

Bài: Cấu trúc một chương trình C++ (Structure of a program)

Xem bài học trên website để ủng hộ Kteam: [Cấu trúc một chương trình C++ \(Structure of a program\)](#).

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài học trước, [XÂY DỰNG CHƯƠNG TRÌNH ĐẦU TIÊN VỚI VISUAL STUDIO 2015](#), bạn đã hiểu được nguyên lý làm việc cơ bản của một IDE, và cách xây dựng một chương trình C++ cơ bản.

Nhưng có thể bạn chưa hiểu được những dòng code mà bạn gõ vào có ý nghĩa gì. Đừng lo lắng, bạn sẽ nắm rõ nó sau bài học này: **Cấu trúc một chương trình C++ (Structure of a program)**.

Nội dung

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Cấu trúc của một chương trình C++
- Cú pháp và lỗi cú pháp trong C++ (Syntax and syntax errors)

Cấu trúc của một chương trình C++

Trước tiên, bạn cần nắm rõ cấu trúc của một chương trình máy tính gồm những gì.

• Các câu lệnh và biểu thức (Statements and expressions)

- Phổ biến nhất trong một chương trình máy tính là **câu lệnh (Statements)**. Trong C++, câu lệnh là đơn vị độc lập nhỏ nhất. Mỗi câu lệnh sẽ là một yêu cầu **Compiler** thực hiện một nhiệm vụ nhất định.
- **Câu lệnh** trong C++ kết thúc bằng một **dấu chấm phẩy ";"**.
- **Ví dụ:** một số câu lệnh cơ bản trong C++:

C++:

```
int nSoNguyen;
nSoNguyen = 69;
nSoNguyen = (6 + 9) * (6 - 9);
std::cout << "Hello HowKTeam.com" << std::endl;
```

- **int nSoNguyen** là **câu lệnh khai báo**, và nSoNguyen là một **biến số nguyên**. Biến được khai báo là tên gọi cho một vùng trong bộ nhớ, dùng để lưu giá trị và có thể thay đổi được. Biến phải được khai báo trước khi sử dụng. Phần này sẽ được nói đến trong bài [BIẾN TRONG C++](#)
- **nSoNguyen = 69** là **câu lệnh gán**. Nó dùng để gán một giá trị cho một biến, giá trị đó có thể là một giá trị cụ thể, hoặc là kết quả của một biểu thức, một hàm.
- **std::cout << "Hello HowKTeam.com" << std::endl** là một **câu lệnh xuất** thuộc thư viện iostream, nó sẽ xuất ra màn hình một câu thông báo bất kỳ.

• Hàm (Functions)

- Hàm là một nhóm các câu lệnh đi cùng nhau, để thực hiện một nhiệm vụ. Mỗi chương trình C++ có ít nhất một hàm là hàm **main()**,

• Thư viện chuẩn C++ (C++ Standard Library)

- Thư viện là một tập hợp các mã được biên dịch sẵn, được đóng gói lại để lập trình viên sử dụng, mà không cần phải viết lại. Ví dụ: bạn viết một chương trình tính toán, bạn có thể include thư viện toán học...

Tiếp theo, bạn đã nắm được các thành phần của một chương trình là gì. Bây giờ, bạn hãy nhìn vào chương trình "Hello HowKteam.com" mà bạn đã xây dựng trong bài trước:

C++:

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

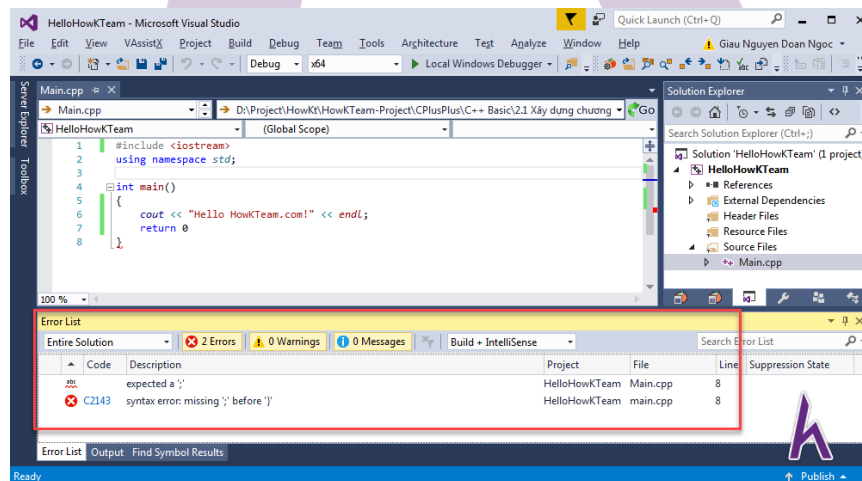
int main()
{
    cout << "Hello HowKteam.com!" << endl;
    return 0;
}
```

- **Dòng 1:** Đây gọi là một chỉ thị tiền xử lý, là một dạng câu lệnh đặc biệt. Dòng này yêu cầu compiler bổ sung thêm nội dung của iostream header vào chương trình. Iostream header sẽ cho phép bạn truy cập vào các function trong iostream library.
- **Dòng 2:** Dòng này cho biết chương trình của bạn sẽ sử dụng namespace std (là một standard namespace được định nghĩa sẵn một lượng lớn các class, object, function thường được sử dụng) là namespace mặc định. Khi tham chiếu đến các class, object, function trong namespace std thì không cần tiền tố std::: Exp: Nếu không khai báo namespace std, thì dòng số 6 bạn phải viết là `std::cout << "Hello HowKteam.com!" << std::endl;`
- **Dòng 3:** Compiler sẽ bỏ qua dòng trống.
- **Dòng 4:** Đây là khai báo main() function, mỗi chương trình C++ phải có một hàm main(). Khi chương trình C++ được chạy, hàm main() sẽ được gọi đầu tiên, và những dòng code trong hàm main() sẽ được thực thi từ trên xuống dưới.
- **Dòng 5, 8:** Mọi dòng code thuộc một hàm phải nằm trong phạm vi dấu {} của hàm đó.
- **Dòng 6:** Là một câu lệnh, nó dùng để hiển thị câu "Hello HowKteam.com" ra màn hình. Cout là một đối tượng thuộc thư viện iostream, << là một output operator. Mọi thông điệp gửi đến cout thông qua toán tử << sẽ được hiển thị lên màn hình. Câu lệnh kết thúc bởi dấu ";".
- **Dòng 7:** là một câu lệnh return. Hàm main() có kiểu int nên bắt buộc phải có một câu lệnh return giá trị kiểu int. Khi chương trình thực thi kết thúc, hàm main() sẽ return một giá trị cho hệ điều hành, dù nó chạy thành công hay không.
 - Nếu giá trị trả về là 0, nghĩa là mọi thứ đã chạy thành công.
 - Nếu giá trị trả về khác 0, nghĩa là đã có gì đó đã chạy sai và chương trình đã dừng lại.

Cú pháp và lỗi cú pháp trong C++ (Syntax and syntax errors)

Cú pháp trong C++ là những **quy tắc viết code** khi xây dựng một chương trình. Khi bạn compile một chương trình, compiler sẽ đảm bảo chương trình của bạn đã đúng cú pháp ngôn ngữ C++ hay chưa. Nếu bạn vi phạm một quy tắc, dù nhỏ hay lớn thì compiler cũng sẽ thông báo cho bạn thông tin về lỗi đó, và bạn bắt buộc phải sửa nó.

- **Ví dụ:** Trong hình bên dưới, bạn thử bỏ đi một dấu ";" sau dòng return 0. Compiler sẽ thông báo cho bạn về thông tin lỗi (mô tả lỗi, ở project nào lỗi, file nào lỗi, dòng nào lỗi) ở cửa sổ Error list. Từ đó bạn có thể dễ dàng fix nó.



Kết luận

Qua bài học này, bạn đã hiểu được cấu trúc một chương trình cơ bản trong C++ và ý nghĩa của nó.

Trong bất cứ ngành nghề nào, chắc chắn bạn không chỉ làm việc một mình, đặc biệt trong lập trình, bạn muốn đồng nghiệp hoặc những thế hệ sau có thể dễ dàng hiểu được và kế thừa những dòng code của bạn viết ra, hoặc để vài năm sau đọc lại bạn vẫn đảm bảo hiểu được mình viết gì trong đó.

Để làm được chuyện đó, ngoài việc tuân thủ các coding convention, naming convention, ... thì một trong những cách truyền đạt ý nghĩa đoạn code của bạn cho mọi người sẽ được nói đến trong bài tiếp theo: [GHI CHÚ TRONG C++ \(Comments\)](#).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.

