

Bài 2: Các thành phần cơ bản của ngôn ngữ C++

1. Chương trình đầu tiên

```
#include <iostream>
```

← **Chỉ thị gọi tệp thư viện/tiền xử lý**

```
using namespace std;
```

← **xác định namespace (phạm vi tên gọi) là std**

```
int main(){
```

← **Hàm main, bắt đầu chương trình chính**

```
    cout << "Hello world!";
```

← **xuất ra màn hình chuỗi Hello Word!**

```
    return 0;
```

← **kết thúc chương trình và trả về giá trị 0.**

```
}
```

1. Chương trình đầu tiên

Như vậy, các thành phần của một chương trình C++ bao gồm:

- Các dẫn hướng `#include` để khai báo các thư viện cần dùng.
- Xác định namespace (thường là `std`)
- Bắt đầu chương trình chính với hàm `main`. Tất cả các lệnh trong chương trình chính được bọc trong cặp dấu ngoặc `{ }`;
- Mỗi câu lệnh (trừ một số trường hợp đặc biệt) đều kết thúc bằng dấu chấm phẩy;

Chú ý

Các chú ý khi viết chương trình:

- Xác định giải thuật rõ ràng trước khi viết.
- Trình bày trong sáng, có cấu trúc, dùng cú pháp đơn giản.
- Ghi chú khi cần.

Ghi chú trong chương trình

- Ghi chú (comments) là các lời giải thích ngay trong chương trình để người khác có thể hiểu được chương trình dễ dàng hoặc chính người lập trình khi đọc lại cũng dễ hiểu. Việc ghi chú không làm ảnh hưởng đến việc dịch và thực thi chương trình vì trình biên dịch sẽ tự động lọc bỏ các dòng ghi chú ra. Có hai cách ghi chú trong chương trình C++:

- **Ghi chú một dòng:** //

```
cout << "Hello world!"; //Xuất thông báo ra màn hình
```

- **Ghi chú nhiều dòng:** bắt đầu bằng dấu /* và kết thúc bằng dấu */

```
cout << "Hello world!";
```

```
/*Xuất thông báo ra màn hình.
```

```
Phải khai báo #include <iostream> trước khi dùng lệnh này*/
```

Đặt tên trong C++:

- Tên (identifier) là chuỗi ký tự dùng để đặt cho biến, hằng, hàm. Có thể dùng ký tự, chữ số và dấu gạch dưới (_) để đặt tên. Phân biệt giữa chữ thường và chữ hoa.

Ví dụ:

chieudai

chieu_rong

dt

- Chú ý: Không dùng khoảng trắng và các từ khóa dành riêng khi đặt tên.

Các kiểu dữ liệu trong C++

- Kiểu dữ liệu trong C++ được chia thành 3 nhóm:

- Các kiểu đơn giản

- Các kiểu có cấu trúc

- Con trỏ

Các kiểu dữ liệu đơn giản:

Tên kiểu	Từ khóa khai báo	Dung lượng	Khoảng giá trị
Số nguyên	int	4 byte	-2 tỉ đến +2 tỉ
Ký tự	char	1 byte	-127 đến 127
Số thực	float	4 byte	
Số thực	double	8 byte	

Chú ý: Dung lượng của từng kiểu dữ liệu có thể khác với bảng liệt kê ở trên, tùy thuộc vào trình biên dịch và hệ điều hành sử dụng.

Kí tự

- Ký tự được xử lý giống như một số nguyên. Giá trị ký tự được biểu diễn bằng ký tự tương ứng trong dấu nháy đơn, ví dụ 'A'. Giá trị số của ký tự chính là mã ASCII của ký tự.
- Một số ký tự đặc biệt thường dùng:

<code>\n</code>	Newline
<code>\t</code>	Tab
<code>\b</code>	Backspace
<code>\r</code>	Return
<code>\\</code>	Backslash
<code>\'</code>	Single quotation
<code>\"</code>	Double quotation

Biến (variable)

- Biến dùng để chứa tạm thời các giá trị khi xử lý trong chương trình. Giá trị của biến sẽ bị xóa khi kết thúc chương trình. Phải khai báo biến trước khi dùng.
- Cú pháp khai báo biến: Tên_kiểu tên_biến;

Ví dụ: `int n; char c;`

- Hai câu lệnh trên khai báo n là biến kiểu số nguyên, c là biến kiểu ký tự.
- Có thể ghép nhiều khai báo biến có cùng kiểu trên 1 dòng,

ví dụ: `float he_so_a, he_so_b, he_so_c;`

Quy tắc đặt tên cho biến

- Tên chỉ có thể chứa kí tự là chữ cái ('a' ..., 'z'; 'A' ..., 'Z'); chữ số('0' ..., '9') và kí tự gạch dưới (_), số kí tự không quá 32.
- Kí tự đầu tiên của tên phải là chữ cái hoặc kí tự gạch dưới
- Trong tên phân biệt chữ hoa và chữ thường
- Các từ khoá của ngôn ngữ không được dùng làm tên biến, tên hằng, hay tên hàm.

Các phép toán trong C++:

- Các phép toán số học:*

Tên phép toán	Ký hiệu	Tác dụng
Cộng	+	Cộng hai số
Trừ	-	Trừ hai số
Nhân	*	Nhân hai số
Chia	/	Chia hai số
Modulo	%	Chia hai số lấy phần dư
Tăng	++	Tăng số nguyên hiện hành lên 1
Giảm	--	Giảm số nguyên hiện hành đi 1

Chú ý: Các phép toán modulo, tăng và giảm chỉ dùng với số nguyên.

Thứ tự thực hiện các phép toán

- Phép tăng, giảm thực hiện trước*
- Phép toán trong ngoặc đơn thực hiện trước*
- Phép toán nhân, chia, modulo thực hiện trước các phép cộng, trừ.*

Chuyển đổi kiểu dữ liệu

- **Chuyển kiểu tự động:** Nếu tất cả các toán hạng trong biểu thức có cùng 1 kiểu thì kết quả sẽ cùng kiểu với toán hạng. Nếu các toán hạng trong biểu thức vừa có số nguyên vừa có số thực thì các số nguyên được tự động chuyển thành số thực trước khi thực hiện phép toán và giá trị của biểu thức sẽ là số thực. Đây là thao tác chuyển kiểu tự động.
- Chuyển kiểu tường minh: Người lập trình cũng có thể ép kiểu theo cách tường minh bằng cách dùng từ khóa `static_cast` theo cú pháp:

`static_cast<ten kiểu>(biểu thức);`

Hoặc: `kiểu_dữ_liệu(biểu_thức)`

Ví dụ: `static_cast<int> (6.7);`

Hằng

- Hằng là đại lượng có giá trị thuộc một kiểu dữ liệu nhất định, nhưng giá trị của hằng không thể thay đổi trong thời gian tồn tại của nó.

Các hằng được định nghĩa bằng từ khoá **const** với cú pháp như sau:

const <kiểu_dữ_liệu> <tên_hằng> = <giá_trị>;

Vị dụ: const float pi = 3.1416;

Lệnh gán (assignment):

- Lệnh này dùng để gán một giá trị cho một biến. Giá trị có thể là một hằng, một biểu thức.

Ví dụ:

```
int n;
```

```
n = 10;
```

```
float p;
```

```
p = (a+b+c)/2;
```


Nhập dữ liệu vào chương trình:

- Dữ liệu nhập (từ bàn phím) được gán cho biến trước khi xử lý trong chương trình.
- Câu lệnh cin (trong thư viện iostream) được dùng để nhập dữ liệu cho biến với cú pháp:
- cin >> tên biến;

Ví dụ:

```
int n;
```

```
cin >> n;
```

Câu lệnh trên khai báo biến n kiểu số nguyên và lấy giá trị nhập từ bàn phím để gán cho n.

Bài tập

- **Bài tập 1:** Viết chương trình nhập điểm các môn: Toán A1, Vật lý, Triết học, Tin học cơ sở và Giáo dục thể chất. Tính điểm trung bình và xuất ra màn hình, biết rằng điểm Toán A1 và Tin học cơ sở có hệ số 2, tất cả các môn khác có hệ số 1.
- **Bài tập 2:** viết chương trình nhập vào một khoảng thời gian tính bằng giây (số nguyên), sau đó xác định có bao nhiêu giờ, bao nhiêu phút, bao nhiêu giây. Ví dụ nhập vào 3672 giây thì số giờ là 1, số phút là 1 và số giây là 12.