


 **CHƯƠNG 2: BIỂU THỨC**

Giảng Viên: Nguyễn Văn Thắng

 **BIỂU THỨC**

- ✓ Biểu thức là một sự kết hợp giữa các **toán tử** (operator) và các **toán hạng** (operand) theo đúng một trật tự nhất định.
- ✓ Mỗi toán hạng có thể là một hằng, một biến hoặc một biểu thức khác.
- ✓ Trong trường hợp, biểu thức có nhiều toán tử, ta dùng cặp dấu ngoặc đơn () để chỉ định toán tử nào được thực hiện trước.

Giảng Viên: Nguyễn Văn Thắng

 **CÁC TOÁN TỬ**

1. Toán tử gán
2. Toán tử số học
3. Các toán tử gán phức hợp
4. Toán tử tăng giảm
5. Toán tử quan hệ
6. Toán tử logic
7. Toán tử điều kiện
8. Toán tử lấy kích thước
9. Chuyển đổi kiểu dữ liệu

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ GÁN(=)

- ✓ Toán tử gán dùng để gán một giá trị nào đó cho một biến
- Ví dụ:
 - $a=5$ // gán giá trị nguyên 5 cho biến a
- ✓ Về trái của phép gán bắt buộc phải là một biến, về phải có thể là hằng, biến hay kết quả của một biểu thức.

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ GÁN(=)

- ✓ Toán tử gán luôn được thực hiện từ trái sang phải
- Ví dụ: $a = b;$ // gán giá trị của biến **a** bằng giá trị đang chứa trong biến **b**
- ✓ Cho phép về phải có thể chứa các phép gán khác.
- Ví dụ: $a = 2 + (b = 5);$
 - tương đương với
 - $b = 5;$
 - $a = 2 + b;$

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ SỐ HỌC

Toán tử	Tên	Ví dụ
+	Cộng	$12 + 4.9$ // kết quả 16.9
-	Trừ	$3.98 - 4$ // kết quả -0.02
*	Nhân	$2 * 3.4$ // kết quả 6.8
/	Chia	$9 / 2.0$ // kết quả 4.5
%	Lấy phần dư	$13 \% 3$ // kết quả 1

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ SỐ HỌC

✓ Trừ toán tử lấy phần dư (%) thì tất cả các toán tử số học có thể chấp nhận pha trộn các toán hạng số nguyên và toán hạng số thực.

- Nếu cả hai toán hạng là số nguyên thì kết quả sẽ là một số nguyên.
- Một hoặc cả hai toán hạng là số thực thì kết quả sẽ là một số thực (real hay double)

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ GÁN PHỨC HỢP

Toán Tử	Ví dụ	Tương đương với
+=	n += 25	n = n + 25
-=	n -= 25	n = n - 25
*=	n *= 25	n = n * 25
/=	n /= 25	n = n / 25
%=	n %= 25	n = n % 25

Giảng Viên: Nguyễn Văn Thắng

Ví dụ:

```
#include <iostream.h>
int main ()
{
    int a, b=3;
    a = b;
    a+=2;    // tương đương với a=a+2
    cout << a;
    return 0;
}
```

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ TĂNG - GIẢM

- Toán tử tăng (++) và toán tử giảm (--) có tác dụng làm tăng hoặc giảm 1 giá trị lưu trong biến. Chúng tương đương với +=1 hoặc -=1. Vì vậy, các dòng sau là tương đương:
a++;
a+=1;
a=a+1;

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ TĂNG - GIẢM

- Toán tử tăng/giảm có 2 dạng:
 - Tiền tố : ++a
 - Hậu tố : a++

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ TĂNG - GIẢM

- Trường hợp toán tử được sử dụng như là một tiền tố ++a: giá trị của a tăng trước khi biểu thức được tính và giá trị đã tăng được sử dụng trong biểu thức
- Trong trường hợp toán tử được sử dụng như là một hậu tố a++ : giá trị trong biến a được tăng sau khi đã tính toán.

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ TĂNG - GIẢM

Ví dụ 1:
 $B=3;$
 $A=++B;$
Kết quả: *A chứa giá trị 4, B chứa giá trị 4*

Ví dụ 2:
 $B=3;$
 $A=B++;$
Kết quả: *A chứa giá trị 3, B chứa giá trị 4*

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ QUAN HỆ

Toán tử	Tên	Ví dụ
$==$	So sánh bằng	$5 == 5$ // kết quả 1
$!=$	So sánh không bằng	$5 != 5$ // kết quả 0
$<$	So sánh nhỏ hơn	$5 < 5.5$ // kết quả 1
$<=$	So sánh nhỏ hơn hoặc bằng	$5 <= 5$ // kết quả 1
$>$	So sánh lớn hơn	$5 > 5.5$ // kết quả 0
$>=$	So sánh lớn hơn hoặc bằng	$6.3 >= 5$ //kết quả1

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ QUAN HỆ

✓ Các toán hạng của một toán tử quan hệ phải ước lượng về một số. Các ký tự là các toán hạng hợp lệ vì chúng được đại diện bởi các giá trị số. Ví dụ (giả sử mã ASCII):

$'A' < 'F'$ // kết quả là 1
 (tương đương với $65 < 70$)

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ LOGIC

Toán tử	Ý nghĩa	Ví dụ
!	Phủ định	!(5 == 5) // kết quả là 0
&&	và	5 < 6 && 6 < 6 // kết quả là 0
	Hoặc	5 < 6 6 < 5 // kết quả là 1

0: SAI (false)

Khác 0: ĐÚNG (true)

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ ĐIỀU KIỆN

- **Cú pháp** E1 ? E2 : E3

Trong đó E1, E2, E3 là các biểu thức.

- **Ý nghĩa:**
 Nếu E1 đúng thì E2 được ước lượng và nó trở thành giá trị của biểu thức.
 Ngược lại, nếu E1 sai, E3 được ước lượng và trở thành giá trị của biểu thức.

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ ĐIỀU KIỆN

- Ví dụ:

$$X = 10$$

$$Y = \underset{\substack{\uparrow \\ \text{E1}}}{X} > \underset{\substack{\uparrow \\ \text{E2}}}{9} ? \underset{\substack{\uparrow \\ \text{E3}}}{100} : 200$$

Nếu $X > 9$ thì Y được gán giá trị 100, ngược lại, nếu X nhỏ hơn 9 thì Y sẽ nhận giá trị là 200.

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ ĐIỀU KIỆN

- Phép toán điều kiện cũng là một biểu thức nên nó có thể được sử dụng như một toán hạng của phép toán điều kiện khác.
- Ví dụ:

```
int m = 1, n = 2, p = 3;
int min = (m < n ? (m < p ? m : p) : (n < p ? n : p));
```

↑
E1
↑
E2
↑
E3

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ ĐIỀU KIỆN(ví dụ)

```
#include <iostream>
int main ()
{
    int a,b,c;
    a=2;
    b=7;
    c = (a>b) ? a : b;
    cout << c;
    return 0;
}
```

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ LẤY KÍCH THƯỚC

- **sizeof(data_Type)** : để tính kích thước của bất kỳ hạng mục dữ liệu hay kiểu dữ liệu nào.
- Kết quả trả về là kích thước của kiểu dữ liệu đã chỉ định theo byte.
- Ví dụ:

```
cout << "char size = " << sizeof(char) << " bytes\n";
```

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ BITWISE

Toán tử	Tên	Ví dụ
~	Phủ Định Bit	~\011' // được \366'
&	Và bit	\011' & \027' // được \001'
	Hoặc bit	\011' \027' // được \037'
^	Hoặc exclusive bit	\011' ^ \027' // được \036'
<<	Dịch trái bit	\011' << 2 // được \044'
>>	Dịch phải bit	\011' >> 2 // được \002'

Giảng Viên: Nguyễn Văn Thắng

TOÁN TỬ BITWISE

Ví dụ :

105 & 7 = 1 // 01101001 & 0000 0111 = 00000001

105 | 7 = 127 // 01101001 | 0000 0111 = 01101111

0x60 = 0x96 /* 0110 1001 = 1001 0110 */

Giảng Viên: Nguyễn Văn Thắng

ĐỘ ƯU TIÊN CỦA CÁC TOÁN TỬ

TOÁN TỬ	ĐỘ ƯU TIÊN	TOÁN TỬ	ĐỘ ƯU TIÊN
()	1	&	8
!, ++, --, sizeof()	2	^	9
*, /, %	3		10
+, -	4	&&	11
<<, >>	5		12
<, <=, >, >=	6	? :	13
=, !=	7	=, +=, -=, *=, /=	14

Giảng Viên: Nguyễn Văn Thắng

CHUYÊN ĐỔI KIỂU DỮ LIỆU

- Phép chuyển đổi kiểu cho phép chuyển đổi dữ liệu từ kiểu này sang kiểu khác.
- Cách cơ bản nhất được thừa kế từ ngôn ngữ C là :
(Kiểu dữ liệu)<biểu thức>

Ví dụ:

Giảng Viên: Nguyễn Văn Thắng

CHUYÊN ĐỔI KIỂU DỮ LIỆU

- (int) 3.14 // chuyển 3.14 sang int để được 3
- (long) 3.14 // chuyển 3.14 sang long để được 3L
- (double) 2 // chuyển 2 sang double để được 2.0
- (char) 122 // chuyển 122 sang char có mã là 122
- (unsigned short) 3.14 // được 3 như là một unsigned short

Giảng Viên: Nguyễn Văn Thắng

BÀI TẬP CHƯƠNG 2

1. Thêm các dấu ngoặc phụ vào các biểu thức sau để hiển thị rõ ràng thứ tự các toán tử được ước lượng:

- $(n \leq p + q \ \&\& \ n \geq p - q \ || \ n == 0)$
- $(++n * q-- / ++p - q)$
- $(n | p \ \& \ q \ \wedge \ p \ll 2 + q)$
- $(p < q ? n < p ? q * n - 2 : q / n + 1 : q - n)$

Giảng Viên: Nguyễn Văn Thắng

BÀI TẬP CHƯƠNG 2

2. Cho biết giá trị của mỗi biến sau đây sau khi khởi tạo nó:

- `double d = 2 * int(3.14);`
- `long k = 3.14 - 3;`
- `char c = 'a' + 2;`
- `char c = 'p' + 'A' - 'a';`

3. Viết một chương trình cho phép nhập vào một số nguyên dương n và xuất ra giá trị của n mũ 2 và 2 mũ n .

Giảng Viên: Nguyễn Văn Thắng

BÀI TẬP CHƯƠNG 2

Bài 3:

- Nhập bán kính đường tròn r . Tính và xuất chu vi, diện tích đường tròn tương ứng.
- (Hướng dẫn: `cv=2*p*r` và `dt=p*r2`
- - Dùng `p=3.14`,
- - hoặc khai báo hằng `PI`,
- - hoặc dùng hằng `M_PI` trong thư viện `<math.h>` của ngôn ngữ lập trình)

Giảng Viên: Nguyễn Văn Thắng

BÀI TẬP CHƯƠNG 2

Bài 4:

4.1) Nhập cạnh a . Tính và xuất chu vi, diện tích hình vuông.
HD: `cv=4*a` và `dt=a2`

4.2) Nhập cạnh a, b . Tính và xuất chu vi, diện tích hình chữ nhật.
HD: `cv=2*(a+b)` và `dt=a*b`


4.3) Nhập cạnh a, h_1, h_2 . Tính và xuất chu vi, diện tích hình thoi.
HD: `cv=4*a` và `dt=1.0/2*h1*h2`

4.4) Nhập cạnh a, b, c, d, h . Tính và xuất chu vi, diện tích hình thang.
HD: `cv=(a+b+c+d)` và `dt=1.0/2*h*(a+b)`

Giảng Viên: Nguyễn Văn Thắng

BÀI TẬP CHƯƠNG 2

Bài 5:



Nhập cạnh a,b,c. Tính và xuất chu vi, diện tích hình tam giác.

HD: $cv=a+b+c$ và $dt=$ với $p=cv/2$

Giảng Viên: Nguyễn Văn Thắng

BÀI TẬP CHƯƠNG 2

- Bài 6:

Nhập 1 góc x. Tính và xuất $\sin x$, $\cos x$, $\tan x$, $\cot x$.

HD: các hàm sin, cos, tan chỉ tính theo đơn vị radian nên chúng ta phải đổi từ độ x sang độ radian t như sau: $t=x*\pi/180 \Rightarrow$
 $\sin x=\sin(t)$, $\cos x=\cos(t)$, $\tan x=\tan(t)$,
 $\cot x=1/\tan x$

Giảng Viên: Nguyễn Văn Thắng

CHƯƠNG 3: CẤU TRÚC ĐIỀU KHIỂN

Giảng Viên: Nguyễn Văn Thắng

LỆNH

- Một chương trình đang chạy nghĩa là nó đang thực thi các câu **lệnh**. Thứ tự mà các câu lệnh được thực hiện được gọi là **dòng điều khiển** (flow control).
- Các câu lệnh đang thực thi có **sự điều khiển** của CPU, khi CPU hoàn thành sẽ được chuyển giao tới một lệnh khác.
- Đặc trưng dòng điều khiển trong một chương trình là **tuần tự**.

Giảng Viên: Nguyễn Văn Thắng

LỆNH

- Dòng điều khiển xem xét và quyết định lệnh nào được thực thi và lệnh nào không được thực thi trong quá trình chạy, vì thế làm ảnh hưởng đến kết quả toàn bộ của chương trình
- C++ cung cấp những hình thức lệnh khác nhau cho các mục đích khác nhau.

Giảng Viên: Nguyễn Văn Thắng

LỆNH

- **Lệnh khai báo** được sử dụng cho định nghĩa các biến
- **Lệnh gán** được sử dụng cho các tính toán đại số đơn giản
- **Lệnh rẽ nhánh** được sử dụng để chỉ định đường dẫn của việc thực thi phụ thuộc vào kết quả của một điều kiện luận lý (True, False)

Giảng Viên: Nguyễn Văn Thắng

LỆNH

- **Lệnh lặp** được sử dụng để chỉ định các tính toán cần được lặp cho tới khi một điều kiện luận lý nào đó được thỏa.
- Các lệnh điều khiển được sử dụng để làm chuyển đường dẫn thực thi tới một đường dẫn khác của chương trình.

Giảng Viên: Nguyễn Văn Thắng

LỆNH

- **Lệnh đơn** là một sự tính toán được kết thúc bằng dấu chấm phẩy(;)
- **Lệnh phức**: Nhiều lệnh đơn có thể kết nối lại thành một lệnh phức và được đặt trong cặp dấu {}

Giảng Viên: Nguyễn Văn Thắng

Cấu trúc điều kiện: if và else

- **Dạng 1:**
Cú pháp:
if (biểu thức)
lệnh;
- Nếu *biểu thức* được ước lượng với kết quả khác 0 (đúng) thì *lệnh* được thực thi.
- Ngược lại, không làm gì cả.

Giảng Viên: Nguyễn Văn Thắng

Cấu trúc điều kiện: *if* và *else*

Ví dụ: khi chia hai giá trị chúng ta muốn kiểm tra điều kiện mẫu số có khác 0 hay không.

```
void main()
{
    if (count != 0)
        average = sum / count;
}
```

Giảng Viên: Nguyễn Văn Thắng

Cấu trúc điều kiện: *if* và *else*

- Dạng 2: cấu trúc if cho phép chúng ta chọn một trong hai lệnh:
- Cú pháp:
if (biểu thức)
lệnh 1;
else
lệnh 2;

Giảng Viên: Nguyễn Văn Thắng

Cấu trúc điều kiện: *if* và *else*

- Nếu điều kiện thỏa thì thực hiện lệnh 1
- Nếu điều kiện không thỏa thì thực hiện lệnh 2
- Ví dụ:

Giảng Viên: Nguyễn Văn Thắng



Cấu trúc điều kiện: *if* và *else*

```
if (balance > 0) {  
    interest = balance * creditRate;  
    balance += interest;  
}  
else {  
    interest = balance * debitRate;  
    balance += interest;  
}
```

Giảng Viên: Nguyễn Văn Thắng
