

# Lệnh rẽ nhánh và lệnh nhảy trong C++

## 1. break

Sử dụng `break` chúng ta có thể thoát khỏi vòng lặp ngay cả khi điều kiện để nó kết thúc chưa được thoả mãn. Lệnh này có thể được dùng để kết thúc một vòng lặp không xác định hay buộc nó phải kết thúc giữa chừng thay vì kết thúc một cách bình thường. Ví dụ, chúng ta sẽ dừng việc đếm ngược trước khi nó kết thúc:

```
// break loop example
#include <iostream.h>
int main ()
{
    int n;
    for (n=10; n>0; n--) {
        cout << n << ", ";
        if (n==3)
        {
            cout << "countdown aborted! ";
            break;
        }
    }
    return 0;
}
```

Kết quả:

```
10, 9, 8, 7, 6, 5, 4, countdown aborted!
```

## 2. continue

Lệnh `continue` làm cho chương trình bỏ qua phần còn lại của vòng lặp và nhảy sang lần lặp tiếp theo. Ví dụ chúng ta sẽ bỏ qua số 5 trong phần đếm ngược:

```
// break loop example
#include <iostream.h>
int main ()
{
    for (int n=10; n>0; n--) {
        if (n==5) continue;
        cout << n << ", ";
    }
    cout << "FIRE! ";
    return 0;
}
```

Kết quả:

```
10, 9, 8, 7, 6, 4, 3, 2, 1, FIRE!
```

### 3. goto

Lệnh này cho phép nhảy vô điều kiện tới bất kì điểm nào trong chương trình. Nói chung bạn nên tránh dùng nó trong chương trình C++. Tuy nhiên chúng ta vẫn có một ví dụ dùng lệnh `goto` để đếm ngược:

```
// goto loop example
#include <iostream.h>
int main ()
{
    int n=10;
    loop: ;
    cout << n << ", ";
    n--;
    if (n>0) goto loop;
    cout << "FIRE! ";
    return 0;
}
```

Kết quả:

```
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, FIRE!
```

### 4. exit

Mục đích của `exit` là kết thúc chương trình và trả về một mã xác định. Dạng thức của nó như sau:

```
void exit (int exit code);
```

`exit code` được dùng bởi một số hệ điều hành hoặc có thể được dùng bởi các chương trình gọi. Theo quy ước, mã trả về 0 có nghĩa là chương trình kết thúc bình thường còn các giá trị khác 0 có nghĩa là có lỗi.

---

Revision #1

Created 5 October 2019 04:17:44 by Laptrinh.vn

Updated 5 October 2019 04:21:46 by Laptrinh.vn