



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



C PROGRAMMING



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

BIỂU THỨC TRONG NGÔN NGỮ C

ONE LOVE. ONE FUTURE.

- Kết hợp các giá trị bằng toán tử hoặc gọi hàm để tạo giá trị mới
- Giá trị trả về luôn có kiểu xác định
- Các toán tử có thể được phân loại
 - theo ngôi (một hoặc hai toán hạng)
 - theo chức năng (toán học, logic, ...)

Biểu thức toán học

- Có toán hạng là giá trị số học và
- trả về là một giá trị số học
- Được xây dựng bởi các toán tử sau:
 - + (toán tử dương)
 - (toán tử âm)
 - + (cộng)
 - (trừ)
 - * (nhân)
 - / (chia)
 - % (lấy số dư)

Ví dụ biểu thức

$$1 + 2 * 3 - 4 / 5$$
$$= 1 + (2 * 3) - (4 / 5)$$


Ví dụ(tiếp)

$$1 + 2 * 3 - 4 / 5 =$$

$$1 + (2 * 3) - (4 / 5)$$

Chia hai số nguyên cho
kết quả vẫn là một số
nguyên



Ví dụ (tiếp)

$$1 + 2 * 3 - 4 / 5 =$$

$$1 + (2 * 3) - (4 / 5)$$

$$= 0$$



Ví dụ (tiếp)

- Dùng số thực để tạo biểu thức có kết quả là số thực

$$1 + 2 * 3 - 4.0 / 5$$

$$= 1 + (2 * 3) - (4.0 / 5)$$

$$= 1 + 6 - 0.8$$

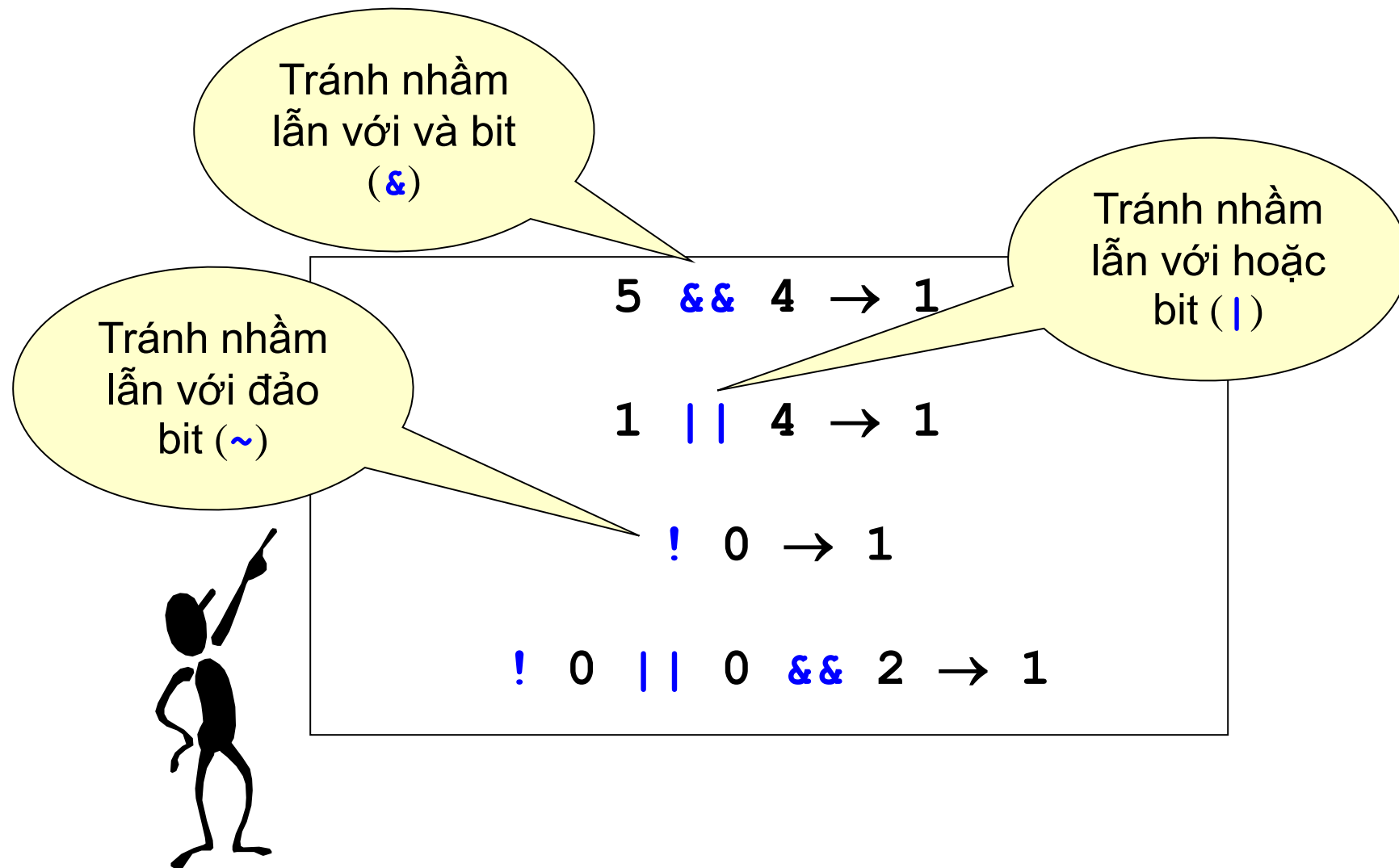
$$= 6.2$$

- Một “kiểu” đặc biệt chỉ có hai giá trị:
 - Đúng (**true**)
 - Sai (**false**)
- Dùng để tạo điều kiện cho lựa chọn hoặc vòng lặp trong một giải thuật
- Biểu thức lôgic
 - Là biểu thức mà chỉ trả về giá trị đúng/sai

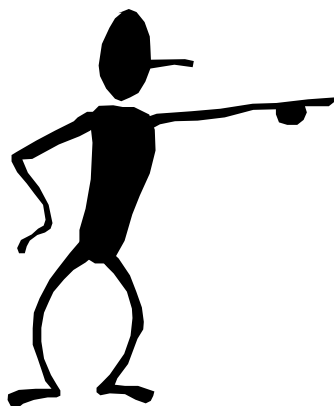
Dùng int như logic

- Trong C, các giá trị logic được biểu diễn bằng số nguyên
 - Giá trị nguyên 0 là sai.
 - Giá trị nguyên khác 0 là đúng (thường dùng số 1)
- Mọi biểu thức trong C đều trả về kết quả số
- Một biểu thức logic được đánh giá là đúng sẽ trả về giá trị 1, ngược lại là 0.

- ...dùng để xây dựng các biểu thức logic
- Và (&&)
- Hoặc (||)
- Phủ định (!)
- So sánh bằng (==)
- So sánh khác (!=)
- So sánh lớn bé (<, >, <=, >=)

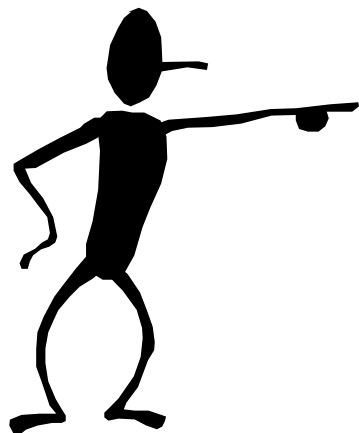


Một biểu thức chỉ sử dụng các toán tử trên bit không phải là biểu thức lôgic. Kết quả của biểu thức này là một số nguyên bất kì


$$5 \ \& \ 4 \rightarrow 4$$
$$1 \ | \ 4 \rightarrow 5$$
$$\sim 0 \rightarrow 0\text{xFFFF}$$

“Ngắn mạch”

Không nhất thiết phải sử dụng tất cả các toán hạng để đánh giá một biểu thức logic, chỉ cần đánh đến khi giá trị biểu thức được xác lập



1 || 2

0 || 1 || 2

1 && 0 && -1

Ví dụ toán tử so sánh

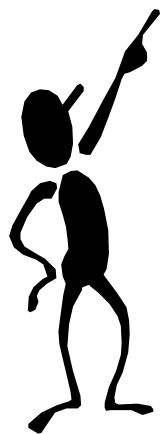
Tránh nhầm
lẫn với phép
gán (=)

$3 == 4 \rightarrow 0$

$3 != 4 \rightarrow 1$

$3 < 4 \rightarrow 1$

$3 < 4 \ \&\& \ 5 > 2 \rightarrow 1$



Thứ tự ưu tiên

- Phủ định và đảo dấu (!, -)
- Nhân chia (*, /, %)
- Cộng trừ (+, -)
- So sánh 1 (<, <=, >, >=)
- So sánh 2 (==, !=)
- Và (&&)
- Hoặc (||)

Lỗi thường gặp

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

```
/* Loi thuong gap */
```

```
int main()
```

```
{
```

```
    int score;
```

```
    scanf("%d", &score);
```

```
    if ( score == 9 || 10 )
```

```
    {
```

```
        printf("Xuat sac\n");
```

```
    }
```

```
    return 0;
```

```
}
```

Giá trị trả về
luôn là 1

Giá trị trả về
là 0 hoặc 1

Lỗi thường gặp (tiếp)

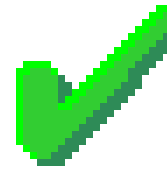
```
#include <stdio.h>

/* Chương trình dung */

int main()
{
    int score;

    scanf("%d", &score);

    if ( score == 9 || score == 10 )
    {
        printf("Xuat sac\n");
    }
    return 0;
}
```



Lỗi thường gặp (tiếp)

```
#include <stdio.h>

/* Loi thuong gap */

int main()
{
    int score;

    scanf("%d", &score);

    if ( 8 <= score <= 10 )
    {
        printf("Giai\n");
    }
    return 0;
}
```

Giá trị trả về
luôn là 1

Giá trị trả về là
0 hoặc 1

- Gán (=) cũng là một toán tử. Vì vậy sử dụng toán tử này tạo ra một biểu thức và trả về giá trị
- Kết quả của biểu thức gán là giá trị bên phải của biểu thức
- Ví dụ:
 $(x = 4) \rightarrow 4$
 $(y = 0) \rightarrow 0$
- Có thể tạo một biểu thức với một chuỗi toán tử gán
 $x = y = z = 4$

Lỗi thường gặp (tiếp)

```
#include <stdio.h>

/* Loi gap nhieu trong C */

int main()
{
    int score;

    scanf("%d", &score);

    if (score = 9 || score = 10)
    {
        printf("Gioi!\n");
    }
    return 0;
}
```

Nhầm với
phép gán

Lỗi thường gặp (tiếp)

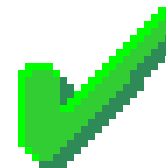
```
#include <stdio.h>

/* Probably the most common C error. */

int main()
{
    int score;

    scanf("%d", &score);

    if (score == 9 || score == 10)
    {
        printf("OK!\n");
    }
    return 0;
}
```



Một số toán tử gán mở rộng

Toán tử	Ví dụ	Biểu thức tương đương
$+=$	$x += 5$	$x = x + 5$
$-=$	$x -= 5$	$x = x - 5$
$*=$	$x *= 5$	$x = x * 5$
$/=$	$x /= 5$	$x = x / 5$
$\%=$	$x \% = 5$	$x = x \% 5$

Toán tử tăng giảm

- `++` là toán tử tăng
`++i` tương đương với $i = i + 1$
- `--` là toán tử giảm
`--j` tương đương với $j = j - 1$
- Có hai dạng viết là tiền tố (`++i`) và hậu tố (`i++`)
- Chúng khác nhau ở giá trị trả về của biểu thức.
- Ví dụ nếu $i = 5$
 - Tiền tố trả về giá trị sau khi đã cộng, $(++i) \rightarrow 6$
 - Hậu tố trả về giá trị trước khi cộng, $(i++) \rightarrow 5$
 - Trong cả hai trường hợp giá trị của i đều tăng lên 1

- Chúng ta có thể viết một biểu thức mà giá trị của nó phụ thuộc vào một điều kiện

<Điều kiện> ? <Biểu thức 1> : <Biểu thức 2>

– Viết biểu thức tính max của a và b

`int max,a,b;`

`...`

`max = (a > b) ? a : b;`

- Phép gán chỉ thực hiện trên biến và giá trị có cùng kiểu
- C có thể thực hiện chuyển kiểu tự động cho phép gán nếu sự chuyển kiểu đó không làm mất thông tin.
- Ví dụ chuyển từ int về float

```
int a;
```

```
float f;
```

```
f = a; /* OK */
```

```
a = f; /* KO */
```

- Trong trường hợp bị mất thông tin ta phải thực hiện ép kiểu.
- Ví dụ nếu chuyển từ float về int
a = **(int)** f;

Tổng hợp thứ tự ưu tiên

Toán tử	Chiều tính toán
-, ++, --, !, ~	<<<<<
*, /, %	>>>>>
+, -	>>>>>
<<, >>	>>>>>
<, <=, >, >=	>>>>>
==, !=	>>>>>
&	>>>>>
	>>>>>
&&	>>>>>
	>>>>>
?:	<<<<<
=, +=, -=, *=, /=, %=, &=, =, <<=, >>=	<<<<<

$$7+5 \& \& 4 < 2+3-2/3 || 5 > 2+1$$

$$(7+5) \& \& 4 < 2+3-(2/3) || 5 > (2+1)$$

$$12 \& \& 4 < (2+3-0) || (5 > 3)$$

$$12 \& \& (4 < 5) || 1$$

$$(12 \& \& 1) || 1$$

$$1 || 1 = 1$$

A graphic on the left side of the slide. It features a dark blue background with a large, stylized circular shape composed of many small red dots. The dots are arranged in a way that creates a sense of depth and movement, with some dots appearing larger and more concentrated than others. In the center of this circular graphic, the word "HUST" is written in a bold, white, sans-serif font.

HUST

THANK YOU !