

# LỜI MỞ ĐẦU

Chào mừng các bạn đến với bài học lập trình C/C++ .

Tôi sẽ là người hướng dẫn (hay là thầy giáo nếu bạn thích 😊) trong suốt các bài học.

Vậy tôi là ai? Tên tôi, hay là nickname của tôi là M@teo21. Tôi đã từng viết rất nhiều bài hướng dẫn khác trên <http://www.siteduzero.com>, tôi cũng chính là người đã tạo nên trang web này.

Và đây không phải là lần đầu tiên tôi viết những bài hướng dẫn cơ bản 😊.  
Nhưng khoan hãy nói về tôi đã, hãy nói về chính bản thân các bạn.

Bạn không hề biết tí gì về lập trình.

Cũng không chắc đã biết “lập trình” là gì nhưng... chắc chắn là bạn đang muốn học lập trình đúng không?

Và bạn phải xác định chính xác mục tiêu của mình là: “học lập trình”.

Nhưng lập trình C / C++... Đó nghĩa là gì?

Và có thật sự tốt nếu ta bắt đầu học từ nó?

Và bạn đã biết lập trình trước đó chưa?

Có phải chúng ta có thể làm tất cả mọi thứ trên máy tính với nó?

Nhiệm vụ quan trọng của chương này là trả lời tất cả những câu hỏi đơn giản như thế.

---

# CHƯƠNG I – NHỮNG ĐIỀU CƠ BẢN VỀ NGÔN NGỮ LẬP TRÌNH C

## 1) "lập trình" là gì?

- Lập trình là gì ?
- Lập trình bằng ngôn ngữ lập trình nào ?
- Lập trình có khó không ?
- Trắc Nghiệm Kiến Thức.

## 2) Công cụ cần có để học lập trình

- Những công cụ cần thiết cho lập trình.
- Các bạn có thể chọn... Dev-C++
- Hay là... Visual C++
- Và cái này nữa... Code::Blocks.
- Hoặc Mac... Xcode.

## 3) Chương trình đầu tiên của bạn

- Console hay là cửa sổ ?
- Những dòng code tối thiểu cần phải có.
- Viết một tin nhắn ra màn hình.
- Những chú thích, vô cùng tiện dụng !

## 4) Thế giới của những biến số

- Công việc của bộ nhớ.
- Cách khai báo một biến.
- Hiển thị giá trị của biến số.
- Cách gán giá trị vào biến số.

## 5) Công cụ tính toán

- Những tính toán cơ bản.
- Phương pháp viết rút gọn.
- Thư viện toán học.
- Trắc Nghiệm Kiến Thức.

## 6) Conditions (điều kiện)

- Condition "if... else".
- Boolean, trung tâm của những condition.
- Condition "switch".
- Ternary : những conditions rút gọn.

## 7) Loops (vòng lặp)

- Thế nào là vòng lặp.
- Vòng lặp while.
- Vòng lặp do... while.
- Vòng lặp for.

## 8) Test Program: Hơ hay kém, trò chơi đầu tiên của bạn

- Chuẩn bị và một vài gợi ý.
- Đáp án.
- Ý tưởng cải tiến.

## 9) Function

- Cách tạo và gọi một function.
- Xem thêm vài ví dụ để hiểu rõ hơn.

## Bài 1: Lập trình là gì?

Chúng ta bắt đầu từ một câu hỏi đơn giản nhất có thể. 😊

Nếu bạn đã biết điều này trước đó, tôi vẫn khuyên bạn nên đọc lại nó. Tôi sẽ bắt đầu bài hướng dẫn từ zero, dành một người không biết tí gì về lập trình. 😊

❓ “lập trình” (programmer, program) nghĩa là gì?

Vâng tôi sẽ tránh làm giống như thầy giáo dạy văn của tôi. Tôi sẽ không đưa ra cho bạn nguồn gốc của từ lập trình (programmer, program). Nhưng dù sao đi nữa nó xuất phát từ một từ latin "programmeus". 😊

Nói đơn giản, lập trình nghĩa là tạo nên những "chương trình máy tính". Những chương trình đòi hỏi máy tính thực hiện tất cả công việc.

**Máy tính bạn chứa đầy các chương trình ở tất cả mọi thể loại:**

- Calculator chính là một chương trình.
- Các công cụ xử lý văn bản của bạn cũng là chương trình.
- Các software dùng để “chat” cũng là chương trình.
- Trò chơi điện tử cũng là chương trình.

Tóm lại, những chương trình ở khắp mọi nơi trên máy tính và cho phép thực hiện bất cứ điều gì.

Bạn có thể tạo ra một chương trình mang tính cách mạng nếu bạn may mắn, hoặc thực hiện một game đánh nhau 3D trên internet. Máy tính của bạn có thể làm tất cả (trừ những việc như làm ra café) 😊



*Trò chơi nổi tiếng Half-life 2, được lập trình bằng C++*

⚠ Xin lưu ý rằng tôi không nói người ta làm ra trò chơi này hoàn toàn chỉ thông qua việc đánh máy viết code. Ý tôi là chúng ta có thể làm những điều đó, nhưng chắc chắn là ngoài việc gõ phím ra, bạn còn có rất nhiều việc khác để làm.

Bạn sẽ không bắt đầu học lập trình bằng việc tạo ra một game 3D. Chẳng khác nào tự mình kết thúc tất cả 😊. Chúng ta hãy bắt đầu từ những điều cơ bản. Đầu tiên là làm sao *hiển thị lên màn hình một tin nhắn*.

Bạn phải học từng thứ từng thứ một, và từ từ bạn sẽ có khả năng thực hiện những chương trình với độ khó ngày càng gia tăng. Mục đích của toàn bộ bài hướng dẫn này giúp bạn có khả năng xoay sở trên bất kì chương trình nào được viết bằng ngôn ngữ C hay C++

## Lập trình bằng ngôn ngữ nào?

Thật sự mà nói, máy tính đúng là một cỗ máy kì lạ: nó chỉ nhận và gửi lại những số 0 và 1.  
Ví dụ, nếu dịch câu: “thực hiện phép tính  $3 + 5$ ” về ngôn ngữ máy tính, nó sẽ có dạng như sau:  
0010110110010011010011110.

(dãy số trên do tôi chế ra đấy 🤖, thật sự thì tôi không giỏi việc dịch ra ngôn ngữ máy tính 😊)

Những số mà bạn thấy ở trên, là ngôn ngữ của máy tính, gọi là **ngôn ngữ nhị phân** (language binary). Máy tính của bạn chỉ hiểu được ngôn ngữ này. Nhưng bạn và tôi, chúng ta hoàn toàn không thể hiểu và học được ngôn ngữ đó. 🤔

Và đây chính là vấn đề đầu tiên của chúng ta:

❓ **Làm cách nào để giao tiếp với máy tính đơn giản hơn việc dùng những số 0 hay 1?**

Máy tính của bạn không nói được tiếng Anh cũng như tiếng Việt. Và không ai quan niệm phải viết một chương trình bằng ngôn ngữ nhị phân, kể cả những nhà lập trình điên nhất cũng không làm chuyện đó. 🤖

Ý tưởng là phải tạo ra một ngôn ngữ mới đơn giản hơn và sau đó nó sẽ được chuyển sang ngôn ngữ nhị phân. Đây là việc của những nhà lập trình chuyên về ngôn ngữ. Và các chương trình trình này đã được tạo ra bởi họ, chúng ta sẽ không cần phải thực hiện lại, thật là may mắn phải không?



Nói đơn giản:

Nếu bạn viết một lệnh bằng ngôn ngữ lập trình nào đó với nội dung:

“thực hiện phép tính  $3+5$ ”

thì chương trình dịch sẽ chuyển thành những dạng như sau:

"0010110110010011010011110"

Tôi sẽ lập một biểu đồ để giúp bạn hiểu rõ hơn:



Biểu đồ cực kì đơn giản về việc thi hành một chương trình 🤖

Ở đây tôi chỉ dùng những từ ngữ đơn giản để giải thích, nhưng trong tin học mỗi vấn đề đều có một thuật ngữ riêng.

Suốt các bài học, bạn sẽ phải học không ít những thuật ngữ đó. Điều đó giúp bạn có thể dễ dàng biểu đạt những vấn đề về tin học, hơn nữa, bạn có thể hiểu được ý của một nhà lập trình nào đó mà bạn sẽ trò chuyện sau này. Hẳn là lúc đó, những người xung quanh sẽ nhìn các bạn với ánh mắt khác thường, đó là lý do mà bạn phải chú ý đến việc học những thuật ngữ 😊

Quay lại với biểu đồ ở trên:

Trong ô đầu tiên: “Chương trình được viết bằng ngôn ngữ lập trình đơn giản”. Cụm từ “Ngôn ngữ lập trình đơn giản” còn được gọi là “**ngôn ngữ bậc cao**”. (*high-level programming language*).

Có rất nhiều “cấp bậc” trong ngôn ngữ lập trình. Và ngôn ngữ bậc càng cao càng gần và giống với ngôn ngữ của chúng ta (cũng giống như tiếng Việt hay tiếng Anh). Ngôn ngữ bậc cao giúp ta dễ dàng hơn trong sử dụng, nhưng nó vẫn có một vài thiếu sót mà bạn sẽ thấy về sau. Có rất nhiều ngôn ngữ bậc cao hay thấp trong tin học, trong số đó bạn có thể dùng để lập trình.

Và đây là một vài ví dụ:

- C
- C++
- Java
- Visual Basic
- Delphi
- ...vv...

Nói thêm rằng ở đây tôi không sắp xếp chúng theo “cấp bậc của ngôn ngữ”, vì thế bạn đừng nghĩ rằng ngôn ngữ đầu tiên sử dụng dễ dàng hơn hay ngược lại. 😊 Đó chỉ là một vài ví dụ bất chợt nảy ra trong đầu của tôi.

(Còn rất nhiều ngôn ngữ khác nữa mà tôi không liệt kê ra hết, vì sẽ rất dài nếu ghi hết ra 😊, xin các bạn bỏ qua cho).

Một số ngôn ngữ có bậc cao hơn các ngôn ngữ khác (về mặt lý thuyết thì dễ dàng sử dụng hơn), chúng ta sẽ xem xét điều này sau, đặc biệt là sự khác nhau giữa ngôn ngữ C và C++.

Một từ ngữ khác mà ta phải nắm đó là: **mã nguồn** (code source). Đó đơn giản là phần mã của chương trình được viết bằng ngôn ngữ bậc cao. Và tất cả những mã nguồn đó sẽ được dịch thành ngôn ngữ nhị phân.

Trong giai đoạn tiếp theo, “chương trình biên dịch” sẽ dịch ngôn ngữ bậc cao đó (C hay C++) sang nhị phân. Chương trình này có tên là **compiler**. Việc biên dịch gọi là **compilation**.

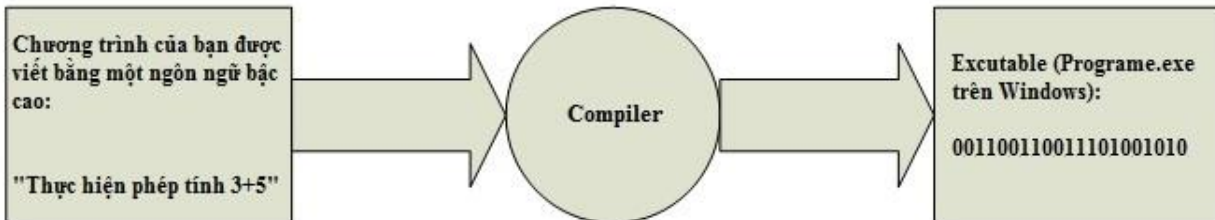


**Quan trọng:** các ngôn ngữ lập trình bậc cao khác nhau sẽ có các compiler khác nhau. Nói cách khác, chúng ta không thể dịch ngôn ngữ C++ cùng với cách mà ta dịch ngôn ngữ Delphi.

⚠ Về sau bạn sẽ nhận thấy, cùng một ngôn ngữ sẽ có nhiều **compilers** khác nhau (compiler của Microsoft, compiler GNU... tôi sẽ nói về chúng ở những chương sau). Rất may mắn là những compiler đó gần như giống nhau (đôi khi chúng có những khác biệt nhỏ và tôi sẽ chỉ cho bạn).

Cuối cùng, chương trình nhị phân được tạo ra bởi compiler được gọi là: **executable**. Các chương trình này trên Windows có đuôi ".exe" giống như EXEcutable.

Quay lại với biểu đồ vừa rồi nhưng thay bằng những thuật ngữ tin học chính xác:



Cùng một biểu đồ nhưng biểu đồ này dùng các thuật ngữ chính xác hơn.

## Tại sao chọn học C/C++?

Như tôi đã nói với bạn ở trên, có rất nhiều ngôn ngữ bậc cao. Chúng ta sẽ bắt đầu bằng một trong số đó.

Nhưng mà bạn phải có một sự lựa chọn giữa:

- **Một ngôn ngữ lập trình có bậc rất cao:** dễ dàng sử dụng, "thông dụng", như Visual basic. Nhưng các ngôn ngữ này có rất nhiều khuyết điểm: đầu tiên là phải mua bản quyền để sử dụng, giá thành mắc, và bị nhiều hạn chế. Ví dụ, chương trình mà bạn viết sẽ chỉ chạy được trên Windows, đừng nghĩ đến việc chạy nó trên Linux hay Macintosh! Hay trên hết, bạn không có thể làm tất cả những gì bạn muốn với dạng ngôn ngữ này, và điều đó khiến bạn cảm thấy hạn chế khi sử dụng.
- **Một ngôn ngữ khác ở vị trí tương đối thấp hơn** (nhưng nó không thấp lắm đâu!): có thể nó sẽ hơi khó hơn visual basic, nhưng chắc chắn một điều là với một ngôn ngữ như C (hay C++) sẽ giúp bạn học thêm rất nhiều trong việc lập trình cũng như hiểu thêm cách hoạt động của máy tính. Và sau đó hoàn toàn đủ khả năng học thêm một vài ngôn ngữ khác nếu bạn muốn. Bạn sẽ tự chủ hơn. Mặt khác, ngôn ngữ C và C++ được sử dụng khá rộng rãi. Nó được dùng để lập trình phần lớn các chương trình bạn biết. Cuối cùng, để lập trình trên C hay C++, bạn không cần phải mua bất kì chương trình nào vượt ngoài túi tiền của bạn, vì ngôn ngữ này hoàn toàn miễn phí!

Và đó là lý do thúc đẩy tôi hướng dẫn bạn ngôn ngữ C trước tiên. Tôi không hề nói rằng chúng ta bắt buộc phải bắt đầu từ đây, nhưng lựa chọn này sẽ giúp bạn có được những kiến thức bền vững về lập trình.

Tôi xem như đây chính là ngôn ngữ lập trình đầu tiên của bạn, và bạn không biết tí gì về lập trình trước đó. Cũng có thể, bạn đã biết lập trình rồi, nhưng việc học lại từ cơ bản không ảnh hưởng xấu tí nào phải không 😊

❓ Khoan đã, có một cái mà tôi vẫn chưa hiểu: Tôi sẽ học một ngôn ngữ gọi là “C / C++” hay tôi sẽ học 2 ngôn ngữ khác nhau một là “C” và một là “C++”?

Câu trả lời tốt nhất là bạn sẽ cùng lúc học cả 2 ngôn ngữ. Không phải là bạn phải tăng cường độ làm việc lên hai lần đâu 😓 mà là 2 ngôn ngữ này khá giống nhau. (Khi tôi nói đến 2 ngôn ngữ cùng lúc, tôi sẽ viết “C / C++”).



Và chúng ta phải hiểu rõ sự khác nhau giữa C và C++ trước khi bắt đầu:

- Giai đoạn đầu, lúc các máy tính có khối lượng tính bằng tấn và có kích thước to như ngôi nhà, người ta đã sáng tạo ra một ngôn ngữ lập trình gọi là **Algol**.
- Sau nhiều cải tiến, người ta đã tạo ra một ngôn ngữ mới gọi là **CPL**, và chính nó đã phát triển thành **BCPL**, sau đó nó được mang tên là **ngôn ngữ B**. (Các bạn không cần phải nắm tắt cả những điều này, tôi viết ra chỉ để có thêm chút đáng vẽ về lịch sử mà thôi 😊).
- Và trong một ngày đẹp trời, người ta đã hoàn tất việc tạo ra một ngôn ngữ mới gọi là ... **ngôn ngữ C**. Qua các sửa đổi, ngôn ngữ này vẫn là một trong những ngôn ngữ được sử dụng nhiều nhất cho đến hôm nay.
- Không lâu sau đó, người ta đề xuất thêm vào ngôn ngữ C một vài thứ để cải tiến, và ngôn ngữ mới này được gọi là **ngôn ngữ C++** hoàn toàn dựa trên nền tảng của C. Ngôn ngữ C++ không có gì khác C ngoại trừ một số cái được thêm vào. (Những cái đó là gì? Chúng ta sẽ thấy về sau).

Có rất nhiều cách để học lập trình. Nhiều người nghĩ rằng tốt hơn hết là học ngay từ “C++”. Điều đó cũng đúng, chúng tương đối giống nhau, C++ chỉ là C được thêm vào “những dấu +”. Ngôn ngữ C không phải là “ngôn ngữ già cỗi bị lãng quên”, ngược lại nó được sử dụng rất nhiều trong thời đại hiện nay. Nó là nền tảng của những hệ điều hành lớn như Unix hay Windows.

Nếu bạn bắt đầu bằng ngôn ngữ C, sau này bạn học ngôn ngữ C++ sẽ nhanh và dễ dàng hơn. Và bạn sẽ không cần phải học lại tất cả những gì đã biết, bạn chỉ cần học những cái được thêm vào ở “C++” (và đây chính là điều tôi cần nói 😊).

⚠️ Có một số hiểu lầm khi có người cho rằng *ngôn ngữ C++ tốt hơn C*, thật ra thì nó chỉ cho phép bạn lập trình theo cách khác. Có thể nói, nó chỉ giúp việc lập trình của ta nhanh hơn và dễ dàng hơn trong việc tổ chức mã nguồn của chương trình.

**Nắm vững vấn đề:** C và C++ không phải là 2 ngôn ngữ cạnh tranh, đối lập với nhau. Chúng ta đều có thể dùng 2 ngôn ngữ này lập trình những cái tương tự. Chỉ là dùng 2 phương pháp lập trình khác nhau.

Điều thuận lợi hơn là sau này có thể dùng C hay C++ tùy theo ý muốn và mục đích của các bạn.

## Lập trình có khó không ?

Đây chính là câu hỏi khiến bạn phải suy nghĩ nhiều đúng không? 😊

Và có phải chúng ta bắt buộc phải là một nhà toán học cực giỏi sau nhiều năm học tập để có thể bắt đầu với việc lập trình?

Câu trả lời khiến bạn yên tâm hơn là **điều đó không đúng**.

Bạn không cần phải có một đẳng cấp toán học thật cao. Những kiến thức bạn cần để bắt đầu học chính là:

- Phép cộng (Tôi hi vọng bạn đã nắm vững nó 😊)
- Phép trừ (híc... híc... )
- Phép nhân (😬...)
- Phép chia (😬)

Tôi hi vọng các bạn đã biết tất cả những phép tính đó. 😊 Và chắc chắn là tôi sẽ giải thích cho bạn trong phần tiếp theo, làm sao máy tính có thể thực hiện các phép tính cơ bản.

Tóm lại, về toán học thì không có gì khó khăn để bạn vượt qua. 😊

Tất cả chỉ phụ thuộc vào chương trình mà bạn muốn thực hiện, nếu đó là chương trình liên quan đến toán học, bạn bắt buộc phải hiểu biết nhiều về toán. Nếu bạn muốn làm một game 3D thì đòi hỏi bạn phải có kiến thức về hình học không gian.

Để học ngôn ngữ “C / C++”, bạn không cần thiết phải có những kiến thức cao cấp nào cả.

### 🔍 Nhưng đâu là cái khó khăn ?

Chúng ta cần phải biết máy tính hoạt động như thế nào để có thể hiểu chúng ta đang làm những gì. Và ở điểm này, hãy yên tâm, tôi sẽ cố gắng hết sức để hướng dẫn cho các bạn.

Một người lập trình cũng cần phải có một vài đặc điểm như sau:

- **Kiên trì**: một chương trình có thể sẽ không chạy tốt trong giai đoạn đầu, vì vậy bạn phải nhẫn nại!
- **Tư duy tốt**: chắc hẳn là bạn không cần phải giỏi lắm về toán, cái bạn cần là suy nghĩ một cách logic.
- **Nhệ nhàn**: Người ta không đánh máy bằng việc gõ búa lên bàn phím. 😬 Điều đó cũng không giúp chương trình của bạn chạy tốt hơn tí nào đâu. 😬

Tóm lại một cách đơn giản, việc học lập trình không đòi hỏi bạn phải thật sự có kiến thức chuyên sâu trong một lĩnh vực nào đó. Một người dốt toán vẫn có thể viết ra một chương trình, cái cần thiết chính là khả năng suy nghĩ của bạn.

## **Tổng Kết:**

- Để tạo ra được một chương trình máy tính, người ta phải viết những chương trình đó dựa trên một loại ngôn ngữ mà máy tính có thể biên dịch lại để hiểu. Người ta gọi đó là “Ngôn ngữ lập trình”.
- Có rất nhiều loại ngôn ngữ máy tính được phân theo nhiều cấp độ. Những ngôn ngữ cao cấp thì dễ sử dụng hơn nhưng chưa chắc sẽ mang lại hiệu quả cao hơn những ngôn ngữ cấp thấp.
- Ngôn ngữ lập trình C mà chúng ta đang được học trong tài liệu này được gọi là ngôn ngữ cấp thấp và nó cũng đang là ngôn ngữ lập trình phổ biến nhất thế giới hiện nay.
- Source code là tập hợp văn bản do bạn viết ra thể hiện ý nghĩa của ngôn ngữ lập trình.
- Compiler là một chương trình biên dịch có khả năng dịch source code sang ngôn ngữ nhị phân, sau đó chuyển thành chương trình Executable (.exe). Chúng ta phải biết rằng trong chương trình nhị phân thì không còn chứa source code.
- Việc lập trình không đòi hỏi bạn phải có một kiến thức chuyên sâu về một lĩnh vực nào đó như Toán học (ngoại trừ một số trường hợp chương trình bạn viết đòi hỏi phải sử dụng những công thức toán học chuyên sâu, vd như các phần mềm mã hóa). Tuy nhiên, một tư duy logic, nhạy bén là điều thật sự cần thiết đối với một lập trình viên.

Chà! Chúng ta kết thúc bài 1 rồi, nhưng các bạn vẫn chưa thấy bất kì dòng code nào giống như chúng ta đã thống nhất trước đó.

Trong chương tiếp theo, các bạn sẽ bắt đầu học cách lập trình với những công cụ đầu tiên. Bạn sẽ được hướng dẫn cài đặt những chương trình cần thiết cho bất kì người học lập trình nào. 😊

## TRẮC NGHIỆM KIẾN THỨC.

Phần này sẽ giúp bạn kiểm tra lại kiến thức đã học được, bạn chỉ việc chọn câu trả lời đúng nhất thôi.

❓ Những file nào được tạo ra từ việc lập trình ?

- A. Những file \*.exe chạy trên Windows
- B. Những hình ảnh (\*.jpg, \*.png, \*.bmp...)
- C. Những đoạn videos (\*.avi, \*.mov...)

❓ Giữa C và C++, ngôn ngữ nào cho phép ta lập trình tốt hơn ?

- A. C
- B. C++
- C. Cả hai đều mạnh mẽ như nhau

❓ Chương trình dịch ngôn ngữ cao cấp thành ngôn ngữ nhị phân gọi là :

- A. Programer
- B. Brumisateur
- C. Compiler

❓ Ngôn ngữ nào bạn sẽ học trong giai đoạn đầu ?

- A. C
- B. C++
- C. Cả hai

Đáp án:

- 1- A
- 2- C
- 3- C
- 4- C