

# Bài: Hàm nội tuyến trong C++ (Inline functions)

Xem bài học trên website để ủng hộ Kteam: [Hàm nội tuyến trong C++ \(Inline functions\)](#)

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, mình đã chia sẻ cho các bạn về vấn đề trả về của hàm thông qua ba phương pháp [HÀM TRẢ VỀ GIÁ TRỊ, THAM CHIẾU VÀ ĐỊA CHỈ \(value, reference, and address\)](#).

Trong bài học này, chúng ta sẽ cùng tìm hiểu về **Hàm nội tuyến trong C++ (Inline functions)**.

## Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về:

- [CƠ BẢN VỀ HÀM VÀ GIÁ TRỊ TRẢ VỀ \(Basics of Functions and Return values\)](#)
- [TRUYỀN GIÁ TRỊ CHO HÀM \(Passing Arguments by Value\)](#)
- [TRUYỀN THAM CHIẾU CHO HÀM \(Passing Arguments by Reference\)](#)
- [TRUYỀN ĐỊA CHỈ CHO HÀM \(Passing arguments by address\)](#)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Đặt vấn đề
- Hàm nội tuyến trong C++ (Inline functions)
- Những vấn đề cần lưu ý khi dùng inline functions
- Tổng kết về hàm nội tuyến

## Đặt vấn đề

**Hàm nội tuyến (inline functions)** là một trong những **tính năng quan trọng của C++**. Vì vậy, trước tiên hãy hiểu **tại sao các hàm nội tuyến được sử dụng và mục đích của hàm nội tuyến là gì?**

Khi một hàm được gọi, CPU sẽ **lưu địa chỉ bộ nhớ** của dòng lệnh hiện tại mà nó đang thực thi (để biết nơi sẽ quay lại sau lời gọi hàm), **sao chép các đối số** của hàm trên ngăn xếp (stack) và cuối cùng **chuyển hướng điều khiển** sang hàm đã chỉ định. CPU sau đó thực thi mã bên trong hàm, **lưu trữ giá trị trả về** của hàm trong một vùng nhớ/thanh ghi và trả lại quyền điều khiển cho vị trí lời gọi hàm. Điều này sẽ tạo ra một lượng chi phí hoạt động nhất định so với việc thực thi mã trực tiếp (không sử dụng hàm).

Đối với các **hàm lớn** hoặc các **tác vụ phức tạp**, tổng chi phí của lệnh gọi hàm thường không đáng kể so với lượng thời gian mà hàm mất để chạy. Tuy nhiên, đối với các hàm nhỏ, thường được sử dụng, thời gian cần thiết để thực hiện lệnh gọi hàm thường nhiều hơn rất nhiều so với thời gian cần thiết để thực thi mã của hàm.

Trong C, chúng ta thường sử dụng hàm Macro, một kỹ thuật tối ưu hóa được sử dụng bởi trình biên dịch để giảm thời gian thực hiện. C++ cung cấp một khái niệm mới tốt hơn, đó là **hàm nội tuyến (inline functions)**.

## Hàm nội tuyến trong C++ (Inline functions)

**Inline functions (hàm nội tuyến)** là một loại hàm trong ngôn ngữ lập trình C++. Từ khóa **inline** được sử dụng để **đề nghị (không phải là bắt buộc)** compiler (trình biên dịch) thực hiện **inline expansion (khai triển nội tuyến)** với hàm đó hay nói cách khác là chèn code của hàm đó tại địa chỉ mà nó được gọi.

:

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

inline int max(int a, int b)
{
    return a > b ? a : b;
}

int main()
{
    cout << max(3, 6) << '\n';
    cout << max(6, 3) << '\n';

    return 0;
}
```

Khi chương trình trên được biên dịch, mã máy được tạo ra tương tự như hàm **main()** bên dưới:

:

```
int main()
{
    cout << (3 > 6 ? 3 : 6) << '\n';
    cout << (6 > 3 ? 6 : 3) << '\n';

    return 0;
}
```

Trong trường hợp này, sử dụng **inline functions** sẽ thực thi nhanh hơn một chút so với hàm thông thường.

## Những vấn đề cần lưu ý khi dùng inline functions

Hãy nhớ rằng, từ khóa **inline (nội tuyến)** chỉ là một yêu cầu cho trình biên dịch, không phải là một lệnh bắt buộc.

Trình biên dịch **có thể không thực hiện nội tuyến** trong các trường hợp như:

- Hàm chứa vòng lặp (for, while, do-while).
- Hàm chứa các biến tĩnh.
- Hàm đệ quy.
- Hàm chứa câu lệnh switch hoặc goto.

Hầu hết các trình biên dịch hiện đại sẽ tự động đặt các hàm nội tuyến nếu cần thiết. Do đó, trong hầu hết các trường hợp, nếu không có nhu cầu cụ thể để sử dụng từ khóa nội tuyến, hãy để trình biên dịch xử lý các hàm nội tuyến cho bạn.

## Tổng kết về hàm nội tuyến

### Ưu điểm:

- Tiết kiệm chi phí gọi hàm.
- Tiết kiệm chi phí của các biến trên ngăn xếp khi hàm được gọi.
- Tiết kiệm chi phí cuộc gọi trả về từ một hàm.
- Có thể đặt định nghĩa hàm nội tuyến (inline functions) trong file tiêu đề (\*.h) (nghĩa là nó có thể được **include** trong nhiều đơn vị biên dịch, hàm thông thường sẽ gây ra lỗi).

### Nhược điểm:

- Tăng kích thước file thực thi do sự trùng lặp của cùng một mã.
- Khi được sử dụng trong file tiêu đề (\*.h), nó làm cho file tiêu đề của bạn lớn hơn.

- Hàm nội tuyến có thể không hữu ích cho nhiều hệ thống nhúng. Vì trong các hệ thống nhúng, kích thước mã quan trọng hơn tốc độ.

## Kết luận

Qua bài học này, bạn đã nắm được những kiến thức về Hàm nội tuyến trong C++ (Inline functions). Lưu ý rằng trong hầu hết các trường hợp, nếu không có nhu cầu cụ thể để sử dụng từ khóa nội tuyến, hãy để trình biên dịch xử lý các hàm nội tuyến cho bạn

Trong bài tiếp theo, chúng ta sẽ cùng tìm hiểu về [NẠP CHỒNG HÀM TRONG C++ \(Function overloading\)](#).

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"**.

