

Hàm nguyên mẫu trong C++

Cho đến giờ chúng ta hoàn toàn phải định nghĩa hàm trước lệnh gọi đầu tiên đến nó, mà thường là trong **main**, vì vậy hàm **main** luôn phải nằm cuối chương trình. Nếu bạn thử lặp lại một vài ví dụ về hàm trước đây nhưng thử đặt hàm **main** trước bất kì một hàm được gọi từ nó, bạn gần như chắc chắn sẽ nhận được thông báo lỗi. Nguyên nhân là một hàm phải được khai báo trước khi nó được gọi như những gì chúng ta đã làm trong tất cả các ví dụ.

Nhưng có một cách khác để tránh phải viết tất cả mã chương trình trước khi chúng có thể được dùng trong **main** hay bất kì một hàm nào khác. Đó chính là *khai báo mẫu cho hàm (hàm nguyên mẫu)*. Cách này bao gồm việc khai báo hàm một cách ngắn gọn nhưng đủ để cho trình dịch có thể biết các tham số và kiểu dữ liệu trả về của hàm.

Dạng của nó như sau:

```
type name ( argument_type1, argument_type2, ... );
```

Đây chính là phần đầu của định nghĩa hàm, ngoại trừ:

- Nó không có bất kì lệnh nào cho hàm. Điều này có nghĩa là nó không bao gồm thân hàm với tất cả các lệnh thường được bọc trong cặp ngoặc nhọn { }.
- Nó kết thúc bằng dấu chấm phẩy (;).
- Trong phần liệt kê các tham số chỉ cần viết kiểu của chúng là đủ. Việc viết tên của các tham số trong phần khai báo mẫu là không bắt buộc.

Ví dụ:

```
// prototyping
#include <iostream.h>

void odd (int a);
void even (int a);

int main ()
{
    int i;
    do {
        cout << "Type a number: (0 to exit)";
        cin >> i;
        odd (i);
```

```

    } while (i!=0);
    return 0;
}

void odd (int a)
{
    if ((a%2)!=0) cout << "Number is odd.\n";
    else even (a);
}

void even (int a)
{
    if ((a%2)==0) cout << "Number is even.\n";
    else odd (a);
}

```

Kết quả:

```

Type a number (0 to exit): 9
Number is odd.
Type a number (0 to exit): 6
Number is even.
Type a number (0 to exit): 1030
Number is even.
Type a number (0 to exit): 0
Number is even.

```

Ví dụ này rõ ràng không phải là một ví dụ về sự hiệu quả. Chắc chắn rằng các bạn có thể nhận được kết quả như trên chỉ với một nửa số dòng lệnh. Tuy nhiên nó giúp cho chúng ta thấy được việc khai báo mẫu các hàm là như thế nào. Hơn nữa, trong ví dụ này việc khai báo mẫu ít nhất một hàm là bắt buộc.

Đầu tiên chúng ta thấy khai báo mẫu của hai hàm **odd** và **even**:

```

void odd (int a);
void even (int a);

```

cho phép hai hàm này có thể được sử dụng trước khi chúng được định nghĩa hoàn chỉnh. Tuy nhiên lý do đặc biệt giải thích tại sao chương trình này lại cần ít nhất một hàm phải được khai báo mẫu là trong **odd** có một lời gọi đến **even** và trong **even** có một lời gọi đến **odd**. Vì vậy nếu không có hàm nào được khai báo trước thì lỗi chắc chắn sẽ xảy ra.

Rất nhiều lập trình viên kinh nghiệm khuyên rằng tất cả các hàm nên được khai báo mẫu, nhất là trong trường hợp có nhiều hàm hoặc chúng rất dài, khi đó việc khai báo tất cả các hàm ở cùng một chỗ cho phép chúng ta biết phải gọi các hàm như thế nào, vì vậy tiết kiệm được thời gian.

Revision #1

Created 5 October 2019 05:50:17 by Laptrinh.vn

Updated 5 October 2019 05:52:47 by Laptrinh.vn