

# Kiểu biến trong C/C++

Một biến cung cấp nơi lưu giữ được đặt tên để chúng ta có thể thao tác. Mỗi biến trong C/C++ có một kiểu cụ thể, mà quyết định: kích cỡ và cách bố trí bộ nhớ của biến; dãy giá trị có thể được lưu giữ bên trong bộ nhớ đó; và tập hợp hoạt động có thể được áp dụng cho biến đó.

Tên biến có thể gồm các ký tự, các chữ số, dấu gạch dưới. Nó phải bắt đầu bởi hoặc một ký tự hoặc một dấu gạch dưới. Các ký tự chữ hoa và chữ thường là khác nhau bởi C/C++ là ngôn ngữ phân biệt kiểu chữ. Biến có thể trong các kiểu giá trị đa dạng như kiểu char, int, float, ...

## Định nghĩa biến trong C/C++

Định nghĩa biến trong C/C++ nghĩa là nói cho compiler nơi và lượng bộ nhớ cần tạo để lưu giữ biến đó. Một định nghĩa biến xác định một kiểu dữ liệu, và chứa danh sách của một hoặc nhiều biến có kiểu đó, như sau:

```
kieu_gia_tri danh_sach_bien;
```

Ở đây, kieu\_gia\_tri phải là một kiểu dữ liệu hợp lệ trong C/C++, gồm char, w\_char, int, float, double, bool hoặc bất kỳ đối tượng nào mà người dùng tự định nghĩa, ... và danh\_sach\_bien có thể chứa một hoặc nhiều tên Identifier (Định danh) phân biệt nhau bởi dấu phẩy. Sau đây là một số khai báo hợp lệ trong C/C++:

```
int j, q, k;
```

```
char v, viet;
```

```
float x, diemthi;
```

```
double luong;
```

Dòng `int j, q, k;` vừa khai báo và định nghĩa các biến `j, q, k`, mà chỉ dẫn compiler để tạo các biến với tên là `j, q, k` với kiểu `int`.

## ADVERTISING

Các biến có thể được khởi tạo (được gán giá trị ban đầu) trong các khai báo. Initializer (phần khởi tạo) gồm một ký hiệu bằng được theo sau bởi một Constant Expression (biểu thức hằng số), như sau:

```
kiểu_gia_trị ten_bien = giaTri;
```

Ví dụ:

```
extern int d = 3, f = 5;    // khai báo biến d và f.
```

```
int d = 3, f = 5;          // định nghĩa và khởi tạo biến d và f.
```

```
byte z = 22;               // định nghĩa và khởi tạo biến z.
```

```
char x = 'vietjack';       // biến x có giá trị là 'vietjack'.
```

Với định nghĩa mà không có phần khởi tạo: các biến được khởi tạo với NULL (tất cả byte có giá trị 0); giá trị khởi tạo của tất cả biến khác không được định nghĩa.

## Khai báo biến trong C/C++

Khai báo biến trong C/C++ chắc chắn với compiler rằng có một biến đang tồn tại với kiểu và tên đã cho, để mà compiler tiếp tục trình biên dịch mà không cần biết đầy đủ chi tiết về biến đó. Một khai báo biến chỉ có ý nghĩa tại thời gian biên dịch, compiler cần khai báo biến thực sự tại thời điểm kết nối chương trình.

Khai báo biến là hữu ích khi bạn đang sử dụng nhiều file, và bạn định nghĩa biến của bạn ở một trong các file đó mà sẽ có sẵn tại thời điểm kết nối chương trình. Bạn sẽ sử dụng từ khóa `extern` để khai báo một biến tại

bất kỳ đâu. Mặc dù, bạn có thể khai báo một biến nhiều lần trong chương trình C/C++, nhưng nó có thể chỉ được định nghĩa một lần trong một file, một hàm, hoặc một khối code.

## Ví dụ

Trong ví dụ sau, một biến được khai báo tại ở phần đầu chương trình, nhưng nó đã được định nghĩa ở bên trong hàm main.

```
#include <iostream>

using namespace std;

// Phan khai bao bien:

extern int a, b;

extern int c;

extern float f;

int main ()
{
    // Phan dinh ngia bien:

    int a, b;

    int c;

    float f;

    // Phan khoi tao bien

    a = 10;

    b = 20;

    c = a + b;
```

```
cout << c << endl ;
```

```
f = 70.0/3.0;
```

```
cout << f << endl ;
```

```
return 0;
```

```
}
```

Khi code trên được biên dịch và thực thi, nó cho kết quả sau:

```
30
```

```
23.3333
```

Giống khái niệm áp dụng trên khai báo hàm, bạn cung cấp tên hàm tại thời điểm khai báo và định nghĩa thực sự của hàm đó có thể được cung cấp ở bất cứ đâu. Ví dụ:

```
// phân khai báo hàm
```

```
int func();
```

```
int main()
```

```
{
```

```
    // phân gọi hàm
```

```
    int i = func();
```

```
}
```

```
// phân định nghĩa hàm
```

```
int func()
```

```
{
```

```
    return 0;
```

```
}
```

## Lvalue và Rvalue trong C/C++

Có hai loại Expression trong C/C++:

- lvalue : Đây là expression liên quan tới một vị trí vùng nhớ. Bất kỳ lvalue có thể xuất hiện ở bên trái hoặc bên phải của một phép gán.
- rvalue : Khái niệm này liên quan tới một giá trị dữ liệu mà được lưu giữ tại một số địa chỉ trong vùng nhớ. Một rvalue là một expression mà không thể có một giá trị được gán tới nó, nghĩa là: rvalue có thể xuất hiện bên phải nhưng không thể xuất hiện bên trái của một phép gán.

Các biến là lvalue có thể xuất hiện bên trái của phép gán. Các Numeric literal (hằng số) là rvalue không thể được gán và không thể xuất hiện ở bên trái của phép gán. Dưới đây là một lệnh hợp lệ trong C/C++:

```
int t = 10;
```

Nhưng lệnh sau là không hợp lệ và sẽ cho một compile-time error (lỗi tại thời điểm biên dịch):

```
16 = 30;
```