

# Chương 2

## Sử dụng những kiểu dữ liệu cơ sở trong chương trình

### Phần c: Các kiểu dữ liệu cơ sở và phép toán

**Nhập môn lập trình**

Trình bày: Nguyễn Sơn Hoàng Quốc

Email: [nshquoc@fit.hcmus.edu.vn](mailto:nshquoc@fit.hcmus.edu.vn)

# Nội dung

- Các kiểu dữ liệu cơ sở và phép toán
  - Kiểu số nguyên
    - Mô hình bộ nhớ
  - Kiểu số thực
  - Kiểu luận lý
  - Kiểu ký tự

# CÁC KIỂU DỮ LIỆU CƠ SỞ VÀ PHÉP TOÁN

cuu duong than cong . com

# Kiểu dữ liệu cơ sở và phép toán

- Dùng để thực hiện các tính toán và xây dựng những kiểu dữ liệu phức tạp hơn.
- Các kiểu dữ liệu bao gồm kiểu
  - kiểu số nguyên (có dấu và không dấu)
  - kiểu số thực
  - kiểu luận lý
  - kiểu ký tự

# Kiểu Số Nguyên

cuu duong than cong . com

cuu duong than cong . com

# Kiểu số nguyên có dấu

- Miền giá trị (số n-bit):  $-(2^{n-1}) \dots +(2^{n-1}-1)$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
char	1	-128 ... +127
int	2	-32.768 ... +32.767
	4	-2.147.483.648 ... +2.147.483.647
short	2	-32.768 ... +32.767
long	4	-2.147.483.648 ... +2.147.483.647
long long	8	-9,223,372,036,854,775,808 ... 9,223,372,036,854,775,807
<i>Một số môi trường lập trình đồng nhất kiểu <b>long long</b> với kiểu <b>long</b> cho nên kiểu này ít được sử dụng trong lập trình ứng dụng.</i>		

# Kiểu số nguyên không dấu

- Miền giá trị (số n-bit):  $0 \dots 2^n - 1$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
<code>unsigned char</code>	1	0 ... 255
<code>unsigned int</code>	2 4	0 ... 65535 0 ... 4.294.967.295
<code>unsigned short</code>	2	0 ... 65535
<code>unsigned long</code>	4	0 ... 4.294.967.295
<code>unsigned long long</code>	8	0 ... 18,446,744,073,709,551,615
<i>Một số môi trường lập trình đồng nhất kiểu <code>unsigned long long</code> với kiểu <code>unsigned long</code> cho nên kiểu này ít được sử dụng trong lập trình ứng dụng.</i>		

# Kiểu số nguyên

- Hằng số nguyên có thể biểu diễn ở 3 dạng
  - **Bát phân**: viết bắt đầu bằng số 0
  - **Thập phân**: viết bắt đầu bằng số từ 1 đến 9
  - **Thập lục phân**: viết đầu bằng 0x
- Ví dụ:
  - `int a = 1506;` // 1506<sub>10</sub>
  - `int b = 01506;` // 1506<sub>8</sub>
  - `int c = 0x1506;` // 1506<sub>16</sub> (0x hay 0X)

## Ví dụ minh họa

- Các trường hợp sau nên dùng biến số nguyên có dấu hay không dấu? Hãy khai báo nếu là số nguyên.
  - **Tính tuổi** của một sinh viên
  - **Tính điểm trung bình** của một sinh viên
  - **Tính lương** cho một nhân viên
  - **Tính tiền** của một căn nhà
  - **Lưu nhiệt độ** của một thành phố

# Kiểu số nguyên

- Các phép toán số học
  - Phép **cộng**:  $+$
  - Phép **trừ**:  $-$
  - Phép **nhân**:  $*$
  - Phép **chia** lấy phần nguyên:  $/$
  - Phép **chia** lấy phần dư:  $\%$
- Ví dụ (với  $a, b$  là hai kiểu số nguyên)
  - $2 + 3, a / 5, (a + b) * 5, \dots$

# Phép gán

- Việc tính toán trong chương trình được thực hiện bằng cách tính toán và chép kết quả tính toán vào một biến nằm bên trái của phép gán.

- Ví dụ:

`sum = a + b; // chép tổng a + b vào biến sum`

`sum = a + 2; // chép tổng a + 2 vào biến sum`

`sum = a + n; // chép tổng a + n vào biến sum`

# Lệnh viết ngắn

- Ví dụ:
  - Viết `sum++` (hay `++sum`) thay cho `sum = sum + 1;`
  - Viết `sum += 2` thay cho `sum = sum + 2;`
  - Viết `sum += n` thay cho `sum = sum + n;`
  - Viết `n = m++` tương đương với `n = m;` rồi `m++;`
  - Viết `n = ++m` tương đương với `++m` rồi `n = m;`
- Việc viết các lệnh cô đọng có thể làm cho chương trình khó đọc, khó bắt lỗi vì vậy không nên lạm dụng!

## Ví dụ minh họa

- Xác định khai báo các biến cần thiết và tính
  - Bình phương của một số  $x$  được khai báo bằng 2
  - Diện tích của hình chữ nhật biết độ dài cạnh là những số nguyên tương ứng là 3 và 4

[cuu duong than cong . com](http://cuuduongthancong.com)

# MÔ HÌNH BỘ NHỚ

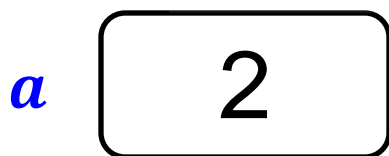
cuu duong than cong . com

cuu duong than cong . com

# Giới thiệu mô hình bộ nhớ

- Mỗi biến khi khai báo được biểu diễn bằng một ô nhớ

– Ví dụ : `int a = 2;`



- Khi tính toán sẽ thay đổi giá trị trên các ô nhớ tương ứng

– Ví dụ : `a = a + 3;`



# Example of memory model

```
int a;
```

Khai báo biến

```
int b, dienTich;
```

```
a = 2;
```

```
b = 3;
```

```
dienTich = a * b;
```

a



cuu duong than cong . com

# Example of memory model

```
int a;
```

```
int b, dienTich;
```

```
a = 2;
```

```
b = 3;
```

```
dienTich = a * b;
```

Khai báo biến

a

A rectangular box representing memory for variable 'a'.

b

A rectangular box representing memory for variable 'b'.

dienTich

A rectangular box representing memory for variable 'dienTich'.

# Example of memory model

```
int a;
```

```
int b, dienTich;
```

```
a = 2;
```

```
b = 3;
```

```
dienTich = a * b;
```

Tính toán

a

2

b

dienTich

# Example of memory model

```
int a;  
int b, dienTich;  
a = 2;  
b = 3;  
dienTich = a * b;
```

Tính toán

a

2

b

3

dienTich

# Example of memory model

```
int a;  
int b, dienTich;  
a = 2;  
b = 3;
```

```
dienTich = a * b;
```

Tính toán

a

2

b

3

dienTich

6

# Kiểu số nguyên

- Các phép toán trên bit cho số nguyên không dấu để:
  - lập trình **thao tác trên các bit**
  - **tăng tốc độ** xử lý của chương trình
- Bao gồm:
  - Phép **AND** bit: **&**
  - Phép **OR** bit: **|**
  - Phép **XOR** bit: **^**
  - Phép **NOT** bit: **~**
- Ví dụ (slide tiếp theo):

# Ví dụ toán tử trên bit

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

2.  void main()
3.  {
4.      unsigned char a = 45;           // 00101101
5.      unsigned char b = 58;           // 00111010
6.
7.      int c1, c2, c3, c4, c5, c6;
8.      c1 = a & b;                       // 00101000
9.      c2 = a | b;                       // 00111111
10.     c3 = a ^ b;                       // 00010111
11.     c4 = ~a;                          // 11010010
12.     c5 = a << 4;                      // 11010000
13.     c6 = a >> 4;                      // 00000010
14. }
```

# Kiểu Số Thực

cuu duong than cong . com

cuu duong than cong . com

# Kiểu số thực

- Cấu trúc lưu trữ được thiết kế theo **chuẩn số chấm động** (floating-point) của IEEE.

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range) (Trị tuyệt đối)
float	4	$1,4 \times 10^{-45} \dots 3,4 \times 10^{38}$
<i>float</i> có độ chính xác đơn (single-precision), chính xác đến 7 chữ số.		
double	8	$4,94 \times 10^{-324} \dots 1,79 \times 10^{308}$
<i>double</i> có độ chính xác kép (double-precision), chính xác đến 15 chữ số.		
long double	10	$\dots 3,4 \times 10^{4932}$
Một số môi trường lập trình đồng nhất kiểu <i>long double</i> với kiểu <i>double</i> cho nên kiểu này ít được sử dụng trong lập trình ứng dụng.		

- Ví dụ: float d = 15.06e-3; //  $15.06 \times 10^{-3}$  (e hay E)

# Ví dụ minh họa

- Trường hợp nào nên dùng biến số thực? Nếu có hãy khai báo.
  - **Tính điểm trung bình** của hai môn chính (**toán** và **văn**) của một học sinh
  - **Tính chu vi** và diện tích của một tam giác
  - **Tính dân số** của một quốc gia

cuu duong than cong . com

# Kiểu số thực

- Các phép toán số học
  - Phép cộng: **+**
  - Phép trừ: **−**
  - Phép nhân: **\***
  - Phép chia: **/**
- Ví dụ :  
`float a = 5.2;`  
`float b = a / 2;`
- Các hàm toán học như căn số, lũy thừa, logarit, ... sẽ được trình bày ở phần sau.

## Ví dụ minh họa

- Viết đoạn chương trình khai báo hai số nguyên  $x = 5$  và  $y = 4$ , tính trung bình cộng của hai số đó.
- Lưu ý:  $x/y$ 
  - Nếu  $x$  và  $y$  là hai số nguyên  $\rightarrow$  chia lấy phần nguyên
  - Nếu  $x$  hoặc  $y$  là số thực  $\rightarrow$  chia theo phép chia số thực

# Kiểu Luận Lý

cuu duong than cong . com

cuu duong than cong . com

# Kiểu luận lý

- Khai báo kiểu **bool** đối với C++ chuẩn hoặc kiểu số nguyên bất kỳ (char, int, ...)
  - Giá trị khác **0** nghĩa là đúng (**true**).
  - Giá trị bằng **0** nghĩa là sai (**false**).
  - *Lưu ý: Kết quả lượng giá một biểu thức luận lý bất kỳ thực hiện bởi C++ luôn cho kết quả là 0 (false) hay 1 (true).*
- Các phép toán
  - Kết hợp: **&&** (and), **||** (or), **!** (not)
  - So sánh: **>**, **>=**, **<**, **<=**, **==**, **!=**

# Ví dụ

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

2. void main()
3. {
4.     bool bVal;
5.     double x=46.7, y=93, z;
6.     bVal = (x==y);
7.     printf("%d\n", bVal);
8.     bVal = (x<y);
9.     printf("%d\n", bVal);
10.    bVal = (2*x>y);
11.    printf("%d\n", bVal);
12.    z = (x>y)*x + (x<=y)*y;
13.    printf("%f\n", z);
14.}
```

```
1. #include <iostream>
2. using namespace std;
3. void main()
4. {
5.     bool bVal;
6.     double x=46.7, y=93, z;
7.     bVal = (x==y);
8.     cout << bVal << endl;
9.     bVal = (x<y);
10.    cout << bVal << endl;
11.    bVal = (2*x>y);
12.    cout << bVal << endl;
13.    z = (x>y)*x + (x<=y)*y;
14.    cout << z << endl;
15.}
```

# Kiểu Ký Tự

cuu duong than cong . com

cuu duong than cong . com

# Kiểu ký tự

- Kiểu ký tự 8-bit
  - Kiểu **char** hoặc **unsigned char**.
  - Lưu mã ASCII của ký tự, giá trị từ 0 đến 255.
  - Một số ký tự nên nhớ

Ký tự	Mã
` ` (khoảng trắng)	32
`0' .. `9'	48 .. 57
`A' .. `Z'	65 .. 90
`a' .. `z'	97 .. 122

# Kiểu ký tự

- Đổi ký tự từ ký tự thường sang ký tự hoa:
  - Nếu  $'a' \leq ch \leq 'z'$  thì  $ch_{mới} = ch - ('a' - 'A')$
  - Ngược lại  $ch_{mới} = ch$
- Trong mọi trường hợp ta có công thức:
  - $ch_{mới} = ch - ('a' - 'A') * (ch \geq 'a' \ \&\& \ ch \leq 'z')$
- Tương tự ta cũng có công thức chuyển ký tự bất kỳ thành ký tự thường:
  - $ch_{mới} = ch - ('A' - 'a') * (ch \geq 'A' \ \&\& \ ch \leq 'Z')$

## Ví dụ minh họa

- Viết chương trình khai báo một biến chứa ký tự 'B'. Đổi ký tự đó thành ký tự thường 'b'

cuu duong than cong . com

cuu duong than cong . com

# Ví dụ

```
1. #include <stdio.h>
2. void main()
3. {
4.     char ch;
5.     ch=65;
6.     printf("ch = %c\n", ch);
7.     ch = 'A';
8.     printf("ch = %c\n", ch);
9.     printf("ch = ");
10.    scanf("%c", &ch);
11.    printf("ASCII code = %d\n", ch);
12.    ch -= ('a' - 'A')*(ch>='a' && ch<='z');
13.    printf("Upper case: %c\n", ch);
14. }
```

```
1. #include <iostream>
2. using namespace std;
3. void main()
4. {
5.     char ch;
6.     ch=65;
7.     cout << "ch = " << ch << endl;
8.     ch = 'A';
9.     cout << "ch = " << ch << endl;
10.    cout << "ch = ";
11.    cin >> ch;
12.    cout <<"ASCII code = " << ch << endl;
13.    ch -= ('a' - 'A')*(ch>='a' && ch<='z');
14.    cout << "Upper case: " << ch << endl;
15. }
```

# Kiểu ký tự

- Kiểu ký tự 16-bit
  - Kiểu `wchar_t` (`#include <wchar>`)
  - Lưu trữ dựa trên bảng mã quốc tế UTF-16 (một dạng mã Unicode)
    - Mã UTF-16 của ký tự thông thường ('0' đến '9', 'A' đến 'Z', 'a' đến 'z', ...) trùng mã ASCII.
  - Hằng ký tự kiểu `wchar_t` được đặt trước bằng chữ L
    - Lưu ý, 'B' và L'B' như nhau (cùng giá trị 66) nhưng kích thước trong bộ nhớ khác nhau (`sizeof('B') = 1`, `sizeof(L'B') = 2`)

## Độ lớn, độ chính xác, vấn đề tràn số (overflow)

- Đọc thêm trong giáo trình Nhập môn lập trình, Chương 2 – Phần III.6, trang 49-56.

cuu duong than cong . com

cuu duong than cong . com