Bài 27: Câu lệnh Goto trong C++ (Goto statements)

Xem bài học trên website để ủng hộ Kteam: Câu lệnh Goto trong C++ (Goto statements)

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 cho các bạn về CÂU ĐIỀU KIỆN SWITCH TRONG C++ (Switch statements).

Hôm nay, mình sẽ giới thiệu cho các bạn về **Câu lệnh Goto trong C++ (Goto statements)**.

> **Kinh nghiệm:** Tránh sử dụng câu lệnh Goto trừ khi bạn có lý do đặc biệt nào đó.

Nội dung

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Tổng quan về câu lệnh Goto trong C++
- Một số vấn đề của câu lệnh Goto

Tổng quan về câu lệnh Goto trong C++

Lệnh goto trong C++ cung cấp một **bước nhảy không điều kiện** từ lệnh **goto** tới lệnh được gán **nhãn** trong **cùng một hàm**.

Cú pháp của lệnh goto:

```
goto label;
..
.
label: statement;
```

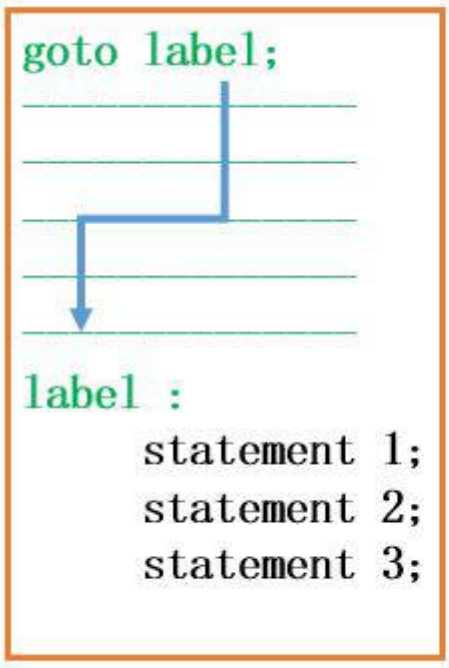
Ví dụ:

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

int main()
{
	int n;

tryAgain: // nhãn

	cout << "Nhap so nguyen duong:";
	cin >> n;

	if (n < 0)
		goto tryAgain; // nhảy đến nhãn tryAgain

	cout << n << " la so nguyen duong" << endl;

	return 0;
}
```

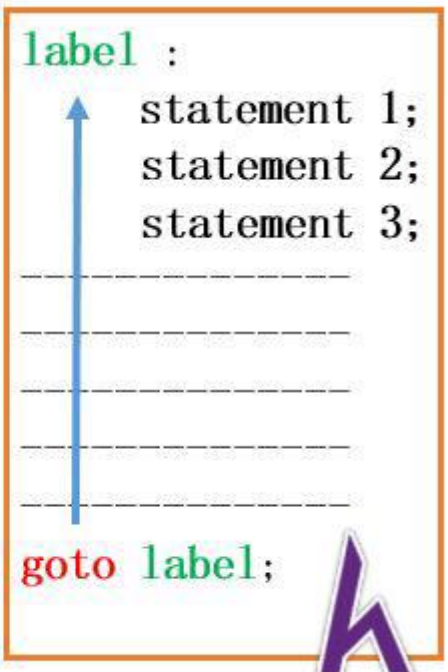
Outputs:

Chương trình trên yêu cầu người dùng nhập một số nguyên dương. Tuy nhiên, nếu bạn nhập một số âm, chương trình sẽ sử dụng lệnh **goto** để nhảy đến nhãn **tryAgain**. Chương trình sẽ lặp lại thao tác nhập và chỉ kết thúc khi người dùng nhập vào một số nguyên dương.

Câu lệnh goto được chia thành **2 loại**:



Forward Reference



Backward Reference

Một số vấn đề của câu lệnh Goto

Phạm vi của nhãn trong câu lệnh Goto

Nhãn trong câu lệnh Goto có **phạm vi hàm**. Các lệnh **goto và nhãn** tương ứng của nó phải **nằm trong cùng một hàm**.

Ví dụ:

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

void print()
{
label:
	cout << "print" << endl;
}

int main()
{
	goto label;
}
```

Chương trình trên có lỗi vì câu lệnh **goto** và nhãn phải nằm cùng một hàm.

Hạn chế của câu lệnh Goto

Bạn không thể nhảy qua một câu lệnh khởi tạo biến trong cùng một khối lệnh.

Ví dụ:

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

void print()
{
label:
	cout << "print" << endl;
}

int main()
```

```
{
        goto label;
}
```

Chương trình trên, câu lệnh Goto đã nhảy qua dòng lệnh khởi tạo biến **x**. Do đó, compiler ném ra một lỗi vì **biến x không xác định được giá trị** khi xuất.

Nhìn chung, bạn **không nên sử dụng câu lệnh Goto**. Vì nó **gây khó khăn** cho việc **theo dấu dòng điều khiển** của một chương trình, làm cho chương trình **khó hiểu và khó chỉnh sửa**.

Trong C++ (và hầu hết các ngôn ngữ bậc cao khác), câu lệnh Goto hầu như không bao giờ được sử dụng. Bất kỳ chương trình nào sử dụng câu lệnh goto đều có thể được viết lại rõ ràng hơn khi sử dụng các cấu trúc khác trong C++.

> **Kinh nghiệm:** Tránh sử dụng câu lệnh Goto trừ khi bạn có lý do đặc biệt nào đó.

Kết luận

Qua bài học này, bạn đã nắm rõ về Câu lệnh Goto trong C++ (Goto statements). Nhìn chung, việc sử dụng câu lệnh Goto bị xa lánh trong C++ (và hầu hết các ngôn ngữ bậc cao khác).

Trong bài tiếp theo, mình sẽ giới thiệu cho các bạn về VÒNG LẶP WHILE TRONG C++ (While statements).

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**".