

Nhập môn Lập trình – IT001

Buổi 03 – Kiểu dữ liệu cơ bản

Mục tiêu buổi học – CĐR

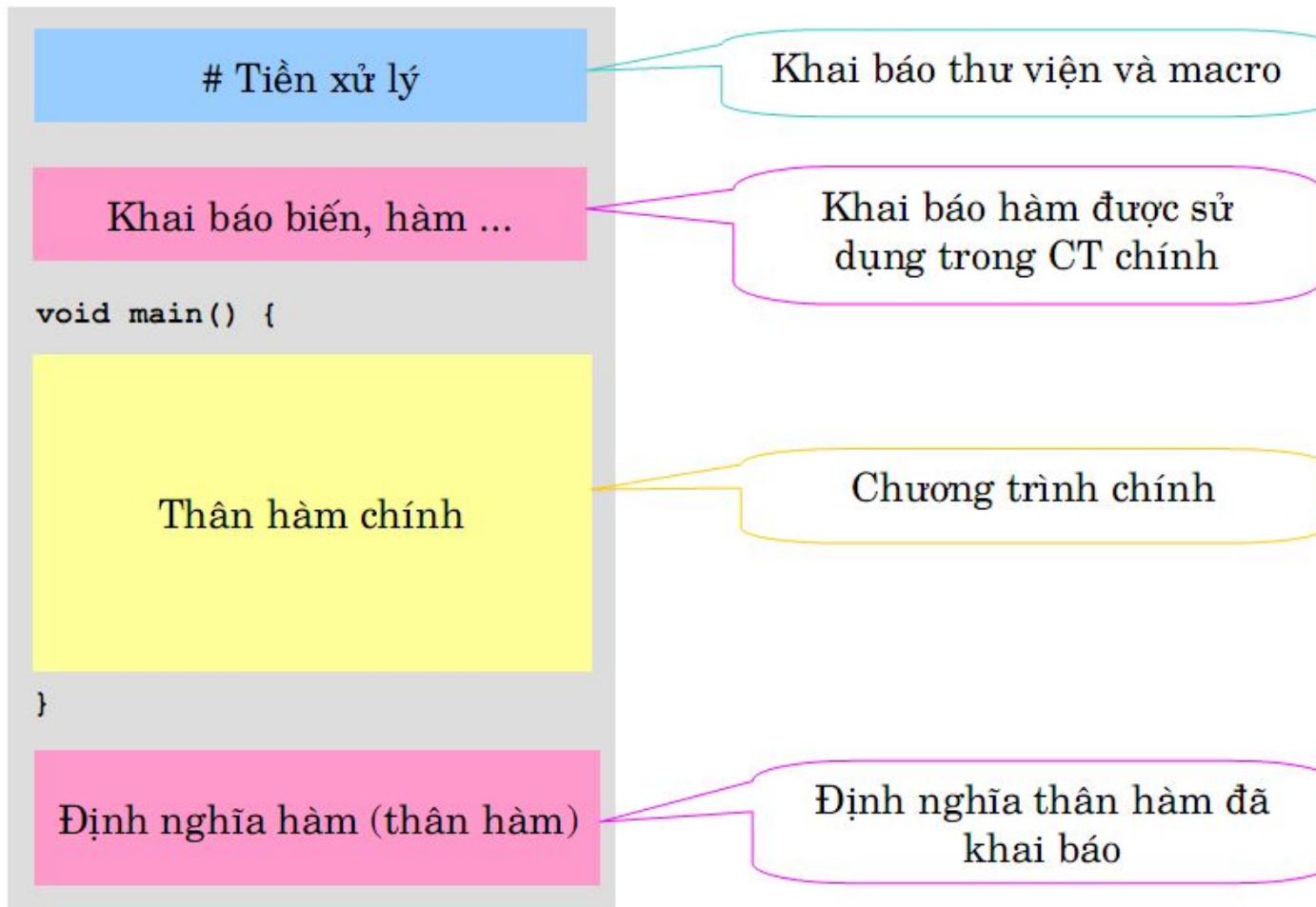
- Hiểu và vận dụng được các kiểu dữ liệu cơ bản



Nội dung

- Cấu trúc một chương trình
- Chương trình đơn giản
- Bộ từ vựng của C
- Quy tắc soạn thảo mã nguồn
- Các kiểu dữ liệu cơ bản và phép toán

Cấu trúc chương trình C/C++



Chương trình đơn giản

```
# include <stdio.h>
# define PI  3.1415
void main ()
{
    float cv,dt, r = 3.1;
    cv=2*r*PI;
    dt=PI*r*r;
    printf("\nChu vi = %10.2f",cv);
    printf("\nDien tich =%10.2f",dt);
}
```

Chương trình đơn giản

```
# include <stdio.h>
# define PI  3.1415
void main ()
{
    float cv,dt, r = 3.1;
    cv=2*r*PI;
    dt=PI*r*r;
    printf("\nChu vi = %10.2f",cv);
    printf("\nDien tich =%10.2f",dt);
}
```

Bộ từ vựng của C

- ❖ Các ký tự được sử dụng
 - Bộ chữ cái 26 ký tự Latinh **A, B, C, ..., Z, a, b, c, ..., z**
 - Bộ chữ số thập phân : **0, 1, 2, ..., 9**
 - Các ký hiệu toán học : **+ - * / = < > ()**
 - Các ký tự đặc biệt : **. , : ; [] % \ # \$ '**
 - Ký tự gạch nối **_** và khoảng trắng **" "**

Bộ từ vựng của C

- Từ khóa (keyword)
 - Các từ **dành riêng** trong ngôn ngữ.
 - **Không** thể sử dụng từ khóa để đặt tên cho biến, hàm, tên chương trình con.
 - Một số từ khóa thông dụng:
 - const, enum, signed, struct, typedef, unsigned...
 - char, double, float, int, long, short, void
 - case, default, else, if, switch
 - do, for, while
 - break, continue, goto, return

Bộ từ vựng của C

- Tên/Định danh (Identifier)
 - Một dãy ký tự dùng để chỉ tên một hằng số, hằng ký tự, tên một biến, một kiểu dữ liệu, một hàm.
 - Không được trùng với các từ khóa và được tạo thành từ các chữ cái và các chữ số nhưng bắt buộc chữ đầu phải là chữ cái hoặc _.
 - Số ký tự tối đa trong một tên là 255 ký tự và được dùng ký tự _ chen trong tên nhưng không cho phép chen giữa các khoảng trắng.

Bộ từ vựng của C

- Ví dụ Tên/Định danh (Identifier)
 - Các tên hợp lệ: GiaiPhuongTrinh, Bai_Tap1
 - Các tên không hợp lệ: 1A, Giai Phuong Trinh
 - **Phân biệt chữ hoa chữ thường**, do đó các tên sau đây khác nhau:
 - A, a
 - BaiTap, baitap, BAITAP, bAltaP, ...

Qui tắc soạn thảo mã nguồn

1. Tên/định danh (Identifier)

- Không sử dụng các từ khóa và tên kiểu cơ sở
- Các ký tự dùng được: 'A'..'Z', 'a'..'z', '0'..'9', '_'
- Phân biệt giữa chữ hoa và chữ thường: n khác N
- Ngắn nhưng đủ khả năng phân biệt, gợi nhớ để nhận biết
- Sử dụng tiếng Anh hoặc tiếng Việt không dấu (kể cả dòng chú thích)

Qui tắc soạn thảo mã nguồn

2. Sau mỗi câu lệnh có chấm phẩy;
3. Đoạn { ... } được coi là nhóm lệnh
4. Cấu trúc mã nguồn theo kiểu phân cấp
5. Bổ sung chú thích đầy đủ, hợp lý
6. Chia một chương trình lớn thành nhiều phần (hàm) nhỏ

Qui tắc soạn thảo mã nguồn

- Dấu chấm phẩy ;
 - Dùng để phân cách các câu lệnh.
 - Ví dụ: `printf("Hello World!"); printf("\n");`
- Câu chú thích
 - Đặt giữa cặp dấu `/* */` hoặc `//` (C++)
 - Ví dụ: `/* Ho & Ten: NVA */, // MSSV: 1512078`

Các kiểu dữ liệu cơ sở

- Turbo C có 4 kiểu cơ sở sau:
 - **Kiểu số nguyên**: giá trị của nó là các số nguyên như 2912, -1706, ...
 - **Kiểu số thực**: giá trị của nó là các số thực như 3.1415, 29.12, -17.06, ...
 - **Kiểu luận lý**: giá trị đúng hoặc sai.
 - **Kiểu ký tự**: 256 ký tự trong bảng mã ASCII.

Kiểu số nguyên

- Các kiểu số nguyên (có dấu)
 - n bit có dấu: $-2^{n-1} \dots +2^{n-1} - 1$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
char	1	-128 ... +127
int	2	-32.768 ... +32.767
short	2	-32.768 ... +32.767
long	4	-2.147.483.648 ... +2.147.483.647

Kiểu số nguyên

- Các kiểu số nguyên (không dấu)
 - n bit không dấu: $0 \dots 2^n - 1$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
unsigned char	1	0 ... 255
unsigned int	2	0 ... 65.535
unsigned short	2	0 ... 65.535
unsigned long	4	0 ... 4.294.967.295

Kiểu số nguyên

```
#include <iostream>
#include <limits>
using namespace std;
void main()
{
    cout << "\nInt Types" << endl;
    cout << "Size of int type is " << sizeof(int) << " bytes." << endl;
    cout << "Signed int min: " << INT_MIN << endl;
    cout << "Signed int max: " << INT_MAX << endl;
    cout << "Unsigned int min: 0" << endl;
    cout << "Unsigned int max: " << UINT_MAX << endl;
}
```

Kiểu số thực

- Các kiểu số thực (floating-point)

- Ví dụ

- $17.06 = 1.706 * 10 = 1.706 * 10^1$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
float (*)	4	$3.4 * 10^{-38} \dots 3.4 * 10^{38}$
double (**)	8	$1.7 * 10^{-308} \dots 1.7 * 10^{308}$

- (*) Độ chính xác đơn (Single-precision) chính xác đến 7 số lẻ.
- (**) Độ chính xác kép (Double-precision) chính xác đến 19 số lẻ.

Kiểu luận lý

- Đặc điểm
 - C ngầm định một cách không tường minh:
 - **false** (sai): giá trị 0.
 - **true** (đúng): giá trị khác 0, thường là 1.
 - C++: **bool**
- Ví dụ
 - 0 (false), 1 (true), 2 (true), 2.5 (true)
 - $1 > 2$ (0, false), $1 < 2$ (1, true)

Kiểu ký tự

- Đặc điểm
 - Tên kiểu: **char**
 - Miền giá trị: 256 ký tự trong bảng mã ASCII.
 - Chính là kiểu số nguyên do:
 - Lưu tất cả dữ liệu ở dạng số.
 - Không lưu trực tiếp ký tự mà chỉ lưu mã ASCII của ký tự đó.
- Ví dụ
 - Lưu số 65 tương đương với ký tự 'A'...
 - Lưu số 97 tương đương với ký tự 'a'.

Biến (variable)

- Biến là gì ?
 - Biến dùng để nhận giá trị do người dùng nhập vào khi chạy chương trình hoặc dùng để chứa giá trị của một biểu thức
- Cú pháp khai báo biến:
 - Kiểu_dữ_liệu tên_biến [= giá trị khởi tạo] ;
- Ví dụ 1:
 - int x;
 - int y=7;
 - int x=y;

Biến (variable)

- Ví dụ 2: Viết chương trình nhập vào 3 số a, b và c. Cho biết a, b, c có tạo thành 3 cạnh của tam giác không ?
- Cần khai báo 3 biến a, b, c. Chẳng hạn
 - `int a;`
 - `int b;`
 - `int c;`Hoặc có thể khai báo chung:
 - `int a,b,c;`

Địa chỉ của biến

- Khi gặp một lệnh khai báo biến, chương trình sẽ cấp phát cho biến một ô nhớ tương ứng có kích thước bằng với kích thước của kiểu dữ liệu biến

- Ví dụ

– `int n;`

– `float x;`



- Để lấy địa chỉ của ô nhớ đã cấp cho biến ta dùng cú pháp: `&tên_biến`

– VD: `&n` ☐ `ffe7`
`&x` ☐ `ffee`

Giá trị của biến

- Giá trị của biến là giá trị được lưu trữ trong ô nhớ dành cho biến đó

- VD:

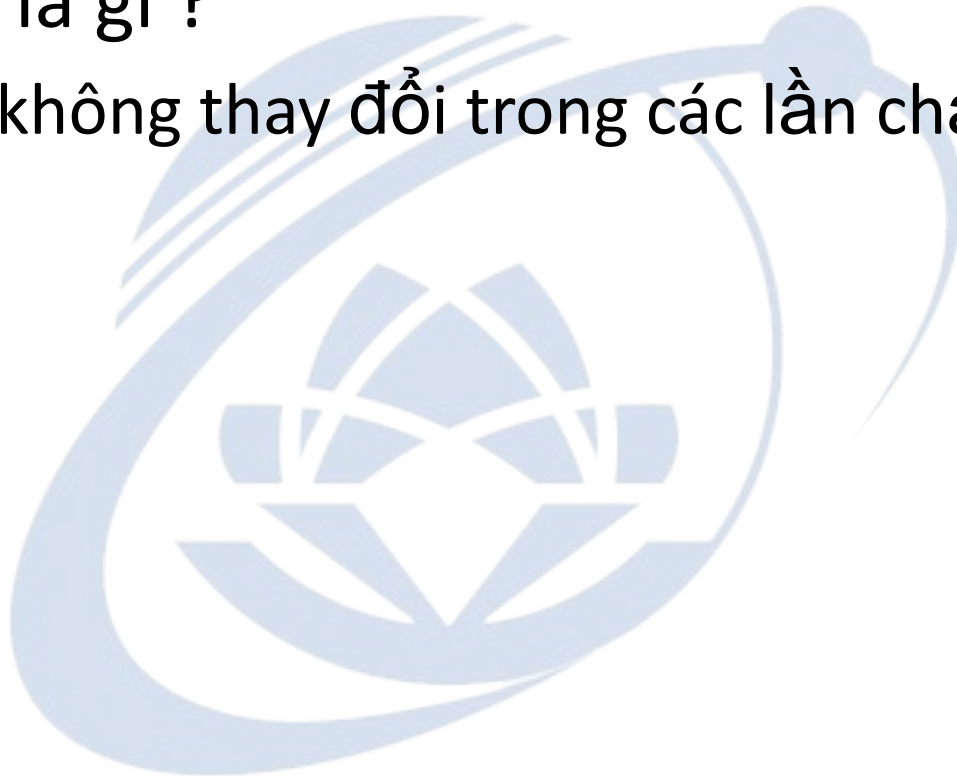
- `int x;`
- `int y=7;`
- `int z=y;`

x	?
y	7
z	7

- Để lấy giá trị của biến ta dùng cú pháp: `tên_biến`
- Để gán cho biến một giá trị mới ta dùng cú pháp: `tên_biến=giá_trị ;`
- VD:
 - Lấy giá trị của biến `y` thì viết: `y` (giá trị của `y` bằng 7)
 - `z=2+7+y;`

Hằng số

- Hằng số là gì ?
 - Giá trị không thay đổi trong các lần chạy chương trình



Hằng số



Hằng số

Cú pháp
`#define <tên hằng> <giá trị>`
hoặc sử dụng từ khóa `const`.



Ví dụ

```
#define MAX 100           // Không có ;  
#define PI 3.14          // Không có ;  
const int MAX = 100;  
const float PI = 3.14;
```

BIỂU THỨC

1. Biểu thức

- Tạo thành từ các **toán tử** (Operator) và các **toán hạng** (Operand).
- Toán tử tác động lên các giá trị của toán hạng và cho giá trị có kiểu nhất định.
- Toán tử: **+**, **−**, *****, **/**, **%**....
- Toán hạng: **hằng**, **biến**, **lời gọi hàm**...
- Ví dụ
 - $2 + 3$, $a / 5$, $(a + b) * 5$, ...

Toán tử gán

- Khái niệm
 - Gán giá trị cho biến.
- Cú pháp
 - $\langle \text{biến} \rangle = \langle \text{giá trị} \rangle;$
 - $\langle \text{biến} \rangle = \langle \text{biến} \rangle;$
 - $\langle \text{biến} \rangle = \langle \text{biểu thức} \rangle;$
 - Có thể thực hiện liên tiếp phép gán.

Toán tử gán

- Ví dụ

```
void main()
{
    int a, b, c, d, e, thuong;
    a = 10;
    b = a;
    thuong = a / b;
    a = b = c = d = e = 156;
    e = 156;
    d = e;
    c = d;
    b = c;
    a = b;
}
```

Các toán tử toán học

- Toán tử 1 ngôi
 - Chỉ có một toán hạng trong biểu thức.
 - **++** (tăng 1 đơn vị), **--** (giảm 1 đơn vị)
 - Đặt trước toán hạng
 - Ví dụ **++x** hay **--x**: thực hiện tăng/giảm **trước**.
 - Đặt sau toán hạng
 - Ví dụ **x++** hay **x--**: thực hiện tăng/giảm **sau**.
- Ví dụ
 - `x = 10; y = x++; // y = 10 và x = 11`
 - `x = 10; y = ++x; // x = 11 và y = 11`

Các toán tử toán học

- Toán tử 2 ngôi
 - Có hai toán hạng trong biểu thức.
 - $+$, $-$, $*$, $/$, $\%$ (chia lấy phần dư)
 - $x = x + y \Leftrightarrow x += y$;
- Ví dụ
 - $a = 1 + 2$; $b = 1 - 2$; $c = 1 * 2$; $d = 1 / 2$;
 - $e = 1 * 1.0 / 2$; $f = \text{float}(1) / 2$; $g = \text{float}(1 / 2)$;
 - $h = 1 \% 2$;
 - $x = x * (2 + 3 * 5)$; $\Leftrightarrow x *= 2 + 3 * 5$;

Các toán tử trên bit

- Các toán tử trên bit
 - Tác động lên các bit của toán hạng (nguyên).
 - **&** (and), **|** (or), **^** (xor), **~** (not hay lấy số bù 1)
 - **>>** (shift right), **<<** (shift left)
 - Toán tử gộp: **&=**, **|=**, **^=**, **~=**, **>>=**, **<<=**

&			 		
	0	1		0	1
0	0	0	0	0	1
1	0	1	1	1	1

^			~		
	0	1		0	1
0	0	1	1	1	0
1	1	0			

Các toán tử trên bit

- Ví dụ

```
void main()
{
    int a = 5;    // 0000 0000 0000 0101
    int b = 6;    // 0000 0000 0000 0110

    int z1, z2, z3, z4, z5, z6;
    z1 = a & b;   // 0000 0000 0000 0100
    z2 = a | b;   // 0000 0000 0000 0111
    z3 = a ^ b;   // 0000 0000 0000 0011
    z4 = ~a;      // 1111 1111 1111 1010
    z5 = a >> 2;  // 0000 0000 0000 0001
    z6 = a << 2;  // 0000 0000 0001 0100
}
```

Các toán tử quan hệ

- Các toán tử quan hệ
 - So sánh 2 biểu thức với nhau
 - Cho ra kết quả 0 (hay false nếu sai) hoặc 1 (hay true nếu đúng)
 - `==`, `>`, `<`, `>=`, `<=`, `!=`
- Ví dụ
 - `s1 = (1 == 2); s2 = (1 != 2);`
 - `s3 = (1 > 2); s4 = (1 >= 2);`
 - `s5 = (1 < 2); s6 = (1 <= 2);`

Các toán tử luận lý

- Các toán tử luận lý
 - Tổ hợp nhiều biểu thức quan hệ với nhau.
 - **&&** (and), **||** (or), **!** (not)

&&	0	1	 	0	1
0	0	0	0	0	1
1	0	1	1	1	1

– Ví dụ

- $s1 = (1 > 2) \text{ \&\& } (3 > 4);$
- $s2 = (1 > 2) \text{ \&\& } (3 > 4);$
- $s3 = \text{!(1 > 2)};$

Toán tử điều kiện

- Toán tử điều kiện
 - Đây là toán tử 3 ngôi (gồm có 3 toán hạng)
 - <biểu thức 1> ? <biểu thức 2> : <biểu thức 3>
 - <biểu thức 1> đúng thì giá trị là <biểu thức 2>.
 - <biểu thức 1> sai thì giá trị là <biểu thức 3>.
- Ví dụ
 - $s1 = (1 > 2) ? 2912 : 1706;$
 - $\text{int } s2 = 0;$
 - $1 < 2 ? s2 = 2912 : s2 = 1706;$

Toán tử phẩy

- Toán tử phẩy
 - Các biểu thức đặt cách nhau bằng dấu ,
 - Các biểu thức con **lần lượt được tính từ trái sang phải.**
 - Biểu thức mới nhận được là **giá trị của biểu thức bên phải cùng.**
- Ví dụ
 - $x = (a++, b = b + 2);$
 - $\Leftrightarrow a++; b = b + 2; x = b;$

Độ ưu tiên của các toán tử

- Quy tắc thực hiện
 - Thực hiện biểu thức trong () sâu nhất trước.
 - Thực hiện theo thứ tự ưu tiên các toán tử.
- => Tự chủ động thêm ()
- Ví dụ
 - $n = 2 + 3 * 5;$
=> $n = 2 + (3 * 5);$
 - $a > 1 \ \&\& \ b < 2$
=> $(a > 1) \ \&\& \ (b < 2)$

Độ ưu tiên của các toán tử

Toán tử

Độ ưu tiên

() [] -> .

! ++ -- - + * (cast) & sizeof

* / %

+ -

<< >>

< <= > >=

== !=

&

|

^

&&

||

?:

= += -= *= /= %= &= ...

,



Viết biểu thức cho các mệnh đề

- x lớn hơn hay bằng 3

$x \geq 3$

- a và b cùng dấu

$((a > 0) \ \&\& \ (b > 0)) \ || \ ((a < 0) \ \&\& \ (b < 0))$

$(a > 0 \ \&\& \ b > 0) \ || \ (a < 0 \ \&\& \ b < 0)$

- p bằng q bằng r

$(p == q) \ \&\& \ (q == r)$ hoặc $(p == q \ \&\& \ q == r)$

- $-5 < x < 5$

$(x > -5) \ \&\& \ (x < 5)$ hoặc $(x > -5 \ \&\& \ x < 5)$

Câu lệnh

- Khái niệm
 - Là một chỉ thị trực tiếp, hoàn chỉnh nhằm ra lệnh cho máy tính thực hiện một số tác vụ nhất định nào đó.
 - Trình biên dịch bỏ qua các khoảng trắng (hay tab hoặc xuống dòng) chen giữa lệnh.

- Ví dụ

```
a=2912;  
a = 2912;  
a  
=  
2912;
```

Câu lệnh

- Phân loại
 - Câu lệnh đơn: chỉ gồm một câu lệnh.
 - Câu lệnh phức (khối lệnh): gồm nhiều câu lệnh đơn được bao bởi { và }
- Ví dụ

```
a = 2912;    // Câu lệnh đơn

{            // Câu lệnh phức/khối lệnh
    a = 2912;
    b = 1706;
}
```

Câu lệnh xuất

- Có 2 cách:
 - Sử dụng lệnh xuất trong C++: cout
 - Sử dụng lệnh xuất trong C: printf
- Lựa chọn tùy thuộc vào lập trình viên

Câu lệnh xuất cout (C++)

- Thư viện

```
#include <iostream>  
using namespace std;
```

- Cú pháp

```
cout << Tham_số_1 << Tham_số_2 << ... << Tham_số_k ;
```

Tham số có thể:

- Văn bản thường (literal text)
- Ký tự điều khiển (escape sequence)
- Biến, hằng số, biểu thức, hàm

- Ví dụ

```
int i=10;  
cout<<"gia trị là "<<i;
```

Văn bản thường (literal text)

- Ví dụ
 - Xuất chuỗi **Hello World**
 - ☐ `printf("Hello "); printf("World");`
 - ☐ `printf("Hello World");`
 - Xuất chuỗi **a + b**
 - ☐ `printf("a + b");`

Ký tự điều khiển (escape sequence)

Gồm dấu \ và một ký tự như trong bảng sau:

Ký tự điều khiển	Ý nghĩa
<code>\a</code>	Tiếng chuông
<code>\b</code>	Lùi lại một bước
<code>\n</code>	Xuống dòng
<code>\t</code>	Dấu tab
<code>\\</code>	In dấu \
<code>\?</code>	In dấu ?
<code>\"</code>	In dấu "

- Ví dụ
 - `printf("\t"); printf("\n");`
 - `printf("\t\n");`

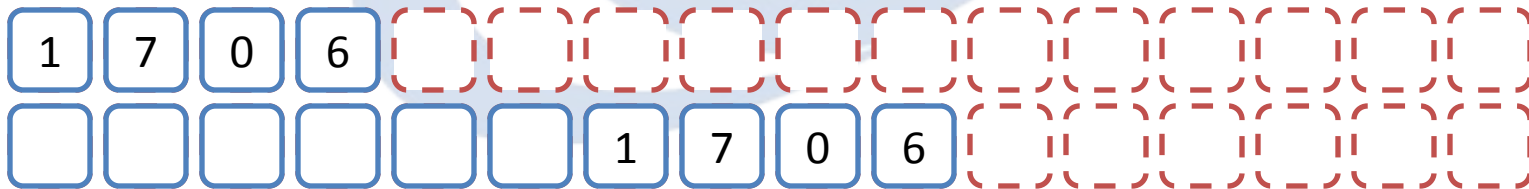
Thiết lập độ rộng khi xuất (dùng cout)

- Cú pháp: **cout.width(n)**

- n là độ rộng mới

Chú ý: độ rộng quy định n chỉ có tác dụng cho một giá trị xuất. Sau đó C++ lại áp dụng độ rộng quy định bằng 0.

```
int a = 1706;
cout<<a<<"\n";
cout.width(10);
cout<<a;
```



Độ chính xác khi xuất (dùng cout)

- Cú pháp: **cout.precision(n)**

- n là độ chính xác áp dụng

Chú ý: độ chính xác được thiết lập sẽ có hiệu lực cho tới khi gặp một câu lệnh thiết lập độ chính xác mới

```
float x = 176.859;  
cout<<x<<"\n";  
cout.precision(5);  
cout<<x<<"\n";
```

[illegible]

Câu lệnh xuất printf (C)

- Thư viện
 - `#include <stdio.h>` (**s**tandard **i**nput/**o**utput)
- Cú pháp
 - `printf(<chuỗi định dạng>[, <đs1>, <đs2>, ...]);`
 - <chuỗi định dạng> là cách trình bày thông tin xuất và được đặt trong cặp nháy kép **" "**.
 - Văn bản thường (literal text)
 - Ký tự điều khiển (escape sequence)
 - Đặc tả (conversion specifier)

Đặc tả (conversion specifier)

- Gồm dấu % và một ký tự.
- Xác định kiểu của biến/giá trị muốn xuất.
- Các đối số chính là các biến/giá trị muốn xuất, được liệt kê theo thứ tự cách nhau dấu phẩy.

Đặc tả	Ý nghĩa	
%c	Ký tự	char
%d, %ld	Số nguyên có dấu	int, short, long
%f, %lf	Số thực	float, double
%s	Chuỗi ký tự	char[], char*
%u	Số nguyên không dấu	unsigned int/short/long

Đặc tả (conversion specifier)

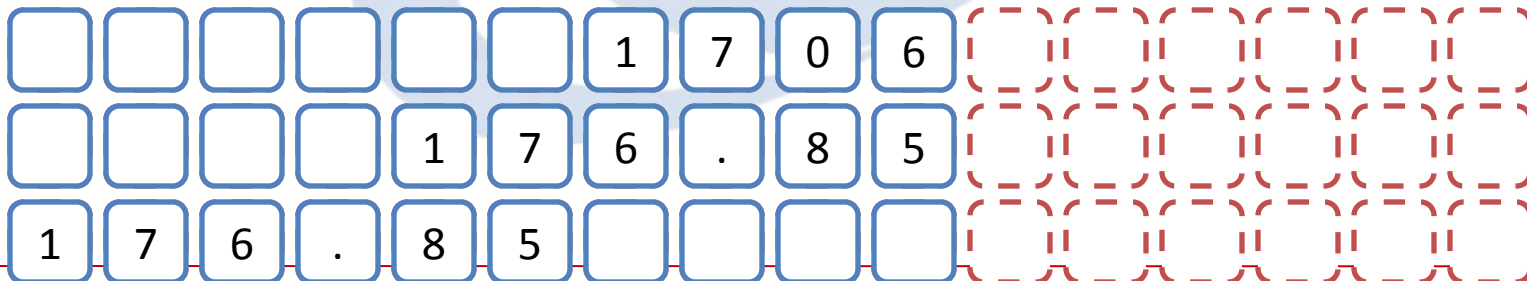
- Ví dụ
 - `int a = 10, b = 20;`
 - `printf("%d", a);` ☐ Xuất ra 10
 - `printf("%d", b);` ☐ Xuất ra 20
 - `printf("%d %d", a, b);` ☐ Xuất ra 10 20

 - `float x = 15.06;`
 - `printf("%f", x);` ☐ Xuất ra 15.060000
 - `printf("%f", 1.0/3);` ☐ Xuất ra 0.333333

Định dạng xuất (printf)

- Cú pháp
 - Định dạng xuất số nguyên: %**nd**
 - Định dạng xuất số thực: %**n.kd**

```
int a = 1706;
float x = 176.85;
printf("%10d", a);printf("\n");
printf("%10.2f", x);printf("\n");
printf("%.2f", x);printf("\n");
```



Định dạng xuất (printf)

- Phối hợp các thành phần
 - `int a = 1, b = 2;`
 - Xuất **1** **cong** **2** **bang** **3** và **xuống dòng**.
 - `printf("%d", a);` // Xuất giá trị của biến a
 - `printf(" cong ");` // Xuất chuỗi " cong "
 - `printf("%d", b);` // Xuất giá trị của biến b
 - `printf(" bang ");` // Xuất chuỗi " bang "
 - `printf("%d", a + b);` // Xuất giá trị của a + b
 - `printf("\n");` // Xuất điều khiển xuống dòng \n
 - `printf("%d cong %d bang %d\n", a, b, a+b);`

Câu lệnh nhập

- Có 2 cách:
 - Sử dụng lệnh nhập trong C++: `cin`
 - Sử dụng lệnh nhập trong C: `scanf`
- Lựa chọn tùy thuộc vào lập trình viên

Câu lệnh nhập cin (C++)

- Thư viện

```
#include <iostream>  
using namespace std;
```

- Cú pháp

```
cin >> Tham_số_1 >> Tham_số_2 >> ... >> Tham_số_k;
```

Lưu ý: tham số không có dạng chuỗi

- Ví dụ

```
int i;  
cin>>i;  
int a,b;  
cin>>a>>b;
```

Câu lệnh nhập scanf (C)

- Thư viện
 - #include <stdio.h> (**s**tandard **i**nput/**o**utput)
- Cú pháp
 - **scanf(<chuỗi định dạng>[, <đs1>, <đs1>, ...]);**
 - <chuỗi định dạng> giống định dạng xuất nhưng chỉ có các đặc tả.
 - Các đối số là tên các biến sẽ chứa giá trị nhập và được đặt trước dấu **&**

Câu lệnh nhập

- Ví dụ, cho a và b kiểu số nguyên
 - `scanf("%d", &a);` // Nhập giá trị cho biến a
 - `scanf("%d", &b);` // Nhập giá trị cho biến b
 - ☐ `scanf("%d%d", &a, &b);`
 - Các câu lệnh sau đây sai
 - `scanf("%d", a);` // Thiếu dấu &
 - `scanf("%d", &a, &b);` // Thiếu %d cho biến b
 - `scanf("%f", &a);` // a là biến kiểu số nguyên
 - `scanf("%9d", &a);` // không được định dạng
 - `scanf("a = %d, b = %d", &a, &b);`

Một số hàm hữu ích khác

- Các hàm trong thư viện toán học
 - `#include <math.h>`
 - 1 đầu vào: **double**, Trả kết quả: **double**
 - `acos`, `asin`, `atan`, `cos`, `sin`, ...
 - `exp`, `log`, `log10`
 - `sqrt`
 - `ceil`, `floor`
 - `abs`, `fabs`
 - 2 đầu vào: **double**, Trả kết quả: **double**
 - `double pow(double x, double y)`

Một số hàm hữu ích khác




- Ví dụ
 - `int x = 4, y = 3, z = -5;`
 - `float t = -1.2;`
 - `float kq1 = sqrt(x1);`
 - `int kq2 = pow(x, y);`
 - `float kq3 = pow(x, 1/3);`
 - `float kq4 = pow(x, 1.0/3);`
 - `int kq5 = abs(z);`
 - `float kq6 = fabs(t);`

Bài tập

1. Nhập năm sinh của một người và tính tuổi của người đó.
2. Nhập 2 số a và b. Tính tổng, hiệu, tích và thương của hai số đó.
3. Nhập tên sản phẩm, số lượng và đơn giá. Tính tiền và thuế giá trị gia tăng phải trả, biết:
 - a. $\text{tiền} = \text{số lượng} * \text{đơn giá}$
 - b. $\text{thuế giá trị gia tăng} = 10\% \text{ tiền}$



Bài tập thực hành

-  4. Nhập điểm thi và hệ số 3 môn Toán, Lý, Hóa của một sinh viên. Tính điểm trung bình của sinh viên đó.
-  5. Nhập bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.
-  6. Nhập vào số xe (gồm 4 chữ số) của bạn. Cho biết số xe của bạn được mấy nút?



Bài tập 1

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int NamSinh, Tuoi;
    printf("Nhap nam sinh: ");
    scanf("%d", &NamSinh);
    Tuoi = 2007 - NamSinh;
    printf("Tuoi cua ban la %d", Tuoi);
    getch();
}
```

Bài tập 2

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int a, b;
    printf("Nhap hai so nguyen: ");
    scanf("%d%d", &a, &b);
    Tong = a + b; Hieu = a - b;
    Tich = a * b; Thuong = a / b;
    printf("Tong cua a va b: %d", Tong);
    printf("Hieu cua a va b: %d", Hieu);
    printf("Tich cua a va b: %d", Tich);
    printf("Thuong cua a va b: %d", Thuong);
}
```

Bài tập 3

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int SoLuong, DonGia, Tien;
    float VAT;

    printf("Nhap so luong va don gia: ");
    scanf("%d%d", &SoLuong, &DonGia);
    Tien = SoLuong * DonGia;
    VAT = Tien * 0.1;
    printf("Tien phai tra: %d", Tien);
    printf("Thue phai tra: %.2f", VAT);
}
```


Bài tập 4

```
#include <stdio.h>
#include <conio.h>

void main()
{
    float T, L, H, DTB;
    int HsT, HsL, HsH;
    printf("Nhap diem Toan, Ly, Hoa: ");
    scanf("%f%f%f", &T, &L, &H);
    printf("Nhap he so Toan, Ly, Hoa: ");
    scanf("%d%d%d", &HsT, &HsL, &HsH);
    DTB = (T * HsT + L * HsL + H * HsH) /
        (HsT + HsL + HsH);
    printf("DTB cua ban la: %.2f", DTB);
}
```

Bài tập 5

```
#include <stdio.h>
#include <conio.h>
#define PI 3.14

void main()
{
    float R, ChuVi, DienTich;
    printf("Nhap ban kinh duong tron: ");
    scanf("%f", &R);
    ChuVi = 2*PI*R;
    DienTich = PI*R*R;
    printf("Chu vi: %.2f", ChuVi);
    printf("Dien tich: %.2f", DienTich);
}
```

Bài tập 6

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int n;
    int n1, n2, n3, n4, SoNut;
    printf("Nhap bien so xe (4 so): ");
    scanf("%d", &n);
    n4 = n % 10; n = n / 10;
    n3 = n % 10; n = n / 10;
    n2 = n % 10; n = n / 10;
    n1 = n;
    SoNut = (n1 + n2 + n3 + n4) % 10;
    printf("So nut la: %d", SoNut);
}
```