

CONFIDENTIAL

# C Programming Introduction

## Week 5: Expressions

For HEDSPI Project

## Phép toán và biểu thức

- Các phép toán

- Cộng +
- Trừ -
- Nhân \*
- Chia /
- Lấy phần dư %

- Ví dụ

- `fag = x % y;`
- `c = a - (a/b)*b;`
- `sum = var1 + var2 + var3;`

## Phép toán và biểu thức

- Thứ tự ưu tiên

- Một số phép toán có thứ tự ưu tiên cao hơn các phép toán khác, cần sử dụng dấu ngoặc () khi cần.

- Thứ tự:

- ()
- \* / \*
- + -

- Ví dụ: `(a + b + c) / 3`

## Câu lệnh so sánh

- Các phép toán quan hệ

- Nhỏ hơn < `a < 5`
- Nhỏ hơn hoặc bằng <= `a <= b`
- Lớn hơn > `a > b + c`
- Lớn hơn hoặc bằng >= `a >= b + 5`
- Bằng == `a == -6`
- Khác != `a != 0`

• Using if statements, relational Example 1

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int num1, num2;
6
7     printf( "Enter two integers, and I will tell you\n" );
8     printf( "the relationships they satisfy: " );
9     scanf( "%d%d", &num1, &num2 ); /* read two integers */
10
11     if ( num1 == num2 )
12         printf( "%d is equal to %d\n", num1, num2 );
13
14     if ( num1 != num2 )
15         printf( "%d is not equal to %d\n", num1, num2 );
16
17     if ( num1 < num2 )
18         printf( "%d is less than %d\n", num1, num2 );
19
20     if ( num1 > num2 )
21         printf( "%d is greater than %d\n", num1, num2 );
22
23     if ( num1 <= num2 )
24         printf( "%d is less than or equal to %d\n",
25               num1, num2 );
26
```

```
27
28 if ( num1 >= num2 )
29     printf( "%d is greater than or equal to %d\n",
30           num1, num2 );
31
32 return 0; /* indicate program ended successfully */
33
```

Enter two integers, and I will tell you  
the relationships they satisfy: 3 7  
3 is not equal to 7  
3 is less than 7  
3 is less than or equal to 7

Enter two integers, and I will tell you  
the relationships they satisfy: 22 12  
22 is not equal to 12  
22 is greater than 12  
22 is greater than or equal to 12

## Biểu thức và phép toán

- Các biểu thức logic

- AND **&&** (a > 0) && (b > 0)
- OR **||** (a <= 0) || (b <= 0)
- Phủ định **!** !(a && c)

## Biểu thức và phép toán

- Các phép toán dịch bit (Bitwise Operators)

- Bitwise AND **&**
- Bitwise OR (Inclusive OR) **|**
- Bitwise XOR (Exclusive OR) **^**
- Dịch trái **<<**
- Dịch phải **>>**
- Bù 1 **~**

- Ví dụ

```
x = 01001011    y = 00101100
~x = 10110100
x & y = 00001000    x | y = 01101111
x ^ y = 01100111    x << 2 = 00101100
```

## Biểu thức và phép toán

- Op là + - \* / % << >> & ^ |
- **expr1**, **expr2** là biến

**expr1 op= expr2**

Tương đương với

**expr1 = (expr1) op (expr2)**

- Ví dụ

- **x += 1;**
  - **x = x + 1;**
- } Giống nhau

## Biểu thức và phép toán

- Biểu thức điều kiện

**expr1 ? expr2 : expr3**

- If **expr1** is *true* do **expr2**
- If **expr1** is *false* do **expr3**

- Ví dụ

```
- a = 5;
- b = 10;
- min = a < b ? a : b;
```

## Biểu thức và phép toán

- Các phép tăng và giảm

- Tăng trước khi dùng **++variable**
- Tăng sau khi dùng **variable++**
- Giảm trước khi dùng **--variable**
- Giảm sau khi dùng **variable--**

- Ví dụ

```
- x = 4;
- y = x++ + 5;    // x = 5, y = 9
- x = 4;
- y = ++x + 5;    // x = 5, y = 10
```

## Biểu thức và phép toán

- Phép ép kiểu

**(type-specifier) expression;**

- Ví dụ

```
- (double) date;
- float var1 = 2.7;
- int var2 = (int) var1;    //var2 = 7
- (char) x;
- (int) d1 + d2;
```

## Bài tập 5.1

- Viết chương trình chuyển đổi khoảng cách từ km sang miles.
- Nhập từ bàn phím giá trị km, đưa ra màn hình giá trị tương ứng = miles.

## Bài 5.2

- Chạy bài 5\_2.c dưới để thấy ý nghĩa của các phép toán logic và quan hệ.
- Thay  $b - a == b - c$  bằng  $a = b - c$ , sau đó giải thích kết quả.

## Bài 5\_2.c

```
#include <stdio.h>

main()
{
    int a = 5, b = 6, c = 7;
    puts("int a = 5, b = 6, c = 7;\n");

    printf("The value of a > b is    \t%i\n\n", a > b);
    printf("The value of b < c is    \t%i\n\n", b < c);

    printf("The value of a + b >= c is \t%i\n\n", a + b >= c);
    printf("The value of a - b <= b-c is\t%i\n\n", a - b <= b - c);

    printf("The value of b - a == b - c is\t%i\n\n", b - a == b - c);
    printf("The value of a * b != c * c is\t%i\n\n", a * b != c * c);
}
```

## Bài 5.3

- Gỡ và dịch bài exercise5\_3.c sau, chương trình minh họa các biểu thức điều kiện.
- Biến đổi chương trình bằng cách bỏ đi các biến abs và max.

## exercise5\_3.c

```
#include <stdio.h>
main()
{
    int n, m, abs, max;

    printf("Enter a positive or negative integer: ");
    scanf("%i", &n);

    printf("\nYou entered %i.\n", n);
    abs = n < 0 ? -n : n;
    printf("Its absolute value is %i.\n", abs);

    printf("\nEnter two integers (e.g. 1 2): ");
    scanf("%i %i", &n, &m);

    printf("\nYou entered %i and %i.\n", n, m);
    max = n > m ? n : m;
    printf("%i is the larger value.\n", max);
}
```

## Bài 5.4

- Bài này minh họa hiện tượng tràn số nguyên xảy ra khi phép toán số học cố tạo ra giá trị số lớn hơn khả năng biểu diễn của nó.
- Gỡ và chạy chương trình để thấy kết quả.

## exercise5\_4.c

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

void main(void)
{
    unsigned int x = UINT_MAX - 1;
    signed int y = INT_MAX - 1;

    printf("x is an unsigned int, occupying %i bytes.\n\n",
        sizeof(x));

    printf("The initial value of x is %u\n", x);
    x++;
    printf("Add 1; the new value of x is %u\n", x);
    x++;
```

## exercise5\_4.c

```
printf("Add 1; the new value of x is %u\n", x);
x++;
printf("Add 1; the new value of x is %u\n", x);

printf("\ny is a signed int, occupying %i bytes.\n\n",
    sizeof(y));

printf("The initial value of y is %i\n", y);
y++;
printf("Add 1; the new value of y is %i\n", y);
y++;
printf("Add 1; the new value of y is %i\n", y);
y++;
printf("Add 1; the new value of y is %i\n", y);

return;
}
```

## Bài 5.5

- Viết chương trình yêu cầu người dùng nhập 2 giá trị double lưu trong 2 biến x,y.
- Sử dụng câu lệnh if để kiểm tra các mối quan hệ giữa x và y.

## Bài 5.6

- 1 nhóm n sinh viên được chia thành 7 nhóm, càng cân bằng càng tốt. (Không có nhóm nào > 2 sinh viên so với các nhóm còn lại). Viết chương trình hiển thị:
  - Số lượng sinh viên trong nhóm nhỏ nhất
  - Số lượng sinh viên trong nhóm lớn nhất
  - Số lượng sinh viên trung bình trong mỗi nhóm
  - Số nhóm trên kích thước trung bình
  - Số nhóm <= kích thước trung bình
  - Số sinh viên trong các nhóm có kích thước >= kích thước trung bình
  - Số sinh viên trong các nhóm có kích thước = kích thước trung bình