**Biến (variables) trong C++ là gì?**

**1. Dữ liệu (Data)**

Trong bài trước về các câu lệnh và cấu trúc của một chương trình, bạn đã biết được phần lớn các code trong một chương trình là câu lệnh và các câu lệnh được nhóm thành các hàm. Các câu lệnh này thực hiện hanh động để tạo ra bất kỳ kết quả nào từ chương trình với yêu cầu nào đó.

Nhưng làm thế nào để các chương trình thực sự tạo ra kết quả nào đó? **Thực chất thì để làm được như vậy thì chúng ta cần phải có các thao tác (đọc, thay đổi và viết) dữ liệu. Trong máy tính, dữ liệu là bất kỳ thông tin nào có thể được di chuyển, xử lý hoặc lưu trữ bởi máy tính**.

**Chú ý: Các chương trình là tập hợp các dòng code thao tác dữ liệu để tạo ra kết quả mong muốn.**

Một chương trình có thể thu thập dữ liệu để làm việc theo nhiều cách như: **từ file hoặc cơ sở dữ liệu, qua mạng,** t**ừ người dùng cung cấp bằng cách nhập thông tin bằng bàn phím hoặc từ lập tình viên đưa trực tiếp các dữ liệu vào code của chính chương trình.** Trong chương trình Hello World đã học thì chuỗi Hello World! Đã được chèn trực tiếp vào code cảu chương trình nhằm cung cấp dữ liệu cho chương trình sử dụng. Và sau đó chương trình sẽ thao tác với dữ liệu này bằng cách gửi nó đến màn hình hiển thị.

**2. Đối tượng và biến (Objects and variables)**

Tất cả các máy tính đều có bộ nhớ, được gọi là RAM (viết tắt của bộ nhớ truy cập ngẫu nhiên), có sẵn để sử dụng cho các chương trình của bạn. Bạn có thể nghĩ RAM là một loạt các hộp thư có thể được sử dụng để giữ dữ liệu trong khi chương trình đang chạy. Và một dữ liệu nào đó được lưu trữ trong bộ nhớ ở đâu đó, được gọi là một giá trị.

**Trong một số ngôn ngữ lập trình cũ hơn (như Apple Basic), bạn có thể truy cập trực tiếp vào các hộp thư này**.

**Trong C++, không cho phép truy cập trực tiếp vào bộ nhớ. Thay vào đó, chúng ta truy cập bộ nhớ gián tiếp thông qua một đối tượng. Một đối tượng là một vùng lưu trữ (thường là bộ nhớ) có giá trị và các thuộc tính liên quan khác (chi tiết về đối tượng, chúng ta sẽ đề cập trong các bài học trong tương lai)**.

**Khi một đối tượng được khai báo, trình biên dịch sẽ tự động xác định vị trí của đối tượng sẽ được đặt trong bộ nhớ.** **Kết quả là, thay vì nói hãy lấy giá trị được lưu trong hộp thư số nào đó, chúng ta có thể nói, hãy lấy giá trị được lưu trữ bởi đối tượng này và trình biên dịch biết vị trí trong bộ nhớ để tìm giá trị đó. Điều này có nghĩa là chúng ta có thể tập trung vào việc sử dụng các đối tượng để lưu trữ và truy xuất các giá trị, và không phải lo lắng về vị trí mà chúng thực sự được đặt trong bộ nhớ.**

**Các đối tượng có thể được dặt tên hoặc không tên (ẩn danh).** Một đối tượng được đặt tên được gọi là một biến và tên của đối tượng được gọi là định danh. Trong các chương trình của chúng ta, hầu hết các đối tượng chúng ta tạo sẽ là các biến.

**3. Khởi tạo biến**

Một đối tượng được khởi tạo đôi khi cũng được gọi là một thể hiện (instance)

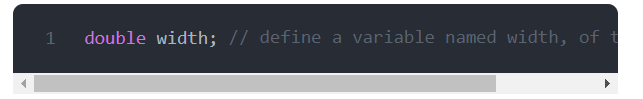
**4. Các kiểu dữ liệu**

Chúng ta đã nói về các biến là một vùng lưu trữ và được đặt tên, có thể lưu trữ một giá trị dữ liệu. Một kiểu dữ liệu (thường được gọi là type) cho trình biên dịch biết biến đó thuộc kiểu giá trị nào (Ví dụ: một số, một chữ cái, văn bản, v.v.).

Trong ví dụ trên, biến x của chúng ta được đưa ra kiểu int, có nghĩa là biến x sẽ đại diện cho một giá trị nguyên. Một số nguyên là một số có thể được viết mà không có thành phần phân số, chẳng hạn như 4, 27, 0, -2 hoặc -12. Nói tóm lại, chúng ta có thể nói rằng x là một biến số nguyên.

Trong C++, kiểu dữ liệu cho biến phải được biết tại thời gian biên dịch (khi chương trình được biên dịch) và kiểu đó không thể thay đổi nếu không biên dịch lại chương trình. Điều này có nghĩa là một biến số nguyên chỉ có thể giữ các giá trị nguyên. Nếu bạn muốn lưu trữ một loại giá trị khác, bạn sẽ cần sử dụng một biến khác.

Số nguyên chỉ là một trong nhiều kiểu mà C++ hỗ trợ. Đối với minh họa ở đây về một ví dụ khai báo một biến sử dụng kiểu dữ liệu double.



C ++ cũng cho phép bạn tạo [các kiểu dữ liệu](https://cafedev.vn/tu-hoc-c-gioi-thieu-ve-cac-kieu-du-lieu-co-ban-trong-c/) do người dùng định nghĩa ra. Đây là điều mà chúng ta sẽ làm rất nhiều trong các bài học tới và nó là một phần của những gì làm cho C ++ trở nên mạnh mẽ.

Chúng ta sẽ gắn bó với các biến số nguyên vì chúng đơn giản về mặt khái niệm, nhưng chúng ta sẽ khám phá nhiều kiểu dữ kiệu khác trong các bài tiếp theo.

**5. Khái báo nhiều biến**

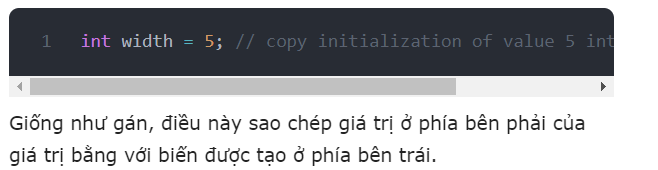
***tránh khai báo nhiều biến trong một câu lệnh (ngay cả khi chúng cùng kiểu). Thay vào đó, hãy khai báo từng biến trong một câu lệnh riêng (và sau đó sử dụng một comment một dòng để ghi lại những gì nó được sử dụng cho cái gì đó).***

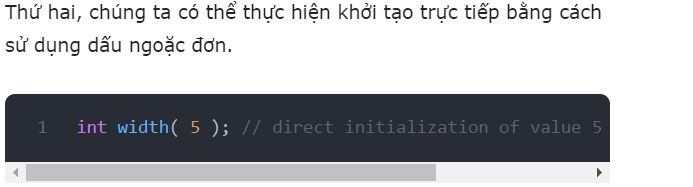
**6. Các cách khởi tạo biến**

**Một nhược điểm của phép gán là nó yêu cầu ít nhất hai câu lệnh: một để khai báo biến và một để gán giá trị cho biến đó.**

**Hai bước này có thể được kết hợp. Khi một biến được khai báo, bạn cũng có thể cung cấp một giá trị ban đầu cho biến đó tại ngay thời điểm khai báo. Điều này được gọi là khởi tạo.**

**C++ hỗ trợ ba cách cơ bản để khởi tạo một biến. Đầu tiên, chúng ta có thể thực hiện sao chép bằng cách sử dụng dấu bằng (=):**

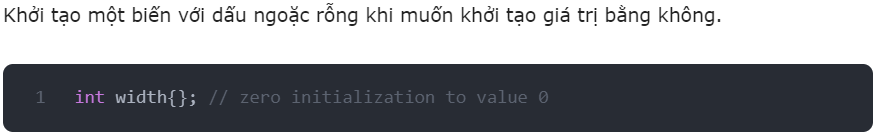




Đối với các kiểu dữ liệu đơn giản (như số nguyên), sao chép và khởi tạo trực tiếp về cơ bản là giống nhau. **Nhưng đối với một số loại nâng cao, khởi tạo trực tiếp có thể là cách tốt hơn so với khởi tạo bằng cách gán lại sau. Trước C++11, khởi tạo trực tiếp được khuyến nghị so với khởi tạo sau trong hầu hết các trường hợp vì tăng hiệu suất**.

<https://cafedev.vn/tu-hoc-c-cach-gan-thay-doi-gia-tri-cua-bien-va-khoi-tao-no/>

**KHỞI TẠO BẰNG DẤU NGOẶC NHỌN**



**Khởi tạo ngoặc nhọn có thêm lợi ích từ việc không cho phép chuyển đổi kiểu của giá trị đang gán với kiểu hiện tại của biến. Điều này có ý nghĩa nếu bạn cố gắng sử dụng dấu ngoặc để khởi tạo một biến có gái trị mà nó không phù hợp với kiểu của biến đó thì trình biên dịch sẽ đưa ra cảnh báo hoặc lỗi.**



Trong đoạn trích trên, chúng ta đang cố gắng gán một số (4.5) có một phần thân số (phần 0.5) cho một biến số nguyên (chỉ có thể giữ các số mà không có các phần phân số). **Sao chép và khởi tạo trực tiếp sẽ bỏ phần phân số, dẫn đến khởi tạo giá trị width bằng 4. Tuy nhiên, khởi tạo với dấu ngoặc sẽ khiến trình biên dịch gặp lỗi(nói chung alf một điều tốt, vì mất dữ liệu hiếm khi được mong muốn).**

**Ưu tiên khởi tạo trực tiếp bằng dấu ngoặc bất cứ khi nào có thể.**