

## Bài 2

# Biến và kiểu dữ liệu – Nhập xuất trong C#

# Mục tiêu 1

- Kiểu dữ liệu trong C#
- Biến và quy tắc đặt tên biến
- Hằng số
- Các loại toán tử
- Câu lệnh và Biểu thức
- Chuyển đổi kiểu dữ liệu
- Kiểu dữ liệu liệt kê
- Chú thích trong C#

# Kiểu dữ liệu

- Kiểu dữ liệu là 1 khái niệm quy định về loại dữ liệu cũng như độ lớn của dữ liệu mà các biến có thể chứa.
- Trong C# kiểu dữ liệu chia làm 2 nhóm

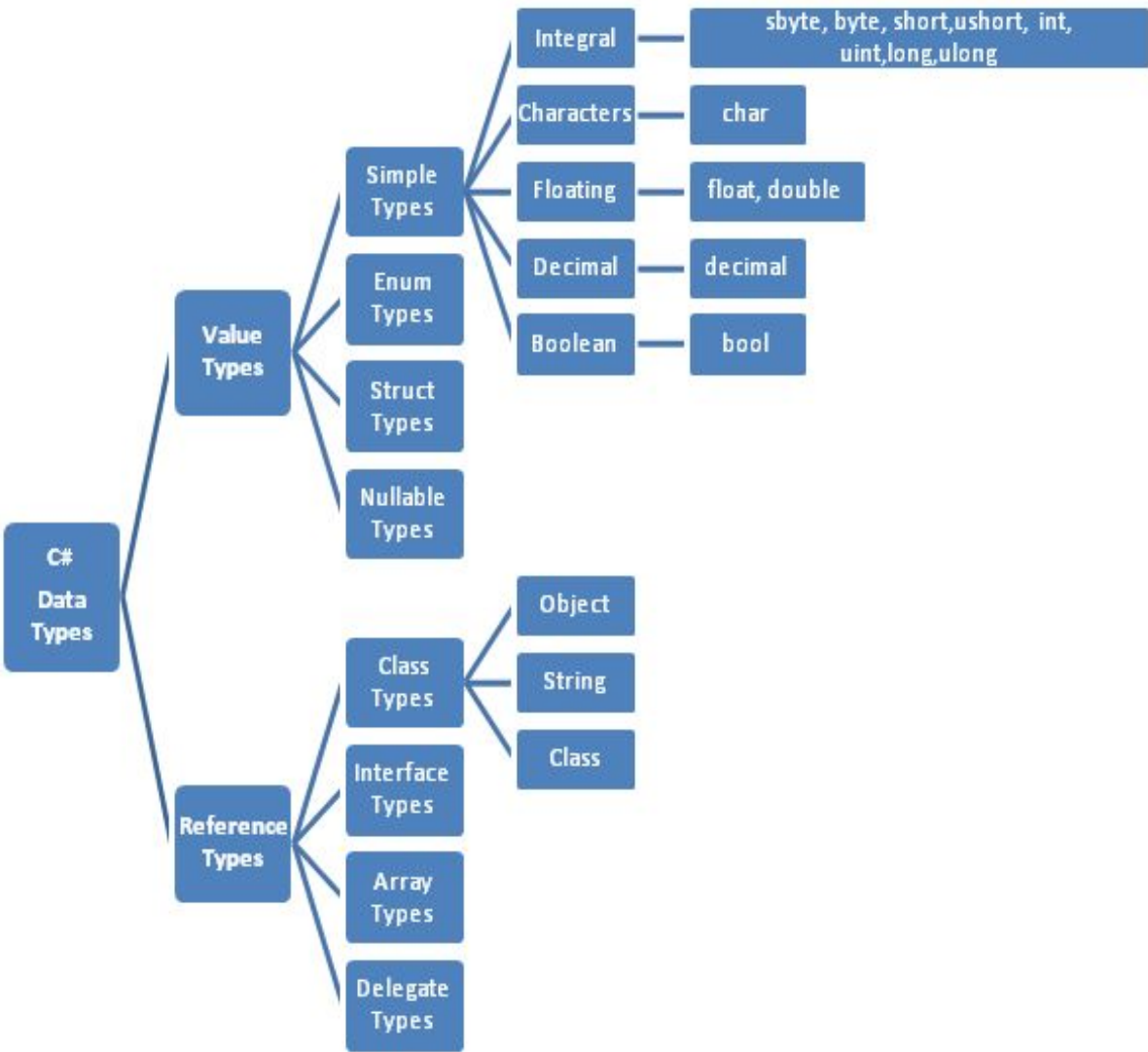


Value Types



Reference Types

# Kiểu dữ liệu



Data Type	Size	Range
byte	Unsigned 8-bit integer	0 to 255
short	Signed 16-bit integer	−32,768 to 32,767
int	Signed 32-bit integer	−2,147,483,648 to 2,147,483,647
long	Signed 64-bit integer	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	32-bit floating point with 7 digits precision	±1.5e−45 to ±3.4e38
double	64-bit floating point with 15-16 digits precision	±5.0e−324 to ±1.7e308
decimal	128-bit floating point with 28-29 digits precision	±1.0 × 10e−28 to ±7.9 × 10e28
char	Unicode 16-bit character	U+0000 to U+ffff
bool	Stores either true or false	true or false

# Biến và quy tắc đặt tên biến

- Biến là đối tượng dùng để lưu trữ tạm thời các giá trị trong quá trình xử lý tính toán.
- Một số lưu ý khi đặt tên biến
  - Phân biệt chữ hoa chữ thường
  - Không có khoảng trắng
  - Không có dấu
  - Không đặt trùng tên với từ khóa
  - Không sử dụng các ký tự đặc biệt (#,\$,%,....)
  - Không bắt đầu bằng số
  - Tên có thể bắt đầu bằng ký tự \_

# Khai báo biến 1-2

- Cú pháp

`<Data_Type> <variable_name>;`

- Ví dụ

`int age;`

`string address;`

- Gán giá trị cho biến

`age=18;`

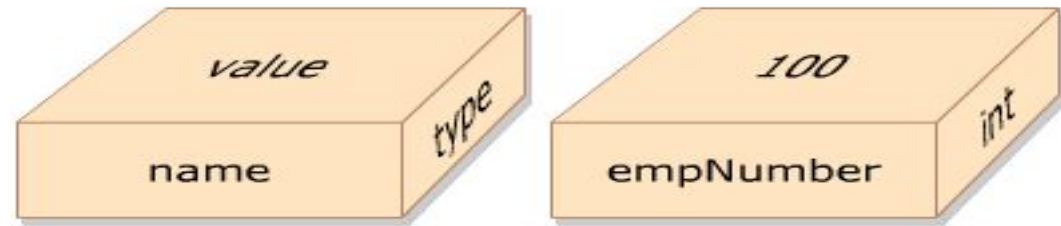
`address="Ha Noi";`

- Khai báo và khởi tạo luôn giá trị

`float score=8.5F;`

- Khai báo nhiều biến trên 1 dòng

`int a,b,c;`



# Khai báo biến 2-2

- Khai báo biến ngầm định: là cách khai báo mà không cần chỉ ra kiểu dữ liệu của biến, tuy nhiên với cách khai báo này các bạn phải khởi tạo giá trị ngay lúc khai báo

- **Cú pháp**

```
var variable_name=value;
```

- **Ví dụ**

```
var name="Xin chào";
```

-> biến name có kiểu chuỗi

# Hằng số

- Trong lập trình, hằng số là một biến được khởi tạo giá trị ngay khi khai báo và giá trị đó không được thay đổi trong chương trình.

- Khai báo Hằng

`const <Data_Type> Constant_Name=value;`

- Trong đó **const** là từ khóa khai báo hằng số
- Ví dụ:

`const double PI=3.14;`

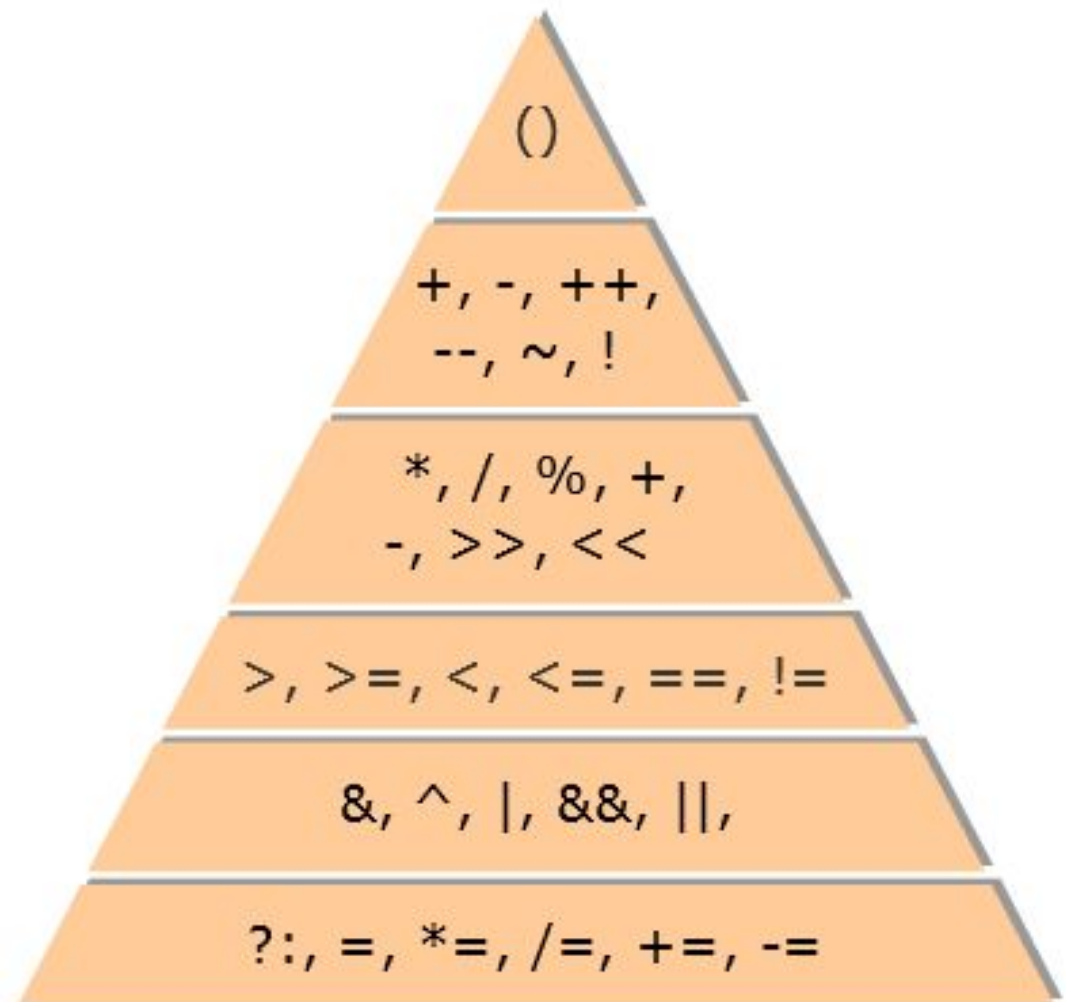


# Các ký tự escape sequence

Escape Sequence	UTF-16 character	Description
\'	\u0027	Single quotation mark
\"	\u0022	Double quotation mark
\\	\u005c	Backslash
\0	\u0000	allow to enter the character with code 0
\a	\u0007	Bell (alert)
\b	\u0008	Backspace
\f	\u000c	Formfeed (next page)
\n	\u000a	New line (next line)
\r	\u000d	Carriage return
\t	\u0009	Horizontal tab
\v	\u000b	Vertical tab

# Các loại toán tử

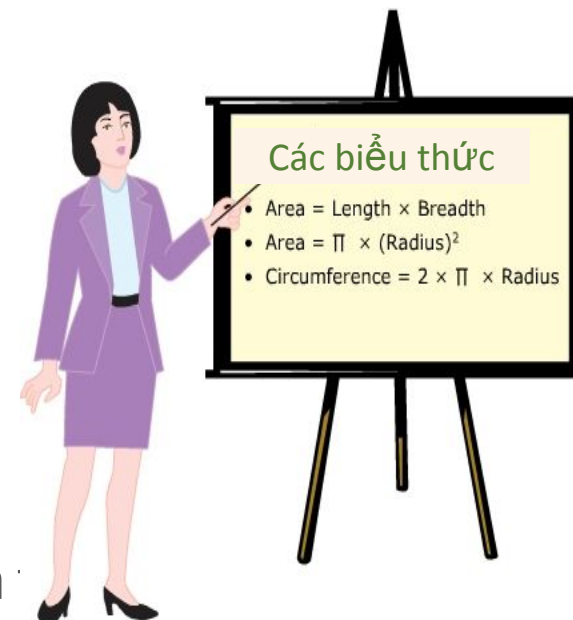
- Toán tử số học
- Toán tử so sánh
- Toán tử logic
- Toán tử gán
- Toán tử tăng/giảm
- Toán tử nối chuỗi
- Toán tử điều kiện



# Câu lệnh và biểu thức

- Các câu lệnh là một nhóm các biến, toán tử, từ khóa của C# kết hợp với nhau để thực hiện một công việc cụ thể nào đó

```
class Circle
{
    static void Main(string[] args)
    {
        const float _pi = 3.14F;
        float radius = 5;
        float area = _pi * radius * radius;
        Console.WriteLine("Area of the circle is " + area);
    }
}
```

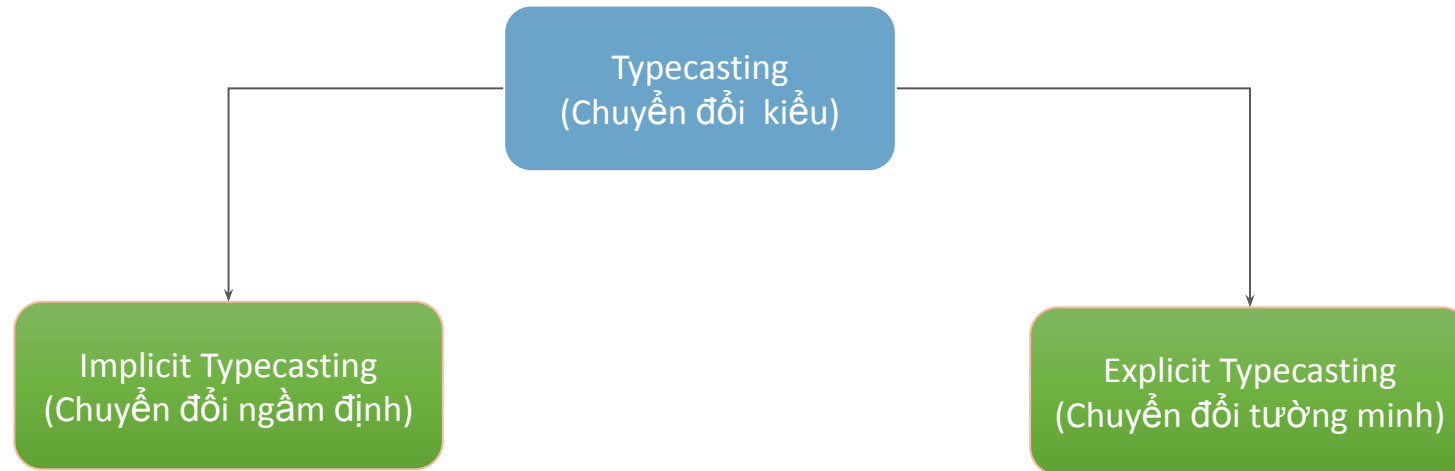


g với nhau và

ết quả.

# Chuyển đổi dữ liệu

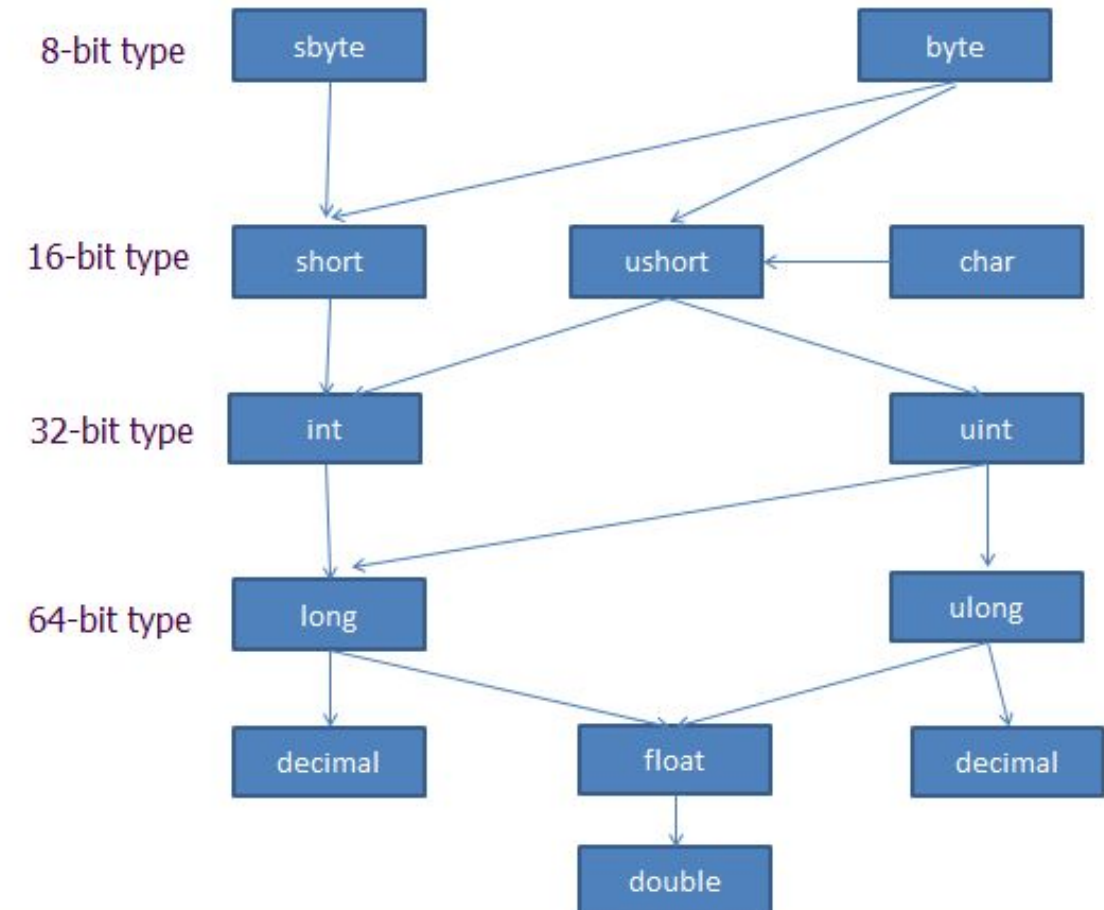
- Là cách thức chuyển dữ liệu của một kiểu này sang dữ liệu kiểu khác
- Có 2 cách chuyển đổi dữ liệu:



# Chuyển đổi ngầm định

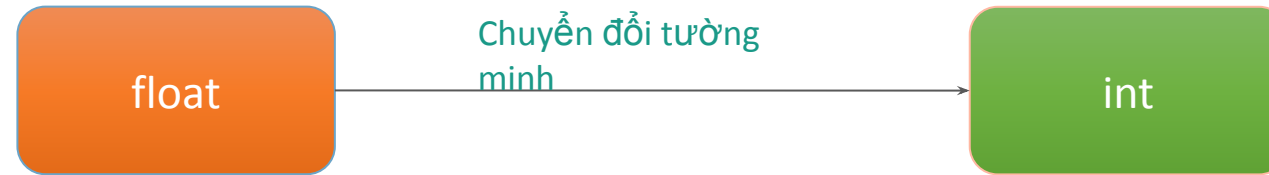
- Là cơ chế chuyển đổi tự động bởi trình biên dịch C#, các kiểu dữ liệu có miền giá trị thấp sẽ tự động chuyển về các kiểu có miền giá trị cao hơn

```
int a = 10;  
double x;  
x = a; //mặc định giá trị của a sẽ lưu vào x và mang kiểu double
```



# Chuyển đổi tường minh

- Là cơ chế chuyển đổi từ các kiểu dữ liệu có miền giá trị cao về các kiểu có miền giá trị thấp hơn.

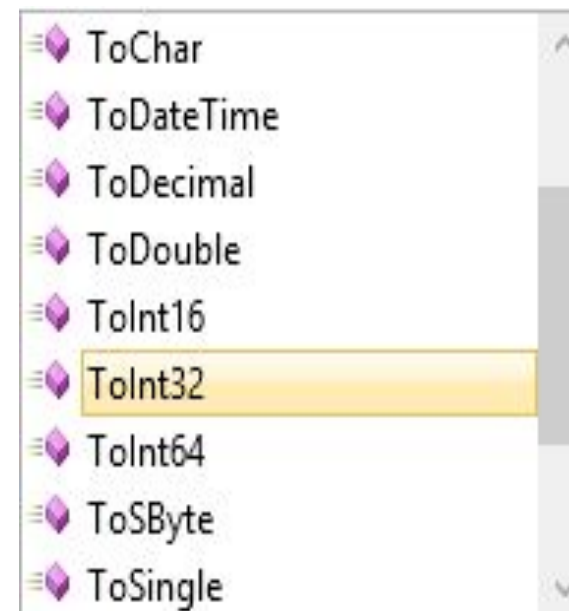


- Phương pháp
  - Dùng các phương thức của lớp Convert
  - Ép kiểu

# Sử dụng lớp Convert

- Lớp này cung cấp các phương thức để chuyển dữ liệu từ một kiểu này qua một kiểu khác.

```
string a = "39";  
string b = "21";  
int a1 = Convert.ToInt32(a);  
int b1 = Convert.ToInt32(b);  
Console.WriteLine(a + b);  
Console.WriteLine(a1 + b1);  
//Ctrl+F5 để xem kết quả:
```



# Ép kiểu

- Là cách ép giá trị của một kiểu nào đó về kiểu mong muốn
- Cú pháp:

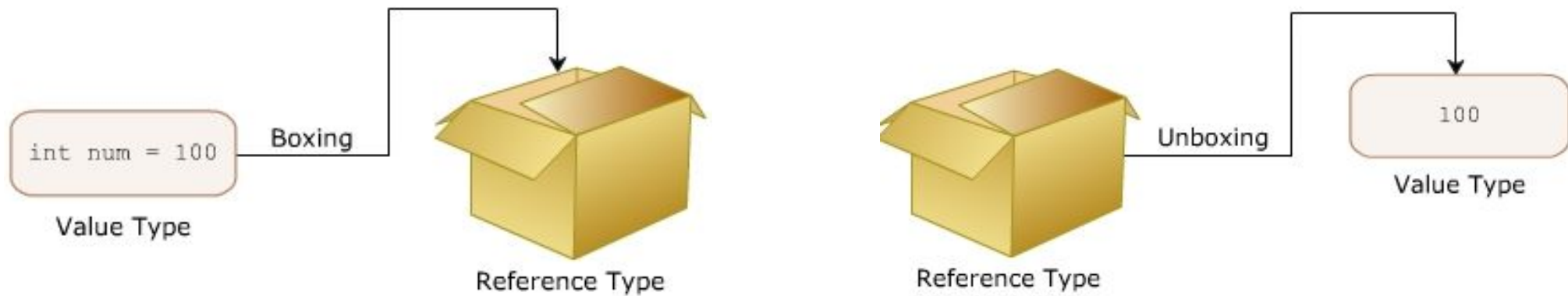
(Data\_type) value;

```
double a = 10.56;  
int b = (int)a;  
Console.WriteLine(a);  
//Ctrl+F5 để xem kết quả:
```



# Boxing và Unboxing

- Boxing là cách chuyển một kiểu giá trị như int, float, double.. thành kiểu tham chiếu như Object.



- Unboxing là `Object x=100;` tham chiếu thành kiểu `int y=(int) x;`

# Kiểu liệt kê (enumeration type) 1-2

- Kiểu liệt kê là một kiểu bao gồm một tập các số nguyên do người dùng định nghĩa, không chỉ vậy, bạn có thể sử dụng các tên để nhớ thay cho các giá trị số nguyên.
- Ví dụ khai báo

```
/// <summary>
/// Định nghĩa bộ liệt kê với các tên
/// ứng với các trí trị số tùy ý
/// </summary>
public enum PostStatus
{
    New=1,
    Viewed=3,
    Pending=2,
    Submitted=4,
    Cancel=0
}
```

```
/// <summary>
/// Định nghĩa bộ liệt kê với các tên
/// ứng với các trí trị số nguyên mặc định từ 0..n
/// </summary>
public enum Position
{
    Top,
    Bottom,
    Left,
    Right
}
```

# Kiểu liệt kê (enumeration type) 2-2

- Sử dụng

```
class Program
{
    static void Main(string[] args)
    {
        /*
         * khai báo biến kiểu liệt kê và gán giá trị
         * lưu ý các giá trị chỉ nằm trong phạm vi đã định nghĩa
         */
        Position position = Position.Right;
        //kiểm tra
        if (position == Position.Right)
        {
            Console.WriteLine("Vi tri la 'Right' tuong ung so nguyen la:" + (int)position);
        }
    }
}
```

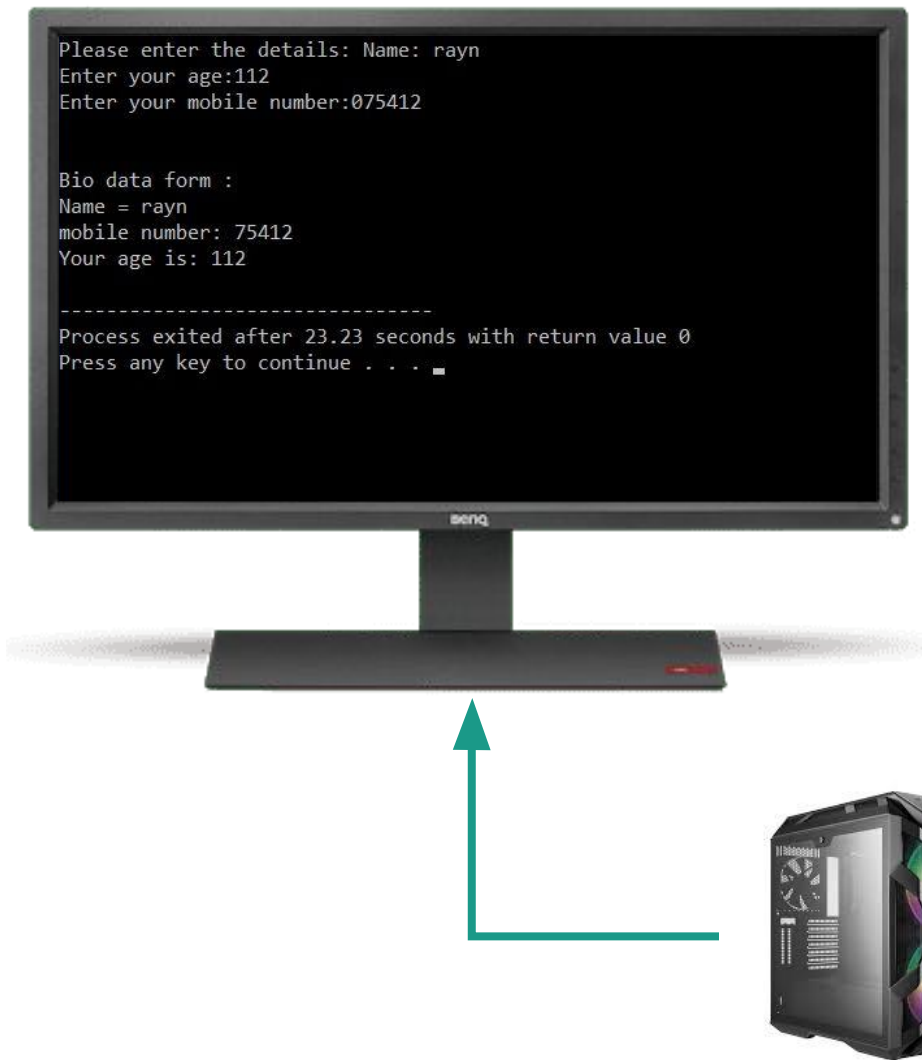
# Chú thích trong C#

- Chú thích giúp chương trình dễ hiểu đối với người đọc, để bảo trì sau này, ta nên tập thói quen viết chú thích cho các đoạn code.
- Có 3 loại comment
  - Comment từng hàng: `“//”`
  - Comment theo block: `/*Comment*/`
  - Comment dạng XML: Chỉ dùng đối với lớp, phương thức ... Ta chỉ cần gõ `///` thì VS sẽ tự sinh code.
  - Sử dụng cặp từ khóa `#region.....#endregion`
- Ví dụ (Xem live code)

# Mục tiêu 2

- Xuất dữ liệu trong ứng dụng Console
- Nhập dữ liệu trong ứng dụng Console
- Định dạng dữ liệu

# Ứng dụng Console là gì?



- Ứng dụng Console là loại ứng dụng cho phép người dùng thực hiện các thao tác nhập xuất ở dạng dòng lệnh. Trên console gồm 3 loại:
  - Nhập chuẩn (Standard in): dùng để nhận các tham số đầu vào
  - Xuất chuẩn (Standard out): dùng để hiển thị kết quả đầu ra
  - Thông báo lỗi (Standard err): dùng để thông báo lỗi



# Xuất dữ liệu ra màn hình Console (1-3)

- Trong C# để hiển thị ra màn hình Console chúng ta sử dụng 1 trong 2 phương thức:
  - `Console.Write()` : Hiển thị kết quả ra màn hình.
  - `Console.WriteLine()` : Hiển thị kết quả ra màn hình và xuống dòng.

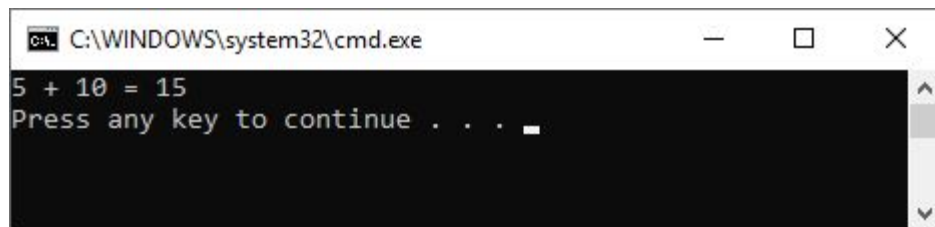
```
Console.WriteLine("Xin chao moi nguoi");  
Console.WriteLine("Toi ten la Nguyen Van A");  
Console.Write("Xin chao moi nguoi");  
Console.Write("Toi ten la Nguyen Van A");  
// Nhấn Ctrl + F5 và kiểm tra sự khác biệt
```

# Xuất dữ liệu ra màn hình Console (2-3)

- Xuất dữ liệu theo vị trí quy định

`Console.Write("{0},{1},...",var1, var2,...);`

```
int a = 5;
int b = 10;
int Tong = a + b;
Console.WriteLine("{0} + {1} = {2}", a, b, Tong);
//Ctrl+F5 để xem kết quả:
```

A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The window has standard minimize, maximize, and close buttons. The command prompt displays the output '5 + 10 = 15' on the first line. The second line shows 'Press any key to continue . . . ' followed by a cursor. A vertical scrollbar is visible on the right side of the window.

```
C:\WINDOWS\system32\cmd.exe
5 + 10 = 15
Press any key to continue . . .
```

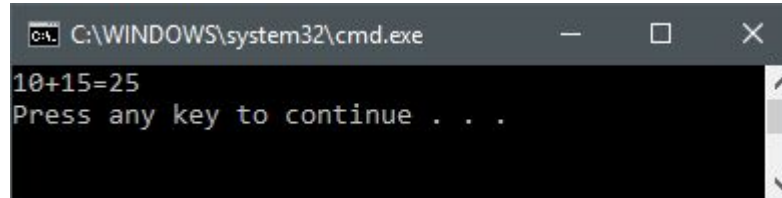


# Xuất dữ liệu ra màn hình Console (3-3)

- Đưa biểu thức trực tiếp vào các vị trí in

`Console.Write($"{exp1},{exp2},...");`

```
int x = 10,y=15,z;  
z = x + y;  
Console.WriteLine($"{x}+{y}={z}");
```

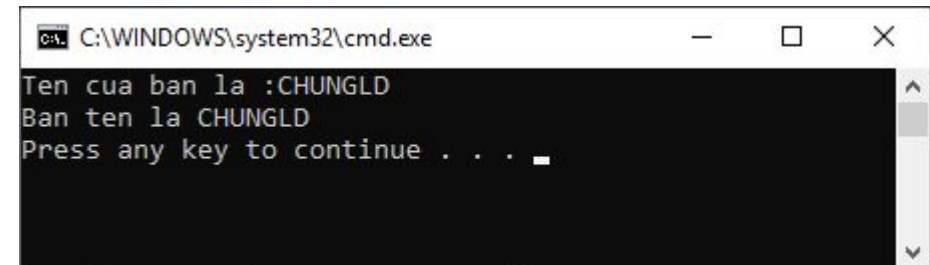
A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The window has standard minimize, maximize, and close buttons. The command prompt displays the output '10+15=25' on the first line and 'Press any key to continue . . .' on the second line. A vertical scrollbar is visible on the right side of the window.

```
C:\WINDOWS\system32\cmd.exe  
10+15=25  
Press any key to continue . . .
```

# Nhập dữ liệu từ màn hình Console

- Trong C#, để nhập dữ liệu từ bàn phím thông qua màn hình Console chúng ta sử dụng 1 trong 2 phương thức của lớp Console:
  - `Console.Read()` : Đọc một ký tự từ bàn phím.
  - `Console.ReadLine()` : Đọc một chuỗi kí tự từ bàn phím.

```
string name = "";  
Console.Write("Ten cua ban la :");  
name = Console.ReadLine();  
Console.WriteLine("Ban ten la {0}", name);  
//Ctrl+F5 để nhập và xem kết quả
```

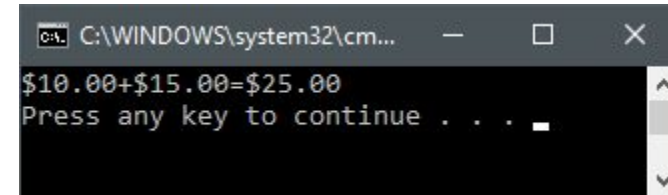


```
C:\WINDOWS\system32\cmd.exe  
Ten cua ban la :CHUNGLD  
Ban ten la CHUNGLD  
Press any key to continue . . .
```

# Các chỉ thị định dạng dữ liệu

- Chỉ thị định dạng là các ký tự đặc biệt được sử dụng để hiển thị các giá trị của các biến với 1 định dạng cụ thể.
  - `Console.WriteLine("{format specifier}", variables)`

```
int x = 10,y=15,z;  
z = x + y;  
Console.WriteLine("{0:C}+{1:C}={2:C}",x,y,z);
```



# Chỉ thị định dạng số

Format Specifier	Name	Description
C or c	Currency	The number is converted to a string that represents a currency amount.
D or d	Decimal	The number is converted to a string of decimal digits (0-9), prefixed by a minus sign in case the number is negative. The precision specifier indicates the minimum number of digits desired in the resulting string. This format is supported for fundamental types only.
E or e	Scientific (Exponential)	The number is converted to a string of the form ‘-d.ddd...E+ddd’ or ‘-d.ddd...e+ddd’, where each ‘d’ indicates a digit (0-9).
F or f	Fixed-point	The number is converted to a string of the form ‘-ddd.ddd...’ where each ‘d’ indicates a digit (0-9). If the number is negative, the string starts with a minus sign.
N or n	Number	The number is converted to a string of the form ‘-d,ddd,ddd.ddd...’, where each ‘d’ indicates a digit (0-9). If the number is negative, the string starts with a minus sign.
X or x	Hexadecimal	The number is converted to a string of hexadecimal digits. Uses “X” to produce “ABCDEF”, and “x” to produce “abcdef”.

# Chỉ thị định dạng ngày tháng thời gian (1-3)

Format Specifier	Name	Description
d	Short date	Displays date in short date pattern. The default format is 'mm/dd/yyyy'.
D	Long date	Displays date in long date pattern. The default format is 'dddd*, MMMM*, dd, yyyy'.
f	Full date/time (short time)	Displays date in long date and short time patterns, separated by a space. The default format is 'dddd*, MMMM* dd, yyyy HH*:mm*'.
F	Full date/time (long time)	Displays date in long date and long time patterns, separated by a space. The default format is 'dddd*, MMMM* dd, yyyy HH*: mm*: ss*'.
g	General date/time (short time)	Displays date in short date and short time patterns, separated by a space. The default format is 'MM/dd/yyyy HH*:mm*'.

# Chỉ thị định dạng ngày tháng thời gian (2-3)

Format Specifier	Name	Description
G	General date/time (long time)	Displays date in short date and long time patterns, separated by a space. The default format is 'MM/dd/yyyy HH*:mm*:ss*'.
m or M	Month day	Displays only month and day of the date. The default format is 'MMMM*dd'.
T	Short time	Displays time in short time pattern. The default format is 'HH*: mm*'.
T	Long time	Displays time in long time pattern. The default format is 'HH*:mm*:ss*'.
y or Y	Year month pattern	Displays only month and year from the date. The default format is 'YYYY MMMM*'.

# Chỉ thị định dạng ngày tháng thời gian (3-3)

Format Specifier	Name
Ddd	Represents the abbreviated name of the day of the week
dddd	Represents the full name of the day of the week
FF	Represents the two digits of the seconds fraction
H	Represents the hour from 0 to 23
HH	Represents the hour from 00 to 23
MM	Represents the month as a number from 01 to 12
MMM	Represents the abbreviated name of the month
s	Represents the seconds as a number from 0 to 59

# Chỉ thị định dạng khác

Format Specifier	Description
0	If the value being formatted contains a digit where '0' appears, then it is copied to the result string
#	If the value being formatted contains a digit where '#' appears, then it is copied to the result string
.	The first '.' character verifies the location of the decimal separator
,	The ',' character serves as a thousand separator specifier and a number scaling specifier
%	The '%' character in a format string multiplies a number with 100 before it is formatted
E0, E+0,E-0, e0, e+0, e-0	If any of the given strings are present in the format string and they are followed by at least one '0' character, then the number is formatted using scientific notation
\	The backslash character causes the next character in the format string to be interpreted as an escape sequence
'ABC' "ABC"	The characters that are enclosed within single or double quotes are copied to the result string
;	The ';' character separates a section into positive, negative, and zero numbers
Other	Any of the other characters are copied to the result string



# HỎI ĐÁP





# TRẢI NGHIỆM THỰC HÀNH