

Cú pháp C++ cơ bản

Khi chúng ta xem xét một chương trình C++, nó có thể được định nghĩa như là một tập hợp của các đối tượng, mà giao tiếp thông qua việc triệu hồi các phương thức của mỗi đối tượng đó. Dưới đây, chúng tôi miêu tả ngắn gọn ý nghĩa của class (lớp), object (đối tượng), method (phương thức) và các biến đối tượng:

- Đối tượng - Đối tượng có các trạng thái và hành vi. Ví dụ: một đối tượng dog có các trạng thái là color, name, breed, và các hành vi là wagging, barking, eating. Một đối tượng là một minh họa của một lớp.
- Lớp - Một lớp có thể được định nghĩa như là một template/blueprint, mà miêu tả hành vi/trạng thái mà đối tượng hỗ trợ.
- Phương thức - Về cơ bản, một phương thức là một hành vi. Một lớp có thể chứa nhiều phương thức. Phương thức là nơi tính logic được viết, dữ liệu được thao tác và tất cả action được thực thi.
- Biến instance - Mỗi đối tượng có tập hợp biến đối tượng duy nhất của nó. Trạng thái của một đối tượng được tạo ra bởi các giá trị được gán cho các biến đối tượng của nó.
- ADVERTISING

Cấu trúc chương trình C++

Bạn theo dõi một đoạn code đơn giản sẽ in *Hello World*.

```
#include <iostream>

using namespace std;
```

```
// Ham main() la noi su thuc thi chuong trinh bat dau
```

```
int main()
```

```
{
```

```
    cout << "Hello World"; // In dong chu Hello World
```

```
    return 0;
```

```
}
```

Chương trình trên có các phần sau:

- Ngôn ngữ C++ định nghĩa một số header, mà chứa tong tin cần thiết và hữu ích cho chương trình của bạn. Với chương trình này, header là `<iostream>` là cần thiết.
- Dòng `using namespace std;` nói cho compiler sử dụng `std` namespace. Namespce là phần bổ sung gần đây cho C++.
- Dòng tiếp theo `Ham main()` la noi su thuc thi chuong trinh bat dau là một comment đơn dòng trong C++. Các comment đơn dòng bắt đầu với `//` và kết thúc ở cuối dòng.
- Dòng `int main()` là hàm main, tại đây việc thực thi chương trình bắt đầu.
- Dòng tiếp theo `cout << "Hello World";` để in dòng chữ "Hello World" trên màn hình.
- Dòng tiếp theo `return 0;` kết thúc hàm `main()` và làm nó trả về giá trị 0 tới tiến trình đang gọi.

Biên dịch và thực thi chương trình C++

1. Dưới đây là cách lưu file, biên dịch và chạy chương trình với command prompt. Bạn theo các bước sau:

- Mở một text editor và thêm đoạn code trên.
- Lưu file với tên: hello.cpp.
- Mở một dòng nhắc lệnh (command prompt) và tới thư mục nơi bạn lưu file đó.
- Soạn 'g++ hello.cpp ' và nhấn Enter để biên dịch code trên. Nếu không có lỗi xảy ra trong code của bạn thì dòng nhắc lệnh sẽ đưa bạn tới dòng tiếp theo và tạo a.out file có thể thực thi.
- Bây giờ, bạn soạn 'a.out' để chạy chương trình.
- Bạn sẽ thấy 'Hello World' được in trên cửa sổ.

```
$ g++ hello.cpp
```

```
$ ./a.out
```

```
Hello World
```

Bạn chắc chắn rằng g++ là trong path của bạn và bạn đang chạy nó trong thư mục mà chứa hello.cpp file.

2. Đó là cách biên dịch và thực thi chương trình tại command prompt. Nếu bạn đang sử dụng Dev-C++, hoặc Microsoft Visual Studio, hoặc Turbo C++, bạn có thể sao chép ví dụ trên, sau đó:

Mở Dev-C++ chẳng hạn, chọn File -> New Source, sau đó dán ví dụ vào source file này và chọn Execute tab, chọn Compile & Run để thực thi chương trình. Bạn cũng có thể tạo một Project mới trong tùy chọn New để đặt tên và lưu cho ví dụ vừa thực hiện.

Dấu chấm phẩy và khối (block) trong C++

Trong C++, dấu chấm phẩy là ký tự kết thúc lệnh (statement terminator). Nghĩa là, mỗi lệnh đơn phải kết thúc bởi một dấu chấm phẩy. Nó chỉ dẫn sự kết thúc của một thực thể logic.

Dưới đây là ví dụ về 3 lệnh khác nhau:

```
x = y;
```

```
y = y+1;
```

```
add(x, y);
```

Một khối (block) là một tập hợp các lệnh được kết nối một cách logic, mà được bao quanh bởi các dấu ngoặc móc mở và đóng. Ví dụ:

```
{
```

```
    cout << "VietJack xin chào các bạn!"; // in dòng chữ VietJack xin chào các bạn!
```

```
    return 0;
```

```
}
```

Hai dạng sau là tương đương, C++ không quan tâm bạn đặt bao nhiêu lệnh trên một dòng. Ví dụ:

```
x = y;
```

```
y = y+1;
```

```
add(x, y);
```

Là giống với:

```
x = y; y = y+1; add(x, y);
```

Định danh (Identifier) trong C++

Một Định danh (Identifier) trong C++ là một tên được sử dụng để nhận diện một biến, hàm, lớp, module, hoặc bất kỳ user-defined item nào (người dùng tự định nghĩa). Một Định danh (Identifier) bắt đầu với một chữ cái từ A tới Z hoặc từ a tới z hoặc một dấu gạch dưới (_) được theo sau bởi 0 hoặc nhiều chữ cái, dấu gạch dưới và chữ số (từ 0 tới 9).

C++ không cho phép các ký tự như @, \$ và % bên trong các Identifier. C++ là ngôn ngữ lập trình phân biệt kiểu chữ. Vì thế, Manpower và manpower là hai Identifier khác nhau trong C++.

Dưới đây là một số ví dụ về Identifier (Định danh) thích hợp:

```
hoang      nam      abc      sinh_vien  a_123  
  
caogia50  _nhanvien  j      a23b9      vietJack
```

Từ khóa trong C++

Bảng dưới liệt kê các từ được dự trữ (dành riêng) trong C++. Những từ này không thể được sử dụng như là constant hoặc biến hoặc bất kỳ tên Identifier (Định danh) nào.

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union

const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

Trigraph trong C++

Một Trigraph là một dãy 3 ký tự mà biểu diễn một ký tự đơn và dãy này luôn luôn bắt đầu với 2 dấu hỏi.

Các Trigraph được mở rộng bất cứ nơi đâu chúng xuất hiện, bao gồm bên trong String literal và Character literal, trong comment, và trong các preprocessor directive (các directive tiền xử lý).

Dưới đây là các dãy trigraph được sử dụng thường xuyên nhất:

Trigraph	Thay thế cho
----------	--------------

??=	#
??/	\
??'	^
??([
??)]
??!	
??<	{
??>	}
??-	~

Tất cả compiler không hỗ trợ Trigraph và chúng được khuyến là không nên sử dụng bởi vì tính khó hiểu của nó.

Khoảng trắng (Whitespace) trong C++

Một dòng mà chỉ chứa khoảng trắng (Whitespace), có thể là một comment, được biết đến như là một dòng trống, và C++ hoàn toàn bỏ qua nó.

Khoảng trắng (Whitespace) là khái niệm được sử dụng trong C++ để miêu tả blank, tab, ký tự newline (dòng mới), và comment. Khoảng trắng (Whitespace) phân biệt các phần của lệnh và giúp compiler nhận diện vị

trí một phần tử trong một lệnh, ví dụ như int, vị trí phần kết thúc và vị trí phần tử tiếp theo bắt đầu. Do đó, trong lệnh sau:

```
int diemthi;
```

Phải có ít nhất một Whitespace (thường là khoảng trống) giữa int và age để compiler có thể phân biệt chúng. Trong lệnh:

```
tongLuong = luongCoBan + phuCap; // tinh tong luong
```

Các ký tự khoảng trắng (Whitespace) giữa luongCoBan và =, hoặc giữa = và phuCap là không cần thiết; tuy nhiên, để giúp cho code của bạn dễ đọc hơn, bạn có thể thêm chúng vào.