ĐẶT TÊN BỊ XUNG ĐỘT VÀ GIỚI THIỆU VỀ NAMESPACES

C++ yêu cầu tất cả các định danh không được mơ hồ. Nếu hai ký tự nhận dạng giống hệt nhau được đưa vào cùng một chương trình theo cách mà trình biên dịch hoặc trình liên kết không thể phân biệt chúng, trình biên dịch hoặc trình liên kết sẽ tạo lỗi. Lỗi này thường được gọi là xung đột đặt tên.

**1. Ví dụ về xung đột đặt tên**



main.cpp



Khi trình biên dịch biên dịch chương trình này, nó sẽ biên dịch a.cpp và main.cpp một cách độc lập, và mỗi tệp sẽ biên dịch không có vấn đề gì.

Tuy nhiên, khi trình liên kết thực thi, nó sẽ liên kết tất cả các định nghĩa trong a.cpp và main.cpp với nhau, đồng thời phát hiện ra các định nghĩa xung đột cho hàm myFcn. Trình liên kết sau đó sẽ hủy bỏ do lỗi. Lưu ý rằng lỗi này xảy ra mặc dù myFcn không bao giờ được gọi!

Hầu hết các xung đột đặt tên xảy ra trong hai trường hợp:

1) Hai (hoặc nhiều) định nghĩa cho một hàm (hoặc biến toàn cục) được đưa vào các tệp riêng biệt được biên dịch vào cùng một chương trình. Điều này sẽ dẫn đến lỗi trình liên kết, như được hiển thị ở trên.

2) Hai (hoặc nhiều) định nghĩa cho một hàm (hoặc biến toàn cục) được đưa vào cùng một tệp (thường thông qua #include). Điều này sẽ dẫn đến lỗi trình biên dịch.

Khi các chương trình ngày càng lớn hơn và sử dụng nhiều ký hiệu nhận dạng hơn, khả năng xảy ra va chạm đặt tên được đưa vào sẽ tăng lên đáng kể. Tin tốt là C ++ cung cấp rất nhiều cơ chế để tránh va chạm đặt tên. Phạm vi cục bộ, giữ cho các biến cục bộ được xác định bên trong các hàm khỏi xung đột với nhau, là một trong những cơ chế như vậy. Nhưng phạm vi cục bộ không hoạt động đối với tên hàm. Vậy làm cách nào để giữ cho các tên hàm không bị xung đột với nhau?

**2. namespace là gì?**

**Namespace là một vùng cho phép bạn khai báo các tên bên trong nó nhằm mục đích phân định, namespace cùng cấp một phạm vi (được gọi là phạm vi của namespace) cho các tên được khai báo bên trong nó – điều này đơn giản có nghĩa là bất kỳ tên nào được khai báo bên trong namespace sẽ không bị nhầm với tên giống hệt nahu trong các phạm vi khác**.

**Tên được khai báo trong namespace sẽ không bị nhầm lẫn với tên giống được khai báo trong phạm vi khác.**

Trong một namespace, tất cả các tên phải là duy nhất, nếu không sẽ dẫn đến xung đột đặt tên.

namespace thường được sử dụng để nhóm các số nhận dạng có liên quan trong một dự án lớn để giúp đảm bảo chúng không vô tình xung đột với các số nhận dạng khác. Ví dụ: nếu bạn đặt tất cả các hàm toán học của mình trong một namespace gọi là math, thì các hàm toán học của bạn sẽ không xung đột với các hàm có tên giống hệt nhau bên ngoài namespace math.

Chúng ta sẽ nói về cách tạo không gian tên của riêng bạn trong một bài học trong tương lai.

**3. namespace toàn cục**

Trong C++, bất kỳ tên nào không được xác định bên trong một lớp, hàm hoặc một namspace được gọi là một phần của namespace được xác định ngầm được gọi là namespace toàn cục (đôi khi được gọi là phạm vi toàn cục).

Trong ví dụ ở đầu bài học, các hàm main() và cả hai phiên bản của myFcn() đều được định nghĩa bên trong namespace toàn cục. Xung đột đặt tên gặp phải trong ví dụ xảy ra vì cả hai phiên bản của myFcn() đều nằm trong vùng namespace toàn cục, vi phạm quy tắc rằng tất cả các tên trong vùng tên phải là duy nhất.

**4. namespace std**

Khi C++ ban đầu được thiết kế, tất cả các mã định danh trong thư viện chuẩn C++ (bao gồm std :: cin và std :: cout) đều có sẵn để sử dụng mà không có tiền tố std :: (chúng là một phần của namespace toàn cục).