Quy tắc phạm vi của một định danh

- Ý nghĩa sử dụng
- Phân loại chi tiết

Ý nghĩa sử dụng

- Định danh là tên của một thành phần, ví dụ như tên biến, tên hàm, các nhãn, dùng để phân biệt với các thành phần khác
- Phạm vi của một định danh là phần của chương trình trong đó định danh tồn tại và được sử dụng
- Hiểu biết về phạm vi sử dụng của một định danh giúp bạn thông thạo hơn khi nhìn nhận và giải quyết vấn đề trong một chương trình

Phân loại cụ thể

Có 4 phạm vi của một định danh:

- Phạm vi hàm
- Phạm vi file
- Phạm vi khối
- Phạm vi nguyên mẫu hàm

Phạm vi hàm

- Các nhãn ví dụ như nhãn sau các case của cấu trúc switch là các định danh chỉ sử dụng trong phạm vi hàm
- Các nhãn này có thể sử dụng ở bất cứ đâu trong thân hàm mà nó xuất hiện
- Ra khỏi hàm không sử dụng được chúng nữa vì các định danh này đã bị ẩn bên trong các hàm mà nó được định nghĩa

Phạm vi file

- Các định danh khai báo bên ngoài tất cả các hàm có phạm vi file
- Tức là chúng có thể được nhìn thấy và sử dụng bởi bất kì hàm nào trong file đó kể từ sau khi nó được khai báo đến hết file
- Các biến toàn cục, các định nghĩa hàm, hàm nguyên mẫu là các định danh có phạm vi file
- Ví dụ:

```
// bien toan cuc:
int x = 5;

void area(int w, int h);
void showX();
void swap(int a, int b);
```

Phạm vi khối

- Các định danh khai báo trong khối giới hạn bởi cặp ngoặc {} có phạm vi khối
- Ví dụ biến khai báo trong khối if, khối for
- Các biến này chỉ tồn tại và được nhìn thấy bởi các thành phần trong cùng khối, các thành phần bên trong khối lồng trong khối hiện tại kể từ sau khi nó được khai báo

Phạm vi khối

- Nếu khối bên trong chứa biến cùng tên khối bên ngoài thì biến bên ngoài sẽ bị ẩn đi đến khi khối bên trong kết thúc
- Ví dụ:

```
if (a != b) { // pham vi khoi
        int tmp = a;
        a = b;
        b = tmp;
}
```

Phạm vi nguyên mẫu hàm

- Các định danh khai báo trong nguyên mẫu hàm chỉ có phạm vi trong nguyên mẫu hàm mà nó được khai báo
- Trình biên dịch sẽ bỏ qua chúng khi biên dịch chương trình
- Chúng có thể được tái sử dụng ở bất cứ đâu trong chương trình mà không hề gây ảnh hưởng gì
- Ví dụ:

```
void area(int w, int h);
void showX();
void swap(int a, int b);
```

Tiếp theo

Hàm đệ quy