

Nhập Môn Lập Trình Biến – Kiểu Dữ Liệu

TS. Tô Văn Khánh
Trường Đại học Công nghệ, ĐHQGHN

Nội Dung

- ▶ Hệ thống số
- ▶ Biến số
- ▶ Kiểu dữ liệu cơ bản
- ▶ Phép toán
- ▶ Hằng số
- ▶ Hàm số toán học

Chương Trình Đầu Tiên

- ▶ Sửa chương trình, không in “Hello, World !!!”
- ▶ Sử dụng chú thích (Ctrl + Shift + C / X – Code::Blocks)

```
/* 01_hello.cpp */  
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    cout << "Hello, World !!!" << endl;  
    return 0;  
}
```

Tổng & Hiệu 2 Số Nguyên Trong C++

Ctrl + Shift + F9 (dịch) & Ctrl + F10 (chạy)

```
/** chương trình C++ tính tổng hiệu 2 số nguyên **/  
/* 01_tong_hieu.cpp */  
#include <iostream>  
using namespace std;  
int main()  
{  
    int soThu1, soThu2;           // khai báo  
    cin >> soThu1;               // nhập dữ liệu  
    cin >> soThu2;               // nhập dữ liệu  
    int tong, hieu;              // khai báo  
    tong = soThu1 + soThu2;      // tính tổng  
    hieu = soThu1 - soThu2;      // tính hiệu  
    cout << "tong: " << tong << endl; // in kết quả  
    cout << "hieu: " << hieu << endl; // in kết quả  
    return 0;  
}
```

Tổng & Hiệu 2 Số Nguyên Trong C++

cout << (ghi ra màn hình) **cin** >> (đọc từ bàn phím)

```
/** chương trình C++ tính tổng hiệu 2 số nguyên **/  
/* 01_tong_hieu.cpp */  
#include <iostream>  
using namespace std;  
int main()  
{  
    int soThu1, soThu2;           // khai báo  
    cin >> soThu1;                // nhập dữ liệu  
    cin >> soThu2;                // nhập dữ liệu  
    int tong, hieu;               // khai báo  
    tong = soThu1 + soThu2;       // tính tổng  
    hieu = soThu1 - soThu2;       // tính hiệu  
    cout << "tong: " << tong << endl; // in kết quả  
    cout << "hieu: " << hieu << endl; // in kết quả  
    return 0;  
}
```

Tổng & Hiệu 2 Số Nguyên Trong C++

operator (toán tử) & operand (toán hạng)

```
/** chương trình C++ tính tổng hiệu 2 số nguyên **/  
/* 01_tong_hieu.cpp */  
#include <iostream>  
using namespace std;  
int main()  
{  
    int soThu1, soThu2;           // khai báo  
    cin >> soThu1;               // nhập dữ liệu  
    cin >> soThu2;               // nhập dữ liệu  
    int tong, hieu;              // khai báo  
    tong = soThu1 + soThu2;       // tính tổng  
    hieu = soThu1 - soThu2;      // tính hiệu  
    cout << "tong: " << tong << endl; // in kết quả  
    cout << "hieu: " << hieu << endl; // in kết quả  
    return 0;  
}
```

Tổng & Hiệu 2 Số Nguyên Trong C++

data (dữ liệu) & variable (biến số)

```
/** chương trình C++ tính tổng hiệu 2 số nguyên **/  
/* 01_tong_hieu.cpp */  
#include <iostream>  
using namespace std;  
int main()  
{  
    int soThu1, soThu2;           // khai báo  
    cin >> soThu1;               // nhập dữ liệu  
    cin >> soThu2;               // nhập dữ liệu  
    int tong, hieu;              // khai báo  
    tong = soThu1 + soThu2;      // tính tổng  
    hieu = soThu1 - soThu2;      // tính hiệu  
    cout << "tong: " << tong << endl; // in kết quả  
    cout << "hieu: " << hieu << endl; // in kết quả  
    return 0;  
}
```

Tổng & Hiệu 2 Số Nguyên Trong C++

data (dữ liệu) & variable (biến số)

```
/** chương trình C++ tính tổng hiệu 2 số nguyên **/  
/* 01_tong_hieu.cpp */  
#include <iostream>  
using namespace std;  
int main()  
{  
    int soThu1, soThu2;           // khai báo  
    cin >> soThu1;               // nhập dữ liệu  
    cin >> soThu2;               // nhập dữ liệu  
    int tong, hieu;              // khai báo  
    tong = soThu1 + soThu2;      // tính tổng  
    hieu = soThu1 - soThu2;      // tính hiệu  
    cout << "tong: " << tong << endl; // in kết quả  
    cout << "hieu: " << hieu << endl; // in kết quả  
    return 0;  
}
```


Hệ Thống Số

► Số nguyên

Cơ số 10: gồm các chữ số 0 1 2 3 4 5 6 7 8 9

10^2	10^1	10^0	
100	10	1	
3	5	2	
300	+	50	+
2	=	352	₁₀

Hệ Thống Số

► Số nguyên

Cơ số 16: gồm các chữ số 0 1 2 3 4 5 6 7 8 9 A B C D E F

16^2	16^1	16^0	
256	16	1	
B	1	A	
2816	+	16	+
		10	=
			2842₁₀

Hệ Thống Số

- ▶ Số nguyên

Cơ số 2: gồm các chữ số 0 1

2^2	2^1	2^0	
4	2	1	
1	0	1	
4	+	0	+
1			= 5₁₀

- ▶ Số nguyên âm: sử dụng bit trái ngoài cùng để biểu diễn

$$\begin{aligned} \mathbf{0000\ 0101_2} &= (0 * -2^7) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0) = \mathbf{5_{10}} \\ \mathbf{1000\ 0101_2} &= (1 * -2^7) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0) = \mathbf{-123_{10}} \end{aligned}$$

Dữ Liệu

- ▶ Dữ liệu là: giá trị, thông tin, ...
- ▶ Chương trình sử dụng dữ liệu để tính toán
- ▶ Dữ liệu được:
 - ▶ nhập vào trong chương trình
 - ▶ tạo ra bởi chương trình
 - ▶ xuất ra ngoài chương trình
- ▶ Ví dụ:
 - ▶ **5** và **3** được nhập vào (**cin**)
 - ▶ dùng phép cộng (toán tử **+**) để tính ra **8**
 - ▶ **8** được xuất ra (**cout**)

Kiểu Dữ Liệu

- ▶ C++ có nhiều kiểu dữ liệu:
 - ▶ dùng cho các mục đích khác nhau
- ▶ Mỗi kiểu dữ liệu, có:
 - ▶ Tên kiểu
 - ▶ Các giá trị / miền giá trị có thể biểu
 - ▶ Số lượng byte
 - 1 byte = 8 bit, 1 bit biểu diễn giá trị 0 hoặc 1 trong hệ nhị phân (cơ số 2)

Kiểu Số Nguyên – Kiểu `int`

- ▶ Kiểu dữ liệu `int` (số nguyên) có thể dùng để thể hiện
 - ▶ số lượng sinh viên lớp học (70)
 - ▶ số lượng môn học (6)
 - ▶ số tín chỉ (23)
- ▶ Kiểu dữ liệu `int` biểu diễn số nguyên từ -2^{31} đến $+2^{31} - 1$
- ▶ Kiểu `int` cần 4 byte để biểu diễn dữ liệu
- ▶ Các giá trị: 1, 2, -10

Kiểu Số Thực – Kiểu **double**

- ▶ Kiểu dữ liệu **double** (số thực) có thể dùng để thể hiện
 - ▶ quãng đường di chuyển (**3.14159**)
 - ▶ nghiệm phương trình bậc hai (**1.4142**)
 - ▶ sử dụng dấu **.** tách phần nguyên và phần thập phân
- ▶ Kiểu dữ liệu **double** biểu diễn số thực từ **-10^{308}** đến **$+10^{308}$**
- ▶ Kiểu **double** cần 8 byte để biểu diễn dữ liệu
- ▶ Các giá trị: **1.0**, **2.2**

Biến Số

- ▶ Dữ liệu thường được lưu vào biến số
- ▶ Dùng biến số để:
 - ▶ lưu dữ liệu nhập vào chương trình
 - ▶ lưu kết quả các phép tính trong chương trình
 - ▶ in kết quả tính toán của chương trình
- ▶ Biến số dùng để lưu dữ liệu, như vậy:
 - ▶ dữ liệu có kiểu (tr.12-14) \Rightarrow biến số cũng có kiểu
 - ▶ biến số có tên
 - sử dụng các biến số với tên khác nhau để lưu các dữ liệu trong chương trình

Khai Báo Biến Số

- ▶ Để sử dụng biến số, phải
 - ▶ đặt tên, ví dụ: **soThu1**, **tong**, **pass_or_fail**
 - ▶ xác định kiểu cho biến số
 - kiểu của dữ liệu: **int**, **double**
 - ▶ xác định giá trị ban đầu của biến số
- ▶ Cú pháp khai báo biến số:
data_type variable_name;
 - ▶ tạo một biến số có tên **variable_name** để lưu dữ liệu kiểu **data_type**
 - ▶ ví dụ: **int soThu1; int tong, hieu;**
 - ▶ ví dụ: **double radius, area;**

Khai Báo Biến Số

- ▶ **Phải khai báo biến số trước khi sử dụng**
 - ▶ nếu không trình biên dịch sẽ báo lỗi chưa khai báo
error: ... was not declared in this scope
- ▶ Sau khi khai báo, phải đặt giá trị ban đầu
 - ▶ nhập dữ liệu:
cin >> soThu1;
 - ▶ gán dữ liệu:
soThu1 = 0;
 - ▶ ngay sau khi khai báo, biến số có dữ liệu
int soThu1; cout << soThu1;
 - in ra dữ liệu có một giá trị nào đó

Phép Gán Biến Số

- ▶ Cú pháp gán / đặt biến số:

variable_name = value;

- ▶ gán dữ liệu **value** vào biến số **variable_name**

variable_name = expression;

- ▶ gán kết quả biểu thức **expression** vào biến số **variable_name**

- ▶ Ví dụ: **x = 24;**

x = (x * 24) / 12;

- ▶ Chú ý:

- ▶ bên trái của phép gán (dấu **=**) phải là biến số
- ▶ **x** có thể ở cả hai bên trong cùng một phép gán

Khởi Tạo Biến Số

- ▶ Cú pháp khởi tạo biến số:

`data_type variable_name = value;`

`data_type variable_name = expression;`

- ▶ Ví dụ:

`int x = 12;`

`int y = 12 + 6;`

`int z = (y * 24) / x;`

- ▶ Câu hỏi khó:

`int w = w + 12 / 6;`

- ▶ sau khởi tạo trên `w` bằng bao nhiêu

Quy Tắc Đặt Tên Biến Số

- ▶ C++ phân biệt chữ hoa và chữ thường
 - ▶ **tong** , **Tong**
- ▶ Tên biến bao gồm các ký tự chữ cái, chữ số, dấu gạch chân “_” (shift –)
 - ▶ **soThu1** , **pass_or_fail** **đúng**
- ▶ Tên biến số không bắt đầu bằng ký tự chữ số
 - ▶ **1PhuLuc** , **2KetLuan** **sai**
- ▶ Tên biến số không có dấu cách “ ”
 - ▶ **trung binh cong** **sai**
- ▶ Tên biến số không trùng với từ khóa C++
 - ▶ **int** , **return** **sai**

Phép Toán Số Học – Kiểu `int`

Phép Toán	Toán Tử	Ví Dụ	Kết Quả
Cộng	+	$a + b$	8
		$1 + 2$	3
Trừ	-	$a - b$	2
		$1 - 2$	-1
Nhân	*	$a * b$	15
		$1 * 2$	2
Chia	/	a / b	1
		$1 / 2$	0
Phần dư	%	$a \% b$	2
		$1 \% 2$	1

Phép Toán Số Học – Kiểu **double**

Phép Toán	Toán Tử	Ví Dụ		Kết Quả
Cộng	+	2.5	+ 1.5	4.0
		1.0	+ 2.2	3.2
Trừ	-	2.5	- 1.5	1.0
		1.0	- 2.2	-1.2
Nhân	*	2.5	* 1.5	3.75
		1.0	* 2.2	2.2
Chia	/	2.5	/ 1.5	1.66.....
		1.0	/ 2.2	0.45.....

Phép Chia Số Học

- ▶ Chia nguyên: cả hai toán hạng là số nguyên
- ▶ Chia thực: một trong hai toán hạng là số thực

```
int  main() {  
    int          i = 1,    j = 2,    k;  
    double       f = 1.0,  g = 2.0,  h;  
    k = i / j;    // chia nguyên k = 0  
    k = j / i;    // chia nguyên k = 2  
    h = f / g;    // chia thực h = 0.5  
    h = i / g;    // chia thực h = 0.5  
    h = f / j;    // chia thực h = 0.5  
    return 0;  
}
```


Kiểu Ký Tự – Kiểu **char**

- ▶ Kiểu **char** cần 1 byte để biểu diễn dữ liệu
- ▶ Trong bảng mã ASCII, số thứ tự - ký tự

```
char x = 65;
```

```
char y = 'A';
```

```
cout << x << endl;
```

```
cout << y << endl;
```

```
cout << (int) x << endl;
```

```
cout << (int) y << endl;
```

Kiểu Xâu Ký Tự – Kiểu **string**

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string first_name, last_name;
    cout << "Please enter your first name: ";
    cin >> first_name;
    cout << "Please enter your last name: ";
    cin >> last_name;
    string full_name;
    full_name = first_name + " " + last_name;
    cout << "Hello, " << full_name << endl;
    return 0;
}
```

Kiểu Logic – Kiểu **bool**

- ▶ Kiểu dữ liệu **bool** biểu diễn hai giá trị từ **false** đến **true**
 - ▶ **false** – sai; **true** – đúng
- ▶ Kiểu **bool** cần 1 byte để biểu diễn dữ liệu
- ▶ Các phép toán so sánh và các phép toán logic trả về giá trị kiểu logic (**false**, **true**)

Phép Toán So Sánh

Phép Toán	Toán Tử	Ví Dụ	Kết Quả
Nhỏ hơn	<	2.0 < 1.0 1 < 2	false true
Nhỏ hơn hoặc bằng	<=	2.0 <= 1.0 1 <= 2	false true
Lớn hơn	>	2.0 > 1.0 1 > 2	true false
Lớn hơn hoặc bằng	>=	1.0 >= 1.0 1 >= 2	true false
Bằng	==	1 == 1 1 == 2	true false
Không bằng (khác)	!=	2 != 2 1 != 2	false true

Phép Toán Logic

Phép Toán	Toán Tử	Ví Dụ	Kết Quả
Và	&&	true && false true && true	false true
Hoặc	 	true false false false	true false
Phủ định	!	!false !true	true false

Bảng Logic

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

a	b	a b
false	false	false
false	true	true
true	false	true
true	true	true

```
int a = 0;  
bool b;  
b = (false) && (1/a); cout << b;  
b = (true) || (1/a); cout << b;
```

Relationship Truth Table

Boyfriend	Girlfriend	Result

Độ Ưu Tiên Các Phép Toán

- Xác định thứ tự để tính giá trị biểu thức

Độ Ưu Tiên

Toán Tử

Cao nhất

$++$, $--$, $!$

$*$, $/$, $\%$

$+$, $-$

$<$, $<=$, $>$, $>=$

$==$, $!=$

$\&\&$

$||$

Thấp nhất

$=$, $+=$, $-=$, $*=$, $/=$, $\%=$

- Sử dụng cặp dấu ngoặc $()$ để dễ đọc

Chuyển Đổi Kiểu Dữ Liệu

- ▶ Giá trị thuộc một kiểu dữ liệu có thể được chuyển sang giá trị thuộc một kiểu dữ liệu khác
- ▶ Ví dụ: $1 \rightarrow 1.0$ hoặc $1.0 \rightarrow 1$
- ▶ 2 cách:
 - ▶ tự động bởi ngôn ngữ lập trình

```
int x = 1.0;           double y = 1.0;
double y = 1;          int x = y;
```
 - ▶ chủ động bởi lập trình viên

```
int x = (int)1.0;       double y = 1.0;
double y = (double)1;   int x = (int)y;
```
- ▶ Lưu ý: chuyển đổi kiểu làm thay đổi độ chính xác của giá trị. Sử dụng cẩn thận

Hằng Số

- ▶ Hằng số: giá trị không được phép thay đổi, nếu thay đổi, dịch báo lỗi
- ▶ Ví dụ: số ngày trong một tuần, số tháng trong một năm, tốc độ ánh sáng, số pi, số e
- ▶ Phải khởi tạo hằng số sử dụng từ khóa **const**
`const int DOW = 7;`
`const double PI = 3.14159;`
`const double SOL = 2.99e+8;`

Hàm Số Toán Học

- ▶ Nhiều hàm số toán học được định nghĩa sẵn
- ▶ Ví dụ: khai căn (**sqrt**), số mũ (**pow**), trị tuyệt đối (**abs**)
- ▶ Để sử dụng các hàm toán học cần tải các thư viện tương ứng

```
#include <cmath>
#include <cstdlib>
double x = sqrt(9);           // x = 3.0
double y = pow(3, 2);         // y = 9.0
int z = rand();
```

Tự Đọc

- ▶ Sử dụng **cout** <<, **cin** >>
- ▶ In dữ liệu ở các định dạng khác nhau
- ▶ Số nguyên trong các hệ cơ số
- ▶ Sử dụng kiểu số nguyên **char** để quản lý các ký tự (xem bảng mã ASCII)
- ▶ Sử dụng kiểu chuỗi ký tự **string**
- ▶ Các toán tử gán phức hợp **+=**, **-=**, ***=**, **/=**, **%=**
- ▶ Các toán tử tăng/giảm một đơn vị **++**, **--**
- ▶ Các hàm toán học trong **cmath**, **cstdlib**