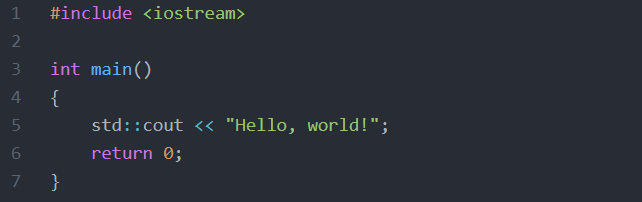
**Tìm hiểu về file Header trong C++**

**1. Headers file và mục đích của nó**

Khi các chương trình phát triển ngày càng lớn (và sử dụng nhiều file hơn), việc khai báo các hàm bạn muốn sử dụng được định nghĩa trong cùng một file trở nên ngày càng tẻ nhạt và làm cho file đó trở nên nhiều dòng code, khó kiểm soát. Sẽ tốt hơn nếu bạn có thể đặt tất cả các khai báo của mình ở một nơi nào đó sau đó định nghĩa chúng khi bạn cần nó.

Các file code C++ (có phần mở rộng là .cpp) không phải là các file duy nhất. Thường thấy trong các chương trình C++. Có một loại file khác được gọi là file header. **Các file header thường có phần mở rộng là .h, nhưng đôi khi bạn sẽ thấy chúng có phần mở rộng là .hpp hoặc không có phần mở rộng nào cả. Mục đích chính cảu file header là giúp chúng ta khai báo một tên của một hàm hoặc biến nào đó**….

**2. Sử dụng header file của thư viện chuẩn**



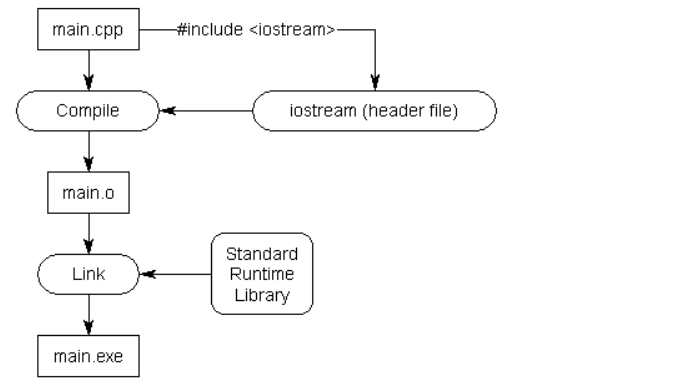
Chương trình này in ra Hello World! Ra console bằng std :: cout. Tuy nhiên, chương trình này không bao giờ cung cấp định nghĩa hoặc khai báo cho std :: cout, vậy làm thế nào để trình biên dịch biết std :: cout là gì?

Câu trả lời là std :: cout đã được khai báo trong file header của iostream. **Khi chúng ta #include, chúng ta sẽ yêu cầu bộ tiền xử lý sao chép tất cả nội dung (bao gồm cả khai báo cho std :: cout) từ file có tên là iostream, vào file mà chúng ta đã include nó.**

Khi bạn #include một file, nội dung của file sẽ được chèn tại thời điểm nó được include. Điều này cung cấp một cách hữu ích để lấy các khai báo từ một file khác.

Xem xét những gì sẽ xảy ra nếu header của iostream không tồn tại. Bất cứ nơi nào bạn đã sử dụng std :: cout, bạn sẽ phải tự code hoặc sao chép thủ công tất cả các khai báo liên quan đến std :: cout vào đầu mỗi file đã sử dụng std :: cout! Điều này sẽ đòi hỏi nhiều kiến ​​thức về std :: cout và sẽ là tốn khá nhiều công. Tệ hơn nữa, nếu một nguyên mẫu hàm thay đổi, chúng tôi phải cập nhật thủ công tất cả các khai báo đó. Nhưng dễ dàng hơn rất nhiều khi chỉ cần #include iostream

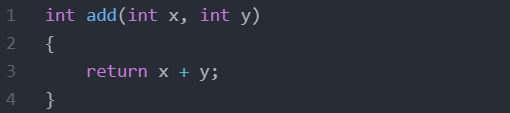
**Khi nói đến các hàm và đối tượng, lưu ý rằng các file header thường chỉ chứa các khai báo hàm và đối tượng, không phải là các định nghĩa hàm và đối tượng**. **std :: cout được khai báo trong header của iostream, nhưng được định nghĩa là một phần của thư viện chuẩn C ++, được liên kết tự động vào chương trình của bạn trong giai đoạn liên kết.**



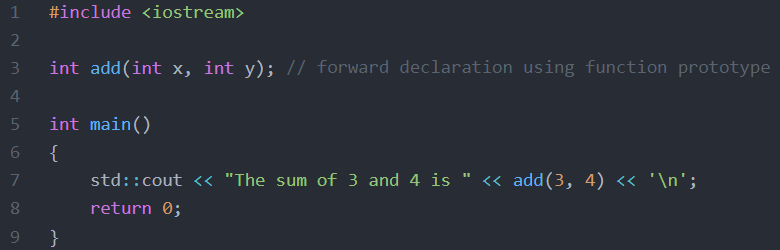
**3. Viết một file header**

Bây giờ, hãy trở lại ví dụ mà chúng ta đã thảo luận trong bài học trước. Khi chúng ta rời đi, chúng ta có hai tệp, add.cpp và main.cpp, trông giống như thế này.

add.cpp:



main.cpp



Trong ví dụ này, chúng ta đã sử dụng một khai báo để trình biên dịch sẽ biết định danh add là gì khi biên dịch main.cpp. Như đã đề cập trước đây, việc thêm thủ công cho mọi hàm bạn muốn sử dụng trong một file khác khá nhiều và phức tạp.

Hãy viết một file header để giảm bớt gánh nặng này. Viết một file header rất dễ dàng, vì các file header chỉ bao gồm hai phần:

* Các tiền chỉ thị trong file header.
* Nội dung thực tế của file header, phải là khai báo cho tất cả các định danh mà chúng ta muốn các file khác có thể nhìn thấy.

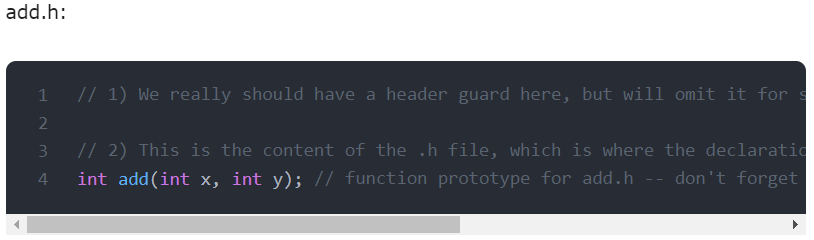
Thêm một file header vào một dự án hoạt động tương tự như thêm một file source. Nếu sử dụng IDE, hãy thực hiện các bước tương tự và chọn loại Header thay vì Source khi được hỏi. Nếu sử dụng dòng lệnh, chỉ cần tạo một file mới trong trình chỉnh sửa yêu thích của bạn.

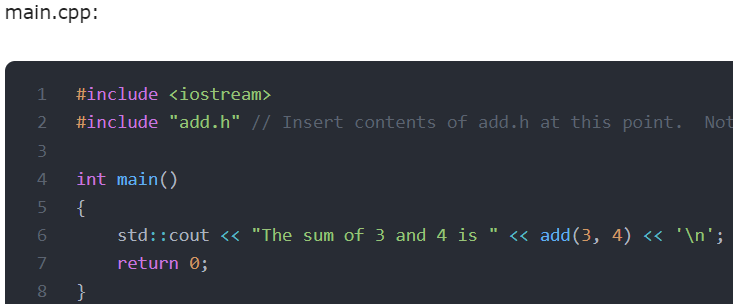
**Sử dụng hậu tốt .h khi đặt tên file header của bạn**

Các file header thường được ghép nối với các file code, với file header cung cấp các khai báo cho cac file code tương ứng. Vì file header của chúng ta sẽ chứa một khai báo cho các hàm được định nghĩa trong add.cpp, nên chúng ta sẽ gọi file header mới là add.h

Nếu một file header được ghép nối với một file code (Ví dụ: add.h với add.cpp), cả hai sẽ cùng có tên (add).

File header như sau:





add.cpp:

