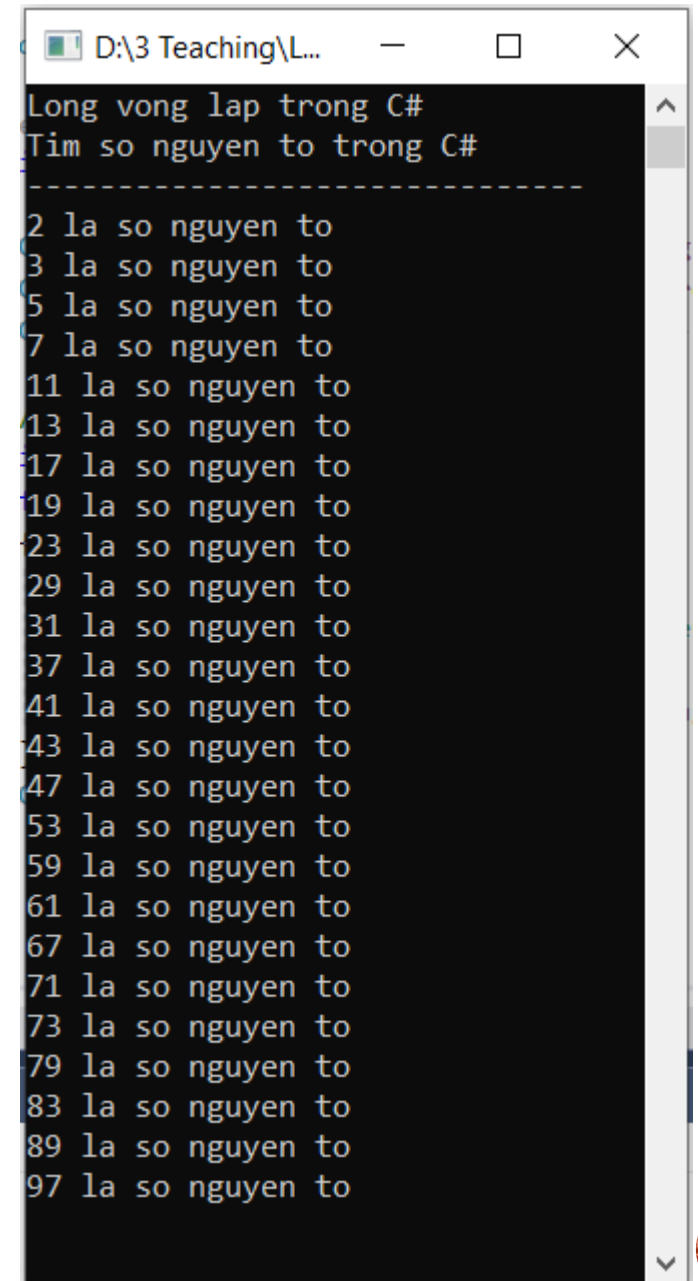
10. VÒNG LẶP

- Lồng các vòng lặp trong C#
- Ví dụ:

0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Long vong lap trong C#");
    Console.WriteLine("Tim so nguyen to trong C#");
    Console.WriteLine("-----");

    /* phan dinh nghia bien cuc bo */
    int i, j;
    for (i = 2; i < 100; i++)
    {
        for (j = 2; j <= (i / j); j++)
            if ((i % j) == 0) break; // neu tim thay uoc so thi khong phai la so nguyen to
        if (j > (i / j))
            Console.WriteLine("{0} la so nguyen to", i);
    }
    Console.ReadLine();
}
```

A screenshot of a Windows console window titled "D:\3 Teaching\L...". The window displays the output of a C# program. The first two lines are "Long vong lap trong C#" and "Tim so nguyen to trong C#", followed by a dashed line separator. Below this, the program lists prime numbers from 2 to 97, each preceded by the text "la so nguyen to". The numbers are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, and 97. The console window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
D:\3 Teaching\L...
Long vong lap trong C#
Tim so nguyen to trong C#
-----
2 la so nguyen to
3 la so nguyen to
5 la so nguyen to
7 la so nguyen to
11 la so nguyen to
13 la so nguyen to
17 la so nguyen to
19 la so nguyen to
23 la so nguyen to
29 la so nguyen to
31 la so nguyen to
37 la so nguyen to
41 la so nguyen to
43 la so nguyen to
47 la so nguyen to
53 la so nguyen to
59 la so nguyen to
61 la so nguyen to
67 la so nguyen to
71 la so nguyen to
73 la so nguyen to
79 la so nguyen to
83 la so nguyen to
89 la so nguyen to
97 la so nguyen to
```

BÀI TẬP

▪ **Viết chương trình:**

1. Nhập 3 chữ cái và hiển thị theo chiều ngược lại.
2. Viết chương trình C# để nhận Username và Password dưới dạng kiểu dữ liệu string từ người dùng. Nếu người dùng nhập sai quá 3 lần, in thông báo lỗi.
3. Giải phương trình bậc hai.
4. Viết một chương trình hiển thị một menu có các lựa chọn để tính diện tích các hình tròn, hình chữ nhật, hình tam giác tương ứng với dữ liệu đã nhập.
5. Viết một chương trình C# hiển thị menu có các lựa chọn để thực hiện các phép toán cơ bản (+, -, *, /) giữa hai số.
6. Viết chương trình C# để nhập một số và sau đó in bảng nhân của số đó.
7. Viết chương trình C# để hiển thị và tính tổng n số lẻ.
8. Viết chương trình C# để nhập một số và tìm giai thừa của số đó.



Q & A