

IT001 – NHẬP MÔN LẬP TRÌNH

CÁC PHÉP TOÁN



double vs long double???

#define vs const

double vs long double???

- Phụ thuộc vào trình biên dịch và hệ điều hành.
- Kiểu double là 8 bytes.
- Thông thường hệ điều hành x86 kiểu long double là 8 bytes, một số hệ điều hành x64 là 16 bytes.

#define vs const

- Sử dụng câu lệnh #define thì khi biên dịch tên hằng sẽ được thay thế bằng giá trị. → Ko cần xài bộ nhớ để lưu hằng
- const là một biến hằng → Chiếm dung lượng trên bộ nhớ



5. Các phép toán
6. Biểu thức
7. Nhập xuất dữ liệu
8. Một số hàm hữu ích
9. Một số ví dụ minh họa



5. Các phép toán

1. Toán tử gán
2. Toán tử toán học
3. Toán tử tăng giảm
4. Toán tử phẩy
5. Toán tử toán học và gán
6. Toán tử bit
7. Toán tử điều kiện
8. Toán tử quan hệ
9. Toán tử luận lý
10. Độ ưu tiên các toán tử



5.1. Toán tử gán - Assignment operator

Dùng để gán giá trị cho 1 biến

```
int x = 10;
```

Gán 10 cho biến x

```
int y = 10;  
int x = y;
```

Gán giá trị của y cho biến x

```
int a, b;  
a = 10;  
b = 4;  
a = b;  
b = 7;
```

a = ?, b = ?
a = 10, b = ?
a = 10, b = 4
a = 4, b = 4
a = 4, b = 7

```
int x;  
int y = 2 + (x = 5);
```

x = 5;
y = 2 + x;

```
int x = y = z = 5;
```

Gán giá trị 5 cho 3 biến z, y, x

5.2. Toán tử toán học - Arithmetic operators



Phép toán	Giải thích	Ví dụ:
+	Cộng	$x = 11 + 3$
-	Trừ	$x = 11 - 3$
*	Nhân	$x = 11 * 3$
/	Chia	$x = 11 / 3.$
/	Lấy phần nguyên	$x = 11 / 3$
%	Lấy phần dư	$x = 11 \% 3$

???

Phép /

Khi nào là phép chia?

→ Khi 1 trong các đối số là số thực

Khi nào là phép lấy phần nguyên?

→ Khi các đối số đều là số nguyên

```
float a = 5 / 2;  
float b = 5 / 2.;  
float c = 5. / 2;  
float d = (float)5/2;
```

a, b, c, d = ???

5.2. Toán tử toán học - Arithmetic operators



```
#include <iostream>
int main(){
    int a = 123456;
    int b = 654321;
    std::cout<<a+b<<"\n";
    std::cout<<a-b<<"\n";
    std::cout<<a*b<<"\n";
    std::cout<<a/b<<"\n";
    std::cin.get();
    return 0;
}
```

777777
-530865

-824525248
0

Kết quả đúng

80779853376
0.188678

Kết quả
không như
mong muốn

Vấn đề:

1. Phép nhân (tràn kiểu dữ liệu)
2. Phép chia (sai logic do sử dụng phép lấy phần nguyên)

Hướng giải quyết:

Sử dụng kĩ thuật ép kiểu

```
std::cout<<(long long)a*b<<"\n";
std::cout<<(float)a/b<<"\n";
```



5.3. Toán tử tăng ++, giảm --

Dùng để tăng ++ hoặc giảm -- 1 đơn vị:

Ví dụ:

Để tăng giá trị của biến a lên 1 đơn vị ta có thể dùng các câu lệnh sau:

```
int a;  
a = a + 1;  
a += 1;  
a++;
```

Sự khác biệt giữa ++x và x++ ???

```
int x = 5;  
int y = ++x;  
// x = 6, y = 6
```

1. ++x → x = 6
2. y = x → y = 6

```
int x = 5;  
int y = x++;  
// y = 5, x = 6
```

1. y = x → x = 5, y = 5
2. x++ → y = 5, x = 6



5.4. 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:

```
int x = 0;  
int y = 2;  
int z = (++x, ++y);
```

1. ++x	→ x = 1
2. ++y	→ y = 3
3. z = y	→ z = 3

5.5. Toán tử toán học & gán - Compound assignment



Toán tử	Ví dụ	Giải thích	Phép toán
<code>+=</code>	<code>x += 5</code>	<code>x = x + 5</code>	Cộng
<code>-=</code>	<code>x -= 5</code>	<code>x = x - 5</code>	Trừ
<code>*=</code>	<code>x *= 5</code>	<code>x = x * 5</code>	Nhân
<code>/=</code>	<code>x /= 5</code>	<code>x = x / 5</code>	Chia hay lấy phần nguyên
<code>%=</code>	<code>x %= 5</code>	<code>x = x % 5</code>	Lấy phần dư
<code><<=</code>	<code>x <<= 5</code>	<code>x = x << 5</code>	Dịch trái
<code>>>=</code>	<code>x >>= 5</code>	<code>x = x >> 5</code>	Dịch phải
<code>&=</code>	<code>x &= 5</code>	<code>x = x & 5</code>	AND
<code>^=</code>	<code>x ^= 5</code>	<code>x = x ^ 5</code>	XOR
<code> =</code>	<code>x = 5</code>	<code>x = x 5</code>	OR

5.6. Toán tử bit



p	q	p & q (AND)	p ^ q (XOR)	p q (OR)	~p (NOT)
0	0	0	0	0	1
0	1	0	1	1	1
1	1	1	0	1	0
1	0	0	1	1	0

Toán tử dịch bit sang trái

3 = 0011

3 << 1 = 0110 = 6

3 << 2 = 1100 = 12

3 << 3 = 1000 = 8

Toán tử dịch bit sang phải

12 = 1100

12 >> 1 = 0110 = 6

12 >> 2 = 0011 = 3

12 >> 3 = 0001 = 1



5.6. Toán tử bit

- Ví dụ toán tử trên bit:

```
int 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
    return 0;
}
```



5.6. Toán tử bit

Ứng dụng của toán tử bit:

1. Kiểm tra chia hết cho 2

```
int number = 5;  
if(number & 1 == true)  
    std::cout<<"So le\n";  
else  
    std::cout<<"So chan\n";
```

```
a & 1    → a % 2  
a & 3    → a % 4  
a & 7    → a % 8
```

2. Tích và thương cho 2^n

```
int a = 2 << 1;  
int b = 2 << 2;  
int c = 8 >> 1;  
int d = 8 >> 2;
```

```
a = 2*21      → a = 4  
b = 2*22      → b = 8  
c = 8/21      → c = 4  
d = 8/22      → d = 2
```



5.7. Toán tử điều kiện

Cú pháp: `<điều_kiện>?<biểu_thức_1>:<biểu_thức_2>`

Ý nghĩa: Nếu `<điều_kiện>`
 đúng thì thực hiện `<biểu_thức_1>`
 ngược lại thì thực hiện `<biểu_thức_2>`

Ví dụ: Giữa 2 số a,b tìm số nào lớn hơn.

```
int a = 1;  
int b = 2;  
int c = (a>b)?a:b;
```

Ứng dụng: Dùng để định nghĩa khi câu lệnh if không thể sử dụng được.
Ví dụ: sử dụng khi định nghĩa **hằng**

Số sinh viên lớn hơn 50 sinh viên thì số lớp bằng 2, ngược lại là 1.

```
int so_sinh_vien = 55;  
const int so_lop = (so_sinh_vien>50)? 2:1;
```

5.8. Toán tử quan hệ



Toán tử	Ký hiệu	Ví dụ	Giải thích
Lớn hơn	>	$x > y$	Nếu x lớn hơn $y \rightarrow \text{true (1)}$ Ngược lại $\rightarrow \text{false (0)}$
Nhỏ hơn	<	$x < y$	Nếu x nhỏ hơn $y \rightarrow \text{true (1)}$ Ngược lại $\rightarrow \text{false (0)}$
Lớn hơn hoặc bằng	\geq	$x \geq y$	Nếu x lớn hơn hoặc bằng $y \rightarrow \text{true (1)}$ Ngược lại $\rightarrow \text{false (0)}$
Nhỏ hơn hoặc bằng	\leq	$x \leq y$	Nếu x nhỏ hơn hoặc bằng $y \rightarrow \text{true (1)}$ Ngược lại $\rightarrow \text{false (0)}$
Bằng	$==$	$x == y$	Nếu x bằng $y \rightarrow \text{true (1)}$ Ngược lại $\rightarrow \text{false (0)}$
Khác	$!=$	$x != y$	Nếu x khác $y \rightarrow \text{true (1)}$ Ngược lại $\rightarrow \text{false (0)}$

5.9. Toán tử luận lý



Toán tử	Ký hiệu	Ví dụ
NOT	!	!x
AND	&&	x && y
OR		x y

Bài tập:

1. (true && true) || false
2. (false && true) || true
3. (false && true) || false || true
4. (5 > 6 || 4 > 3) && (7 > 8)
5. !(7 > 6 || 3 > 4)

1	2		&&	
false	false	false	false	
false	true	true	false	
true	false	true	false	
true	true	true	true	

1. true
2. true
3. true
4. false
5. false

5.10. Độ ưu tiên toán tử Precedence of operators



Mức độ	Toán tử	Nhóm ưu tiên	Mức độ	Toán tử	Nhóm ưu tiên
1	::	Trái sang phải	5	* / %	Trái sang phải
2	++ --	Trái sang phải	6	+ -	Trái sang phải
	()		7	<< >>	Trái sang phải
	[]		8	< > <= >=	Trái sang phải
	. ->		9	== !=	Trái sang phải
3	++ --	Phải sang trái	10	&	Trái sang phải
	~ !		11	^	Trái sang phải
	+ -		12		Trái sang phải
	& *		13	&&	Trái sang phải
	new delete		14		Trái sang phải
	sizeof		15	= *= /= %= += -= >>= <<= &= ^= =	Phải sang trái
	(type)			?:	
4	.* ->*	Trái sang phải	16	,	Trái sang phải



5.10. Độ ưu tiên toán tử

Bài tập:

1. $x = 3 + 4 + 5;$
2. $x = y = z;$
3. $z *= ++y + 5;$
4. $a || b \&\& c || d;$

Bài giải:

1. $x = ((3 + 4) + 5);$
2. $x = (y = z);$
3. $z *= (++y) + 5;$
4. $(a || (b \&\& c)) || d;$



5. Các phép toán

Bài tập:

Bài 1: Tính

1. $(5 > 3 \ \&\& \ 4 < 8)$
2. $(4 > 6 \ \&\& \ \text{true})$
3. $(3 \geq 3 \ || \ \text{false})$
4. $(\text{true} \ || \ \text{false}) ? 4 : 5$

Bài 2: Tính

1. $7 / 4$
2. $14 \% 5$
3. $3 / 0$

6. 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 hạng: **hằng, biến, lời gọi hàm...**
- Ví dụ:
 - `int a = 2 + 3;`
 - `int b = a / 5;`
 - `int c = (a + b) * 5;`
 - `int d = (x >= 3);`
 - `(x >= 0) ^ (y < 0)` (**Biểu thức này kiểm tra gì?**)
 - `int year = 2000;`
`int month = 29;`
`if((year%4!=0) || (year%400!=0))`
`return 28;`



7. Nhập xuất dữ liệu

1. Câu lệnh xuất dữ liệu
2. Câu lệnh xuất `std::cout<<`
3. Câu lệnh xuất `printf`
4. Câu lệnh nhập
5. Câu lệnh `std::cin>>`
6. Câu lệnh `scanf`



7.1. Câu lệnh xuất

Có 2 cách:

Cách 1: Sử dụng lệnh xuất trong C++: **cout**

Cách 2: Sử dụng lệnh xuất trong C: **printf**

Lựa chọn tùy thuộc vào lập trình viên



7.2. Câu lệnh xuất cout (C++)

Thư viện: `#include <iostream>`

Cú pháp: `std::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=0;
std::cout<<"Chương trình xuất giá trị i \n";
std::cout<<"giá trị là "<<i;
```

```
Chương trình xuất giá trị i
10
```



7.2. Xuất văn bản thường (literal text)

Cú pháp: `std::cout<<"Chuoi can in";`

Ví dụ:

```
// Thư viện hỗ trợ nhập xuất
#include <iostream>
int main()
{
    // Xuất thông báo
    std::cout<<"Chao ban! Toi co the giup gi khong?";
    // Câu lệnh dùng màn hình kiểm tra kết quả
    std::cin.get();
    return 0;
}
```

Chao ban! Toi co the giup gi khong?



7.2. Ký tự điều khiển (escape sequence)

Các kí tự điều khiển:

"\n" // Xuống dòng

"\a" // Phát ra thông báo

"\b" // Lùi con trỏ 1 vị trí

"\t" // Dấu tab

"\" " // In kí tự "

"\\" // In kí tự \

Ví dụ:

```
#include <iostream>
int main()
{
    std::cout<<"Chao ban!\nToi co the giup gi khong?";
    std::cin.get();
    return 0;
}
```

Chao ban!

Toi co the giup gi khong?



7.2. Thiết lập độ rộng khi xuất

Cú pháp: **cout.width(n)** - Với 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.

Ví dụ:

```
#include <iostream>
int main()
{
    int a = 1706;
    std::cout<<a<<"\n";
    std::cout.width(10);
    std::cout<<a;
    std::cin.get();
    return 0;
}
```





7.3. Câu lệnh xuất printf (C)

Thư viện

#include <stdio.h> (standard input/output)

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 "", gồm:

- Văn bản thường (literal text)
- Ký tự điều khiển (escape sequence)
- Đặc tả (conversion specifier)



7.3. 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



7.3. 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

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");
```

[illegible]



7.3. 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);`



7.4. Câu lệnh nhập

Có 2 cách:

Cách 1: Sử dụng lệnh nhập trong C++: `cin`

Cách 2: Sử dụng nhập trong C: `scanf`

Lựa chọn tùy thuộc vào lập trình viên



7.5. Câu lệnh nhập `std::cin>>` (C++)

Thư viện:

```
#include <iostream>
```

Cú pháp:

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

Lưu ý:

Tham số ko có dạng chuỗi

Ví dụ:

```
#include <iostream>
int main()
{
    int namsinh = 0; //Buoc 1
    std::cout<<"Nam sinh: "; // Buoc 2
    std::cin>>namsinh; // Buoc 3;
    std::cin.get();
    return 0;
}
```



7.5. Chương trình cộng 2 số nguyên

```
#include <iostream>
int main()
{
    int a = 0, b = 0;
    std::cout<<"Chương trình cộng 2 số a, b"<<"\n";
    std::cout<<"a: ";
    std::cin>>a;
    std::cout<<"b: ";
    std::cin>>b;
    std::cout<<"a + b = "<<a + b<<"\n";
    std::cin.get();
    return 0;
}
```

Chương trình cộng 2 số a,b

a = 5

b = 6

a + b = 11

Viết chương trình tính $a + b$, $a - b$, $a * b$, a / b .



7.6. Câu lệnh nhập scanf (C)

Thư viện

- #include <stdio.h> (standard input/output)

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 &



7.6. Câu lệnh nhập scanf

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);`



8. Một số hàm hữu ích khác

Các hàm trong thư viện toán học

Thư viện: `#include <math.h>`

- 1 đối số đầu vào: **double**, trả kết quả: **double**
 - `acos`, `asin`, `atan`, `cos`, `sin`, ...
 - `exp`, `log`, `log10`
 - `sqrt`
 - `ceil`, `floor`
 - `abs`, `fabs`
- 2 đối số đầu vào: **double**, trả kết quả: **double**
 - `double pow(double x, double y)`



8. Chương trình ví dụ các hàm hữu ích

```
#include <iostream>
#include <math.h>
int main()
{
    float x = 2;
    std::cout<<cos(x)<<"\n"; // Hàm cos
    std::cout<<sin(x)<<"\n"; // Hàm sin
    std::cout<<tan(x)<<"\n"; // Hàm tan
    std::cout<<acos(x)<<"\n"; // Hàm arc cos
    std::cout<<asin(x)<<"\n"; // Hàm arc sin
    std::cout<<atan(x)<<"\n"; // Hàm arc tan
    std::cout<<log(x)<<"\n"; // Hàm log thường
    std::cout<<log10(x)<<"\n"; // Hàm log 10
    std::cout<<sqrt(x)<<"\n"; // Hàm căn bậc 2
    std::cout<<fabs(-x)<<"\n"; // lấy giá trị tuyệt đối
    std::cout<<pow(x,2); //Hàm mũ
    std::cin.get()
    return 0;
}
```

```
-0.416147
0.909297
-2.18504
nan
nan
1.10715
0.693147
0.30103
1.41421
2
4
```



9. Bài tập minh họa

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. $\text{thuế giá trị gia tăng} = 10\% \text{ tiền}$
4. Nhập bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.



9. Bài tập minh họa

1. Nhập năm sinh của một người và tính tuổi của người đó.

```
#include <iostream>
int main()
{
    int namsinh = 0;
    std::cout<<"Vui long nhap nam sinh: ";
    std::cin>>namsinh;
    std::cout<<"Ban " <<2016-namsinh<<" tuoi"<<"\n";
    std::cin.get();
    return 0;
}
```



9. Bài tập minh họa

2. Nhập 2 số a và b. Tính tổng, hiệu, tích và thương của hai số đó.

```
#include <iostream>
int main()
{
    int a, b;
    std::cout<<"Nhập a = ";
    std::cin>>a;
    std::cout<<"Nhập b = ";
    std::cin>>b;
    std::cout<<"a + b = "<<a+b<<"\n";
    std::cout<<"a - b = "<<a-b<<"\n";
    std::cout<<"a * b = "<<(long long)a*b<<"\n";
    std::cout<<"a / b = "<<(double)a/b<<"\n";
    std::cin.get();
    return 0;
}
```



9. Bài tập minh họa

3. Nhập tên sản phẩm, số lượng và đơn giá. Tính tiền và thuế giá trị gia tăng phải trả, biết:

- a. tiền = số lượng * đơn giá
- b. thuế giá trị gia tăng = 10% tiền

```
#include <iostream>
int main()
{
    int so_luong = 0, don_gia = 0;
    std::cout<<"Vui long nhap so luong: ";
    std::cin>>so_luong;
    std::cout<<"Vui long nhap don gia: ";
    std::cin>>don_gia;
    std::cout<<"Tien: "<<so_luong * don_gia<<"\n";
    std::cout<<"VAT: "<<so_luong * don_gia * 0.1<<"\n";
    std::cin.get();
    return 0;
}
```



9. Bài tập minh họa

4. Nhập bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.

```
#include <iostream>
#define PI 3.14
int main()
{
    float r = 0;
    std::cout<<"Nhập bán kính đường tròn: ";
    std::cin>>r;
    std::cout<<"Chu vi: "<<2 * PI * r<<"\n";
    std::cout<<"Diện tích: "<<PI * r * r<<"\n";
    return 0;
}
```

9. Bài tập về nhà



1. Cho số xe (gồm 4 chữ số) của bạn. Cho biết số xe của bạn được mấy nút?
2. Cho 1 ký tự chữ thường. In ra ký tự chữ hoa tương ứng.
3. Cho 3 số nguyên. Cho biết số lớn nhất và nhỏ nhất?
4. Viết chương trình cho 2 giờ (giờ, phút, giây) và thực hiện cộng, trừ 2 giờ này.
5. Tổng các bội số của 3 và 5 nhỏ hơn 10 là 23.
Ví dụ: Ta có các bội số: 3, 5, 6, 9 \rightarrow Tổng: 23
Tính tổng các bội số của 3 và 5 nhỏ hơn 1000.

- 5. Các phép toán
- 6. Biểu thức
- 7. Nhập xuất dữ liệu
- 8. Một số hàm hữu ích
- 9. Một số ví dụ minh họa

