Nếu bạn đang học công nghệ thông tin, xin chúc mừng bạn, bạn đang đi đúng hướng học ngành công nghệ thông tin là lập trình đó. Còn bạn đang học điện tử viễn thông, Cơ điện tử,Khoa điện thì xin chúc mừng bạn , bạn có ít cơ hội tiếp cận công viêc chính công viêc của mình. Các bạn phải theo chuyên ngành phụ lập trinh sẽ là nghề chính của bạn! Ủa tại sao vậy , điện tử viễn thông có khác công nghệ thông tin mà, có khác gì đâu! Nếu bạn còn thắc mắc như vậy thì tôi xin phép nêu ra định nghĩa này mà có chương trình học của bạn không nhé:

* Design patem
* OOP progaming
* SQL
* kiến thức về [Cấu trúc dữ liệu](https://nguyenvanhieu.vn/category/cau-truc-du-lieu-va-giai-thuat/)(array, stack, queue, linklist, map…), giải thuật cơ bản(tìm kiếm, sắp xếp…)
* Hệ thống phân tán
* Hệ thống Realtime
* Rời rac

Ô rất nhiêu vấn dề vân vân và mây mây. Tại sao admin đưa ra cơ sở như vậy nhỉ! Vì các bạn đã làm đúng chuyên ngành các bạn đẫ được học chưa! Nào chúng ta sẽ rảo bước định nghĩa chuyên ngành các bạn đang học nhé!

 Khoa điện : Chuyên về sản xuất điện năng, thiết kế, bảo trì, hướng dẫn người tiêu dùng điện,… Công việc cụ thể của kỹ sư điện là thiết kế máy phát điện, hệ thống điện ,đường dây tải điện,tổ chức thi công, bảo dưỡng hệ thống điện nhà máy, tòa nhà văn phòng,…

Khoa điện tử- viễn thông: Nghiên cứu các thiết bị điện tử, mạch tích hợp và bộ vi xử lý. thiết kế, chế tạo, thử nghiệm, bảo trì và giám sát việc sản xuất các thiết bị điện tử. thiết kế máy tính, thiết kế mạch truyền sóng, bảo trì thiết bị viễn thông,

Khoa điện tử- Điện tử mãy tính : lập trình cho phần cứng, thiết kế phần cứng, thiết kế hệ thống máy tình và truyền thông

Khoa công nghệ thông tin: Đây là khoa chuẩn để lập trình chuyên sâu, chuyên lập trình thường mai điện tử, đúng kiểu công nghệ thông tin mà các bạn thường thấy website hay ứng dụng máy tính hoặc di động

Tự nhiên sinh ra câu hỏi! Quái sao phân chia như vậy những anh làm FPT ,VNPT toàn kỹ sư điện tử, điện ,cơ điện mà làm lập trình ầm ầm như nào nhỉ! Khó hiểu sao ta! Đang tự hỏi như vậy trái ngành không nhỉ , Tại sao tụi ngành mình sao phải làm việc công nghệ thông tin. Xin thưa rằng, thị trường là vậy, thì trường như nào thì yêu cầu như thế !

Các bạn muốn tìm hiểu thị trường xin truy cập link này nhé

// bắt đầu link

CNTT thường đặt ra 2 sản phẩm

* Product : tự làm sản phẩm
* Outsourcing

Theo Wikipedia outsuoc

**Thuê ngoài** hay **gia công** là một thuật ngữ trong lĩnh vực kinh tế. Nó là việc một [thể nhân](https://vi.wikipedia.org/wiki/Th%E1%BB%83_nh%C3%A2n) hay [pháp nhân](https://vi.wikipedia.org/wiki/Ph%C3%A1p_nh%C3%A2n) chuyển giao việc thực hiện toàn bộ một chức năng sản xuất-kinh doanh nào đó, bao gồm cả tài sản vật chất và nhân lực cho một nhà cung cấp dịch vụ bên ngoài [chuyên môn hóa](https://vi.wikipedia.org/w/index.php?title=Chuy%C3%AAn_m%C3%B4n_h%C3%B3a&action=edit&redlink=1) trong lĩnh vực đó, gọi là [nhà thầu phụ](https://vi.wikipedia.org/w/index.php?title=Nh%C3%A0_th%E1%BA%A7u_ph%E1%BB%A5&action=edit&redlink=1). Dịch vụ có thể được cung cấp bên trong hay bên ngoài công ty khách hàng; có thể thuộc nước sở tại hoặc ở nước ngoài. Các chuyển giao như vậy nhằm mục đích hạ [giá thành](https://vi.wikipedia.org/w/index.php?title=Gi%C3%A1_th%C3%A0nh&action=edit&redlink=1) và nâng cao tính cạnh tranh. Nó khác với việc mua bán sản phẩm từ nhà cung cấp ở chỗ giữa hai bên có sự trao đổi thông tin để quản lý việc sản xuất-kinh doanh đó, tức là có sự hợp tác trong sản xuất.

Hiện nay, hoạt động thuê ngoài được áp dụng rất nhiều trong ngành công nghệ thông tin. Các công ty hàng đầu trong ngành công nghệ thông tin có thể thuê người bên ngoài thực hiện mảng công nghệ thông tin của công ty mình. Một đối tác như vậy sẽ cung cấp nhân lực làm việc sát cánh với nhân viên của công ty, có khả năng nắm bắt các vấn đề của riêng công ty và từ đó giúp công ty chuyển giao và thực hiện các giải pháp thích hợp.

Thuê ngoài gần giống với [chuyển ra ngoài](https://vi.wikipedia.org/wiki/Offshoring) khi thể nhân hay pháp nhân thuê mướn nhà thầu phụ tại nước ngoài, nhưng khác khi họ chỉ thuê mướn nhà thầu phụ trong cùng quốc gia hoặc khác ở chỗ thuê ngoài luôn là thuê mướn nhà thầu phụ không thuộc về pháp nhân đó, trong khi chuyển ra ngoài có thể chỉ là việc chuyển giao công việc cho chi nhánh của chính pháp nhân đó tại nước ngoài.

Thông tin chi tiết bạn có thể đọc tại <https://vi.wikipedia.org/wiki/Thu%C3%AA_ngo%C3%A0i>

Product

là các sản phẩm có mối liên hệ nhu cầu cung cầu mật thiết với nhau nhau . Đơn giản sản phẩm là quá trình phát triển từ quá trình tìm hiểu nhu cầu phát triển=> nghiên cứu sản phẩm => phát triển => đưa tới khách hàng .

* Vì vậy Outsourcing là 1 phần của dự án product .

CNTT có hay làm product hay outsouce ờ thì có cả 2 . theo thông tin

<https://vnexpress.net/kinh-doanh/chien-luoc-tim-va-giu-cua-cong-ty-phan-mem-viet-nam-3828854.html>

Các số liệu mới đây đều chỉ ra rằng Việt Nam là một trong những 'miền đất hứa' của ngành gia công phần mềm (IT outsourcing) với mức độ tăng trưởng khả quan và đều đặn.

Theo đó, báo cáo tổng kết năm 2017 của Bộ Thông tin và Truyền thông cho biết xuất khẩu phần mềm cả nước đạt khoảng 58.500 tỷ đồng, tăng 4,4% so với cùng kỳ năm 2016. Công bố năm 2017 của hãng tư vấn AT Kearney cũng chỉ ra rằng Việt Nam xếp thứ 6 trong top 55 quốc gia hấp dẫn nhất thế giới về gia công phần mềm.

Ở việt nam là nước công nghệ thấp nên không có nhiều cơ hội cho ngành khoa điện . Nếu bạn đã biết thì ở nước ngoài ngành cơ điện tử có mức lương cao nhất, nhưng ở việt nam sẽ là thấp.

<http://tapchicongthuong.vn/bai-viet/10-nganh-hoc-co-muc-luong-cao-nhat-o-my-62267.htm>

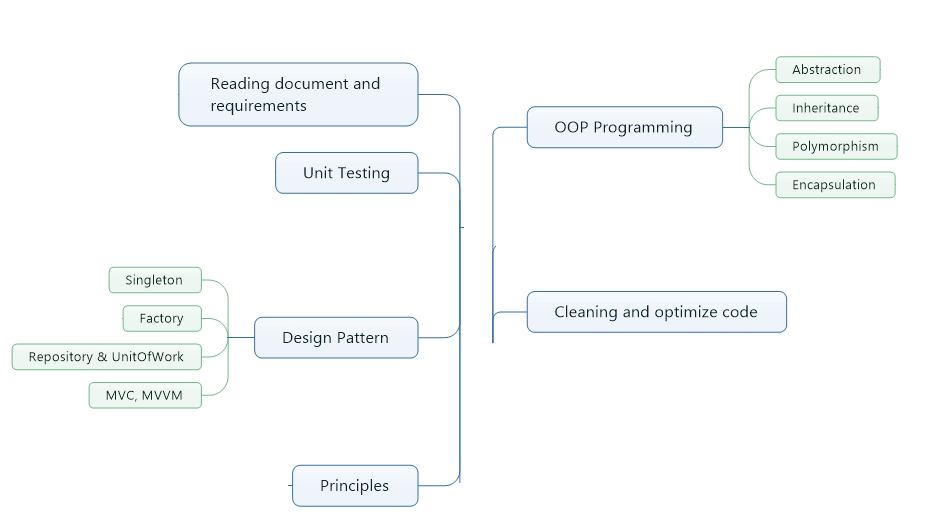
Nếu đọc ở đây chắc bạn phải thấy vọng không nào! Bạn yên tâm đời không tuyeeeeth đường sống nếu bạn cố gang nỗ lực cố gắng

// kết thúc link

Nếu bạn đang nghĩ mình có lợi thế cạnh tranh công nghệ thông tin được không nhỉ! Có thể bạn nghĩ là không

Vậy kỹ năng lập trình các bạn nên tập trung những gì

Lập trình c/c++



Nào chúng ta sẽ bát đầu phát triển lập trình c/ c++ từ fresher , junio, Sennio:

Các bạn đi từ từ nhe đừng có nóng vội!

Fresher là mới bắt đầu => kinh nghiệm=> chuyên gia < không đơn vị nào định nghĩa trình độ này nhưng trình độ này có tiêu chí riêng nên chúng ta có thể định hình được>

Mọi thứ không nhanh được đâu , mọi người từ từ nhé. Ở đây tớ khuyến khích học đi đôi với hành

Nào chúng ta bắt đầu fresher nhé:

Chúng ta sẽ học bài lâp trình cơ bản

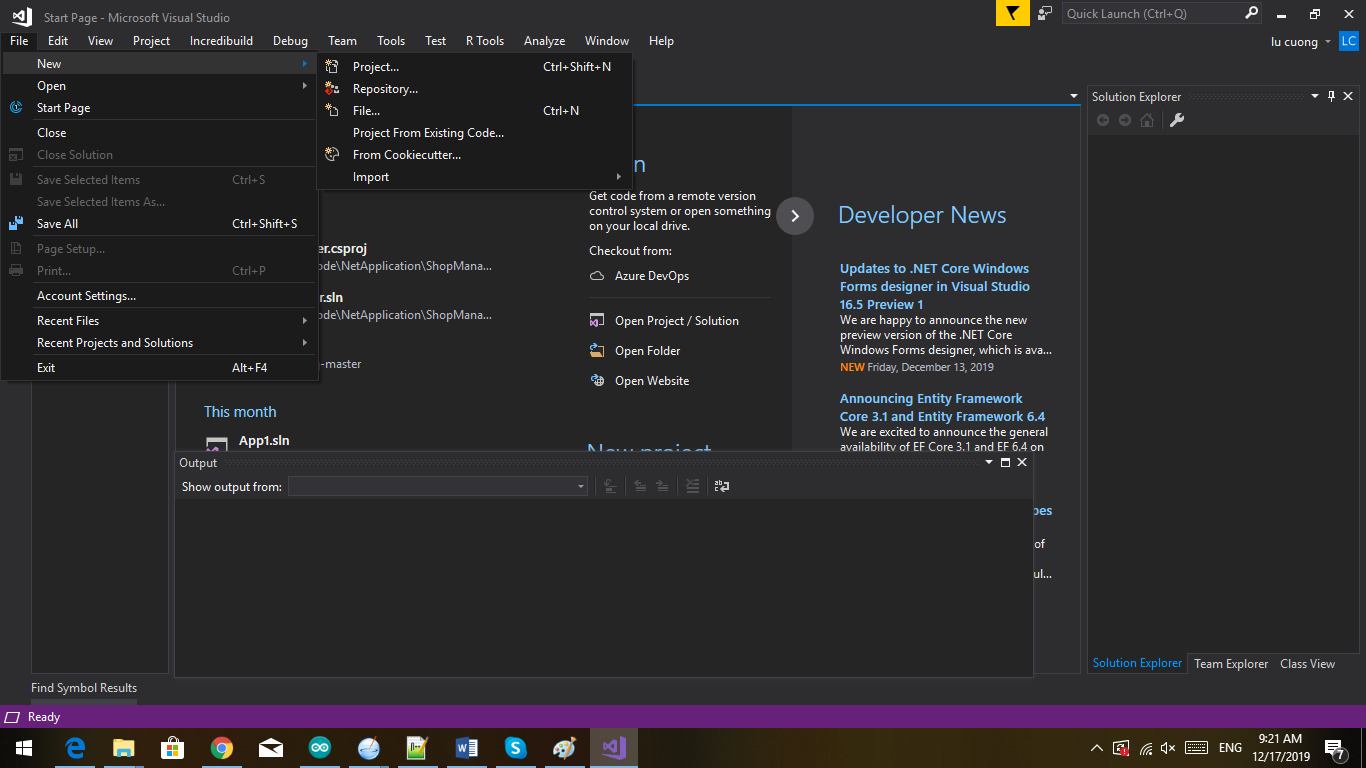
ở đây chúng ta chỉ lập trình c đơn thuần thôi, không cần đao to búa lớn như kiểu nhúng hay linux vì các bạn nên định nghĩa lập trình là thứ gì đã

Hướng dẫn tải và cài đặt visual bạn tham khảo google nhé, Vì các phiên bản visual đã có những kiểu cài đặt khác nhau, cách thức cài đặt khác nhau!

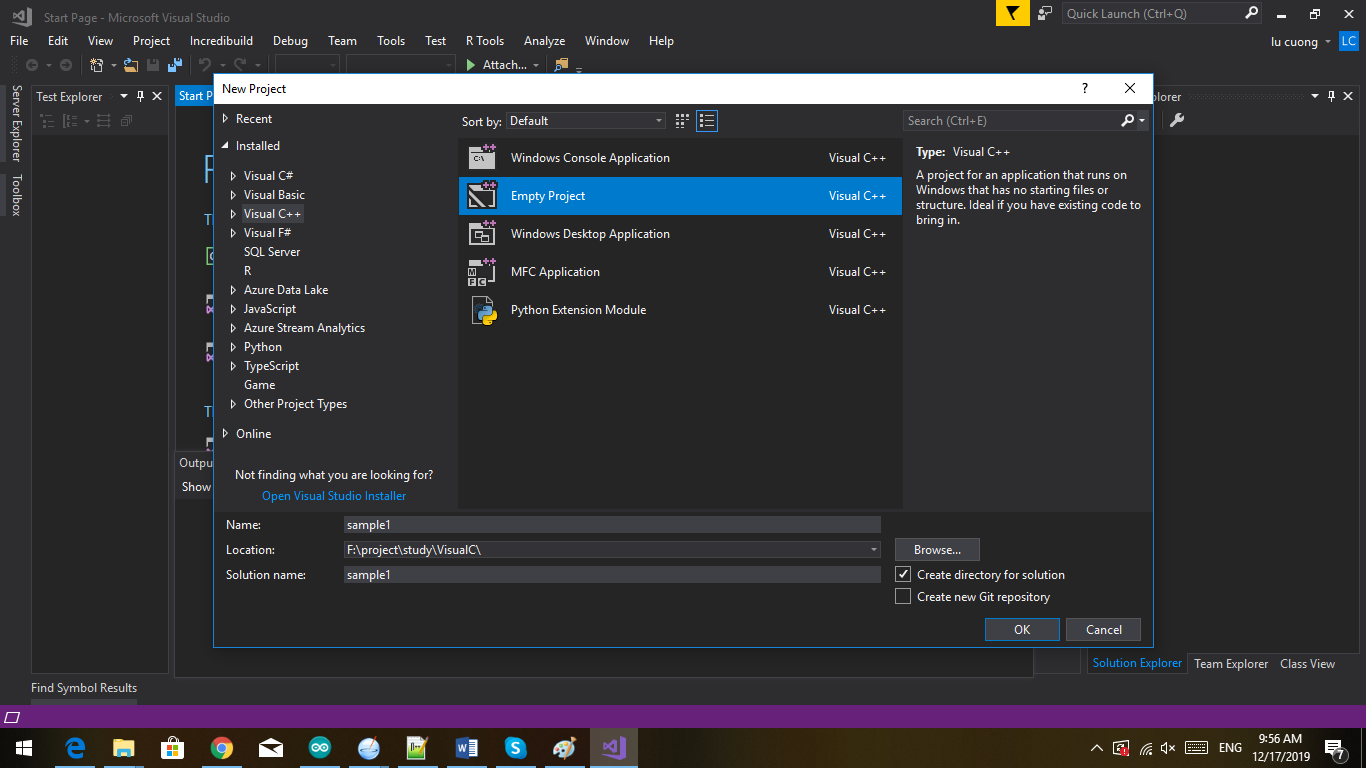
Bản hiện tại mình visual community nhé. Phiên bản này miễn phí vì vậy các bạn thoải mái download và sử dụng

Việc đầu tiên bạn bật visual ra

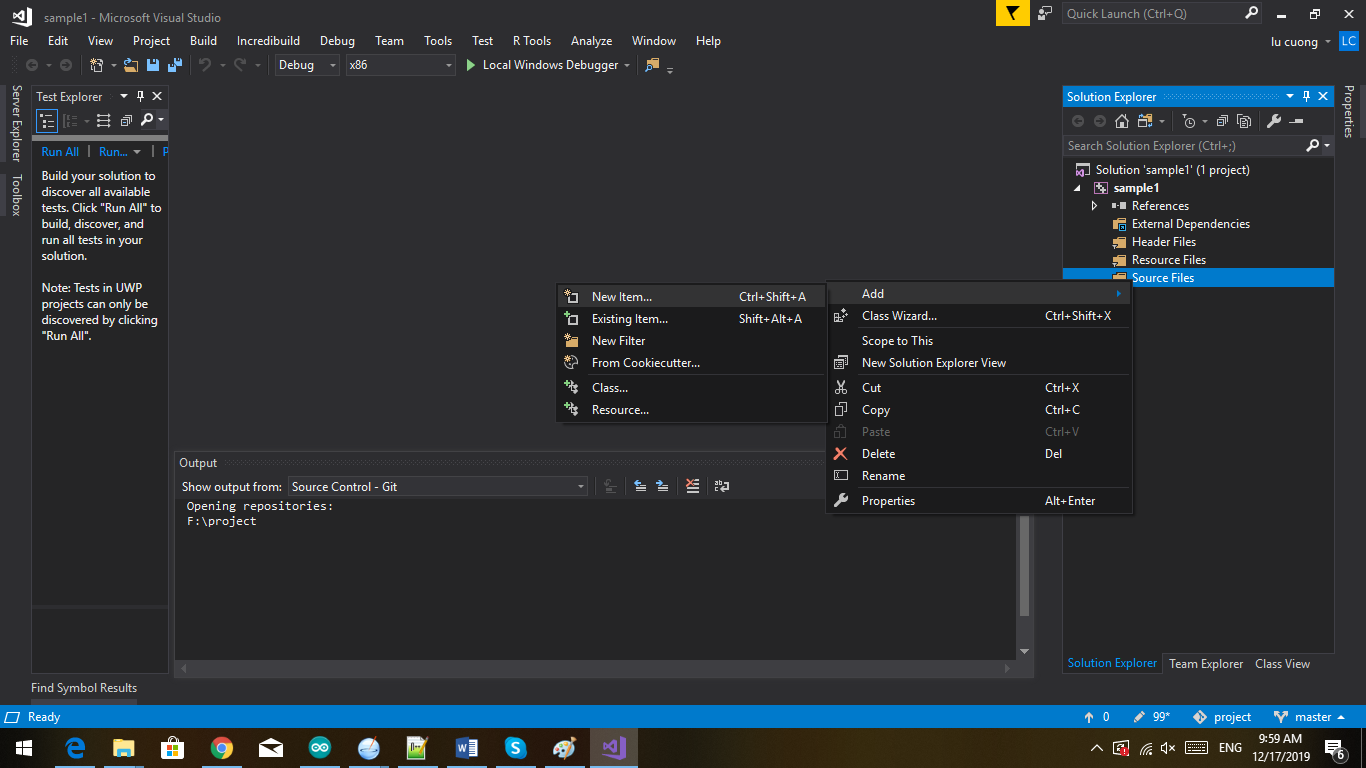
Bạn chọn new project



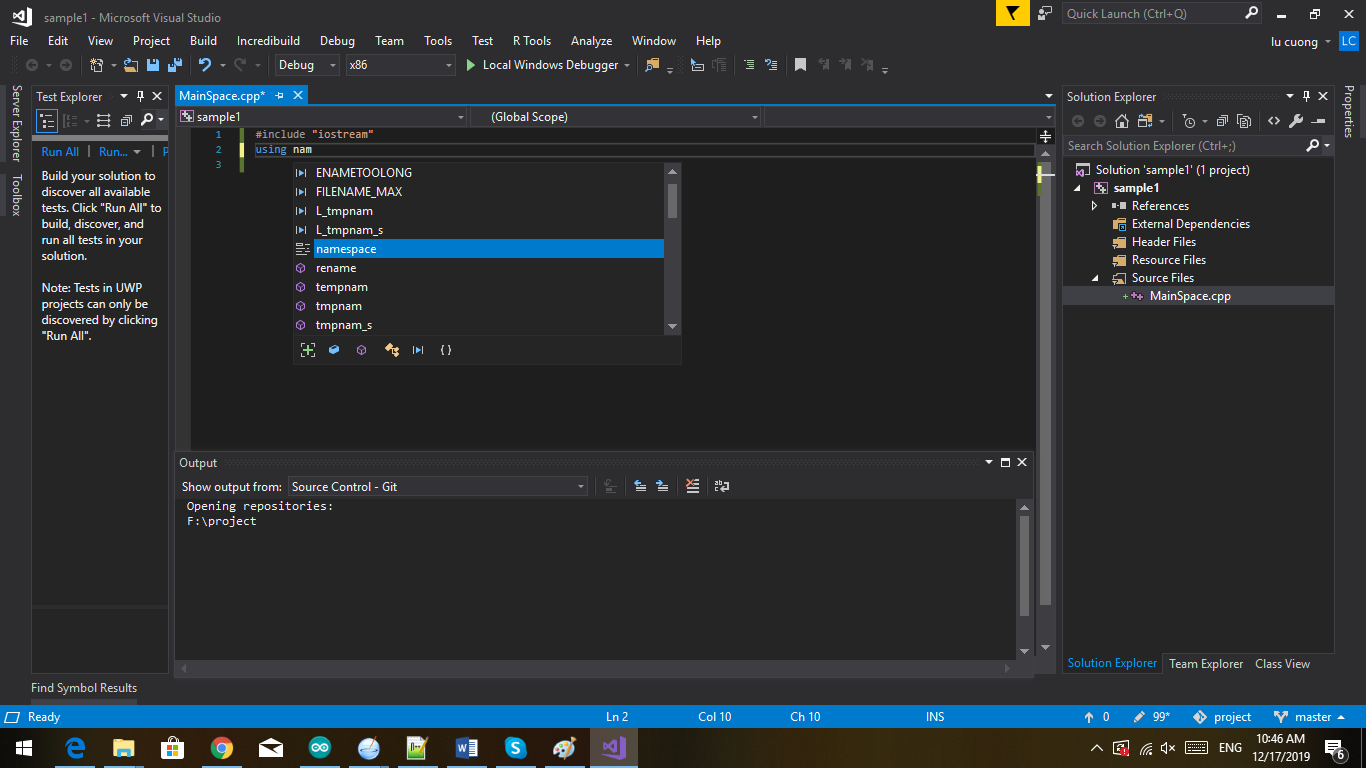
Các bạn chọn C++



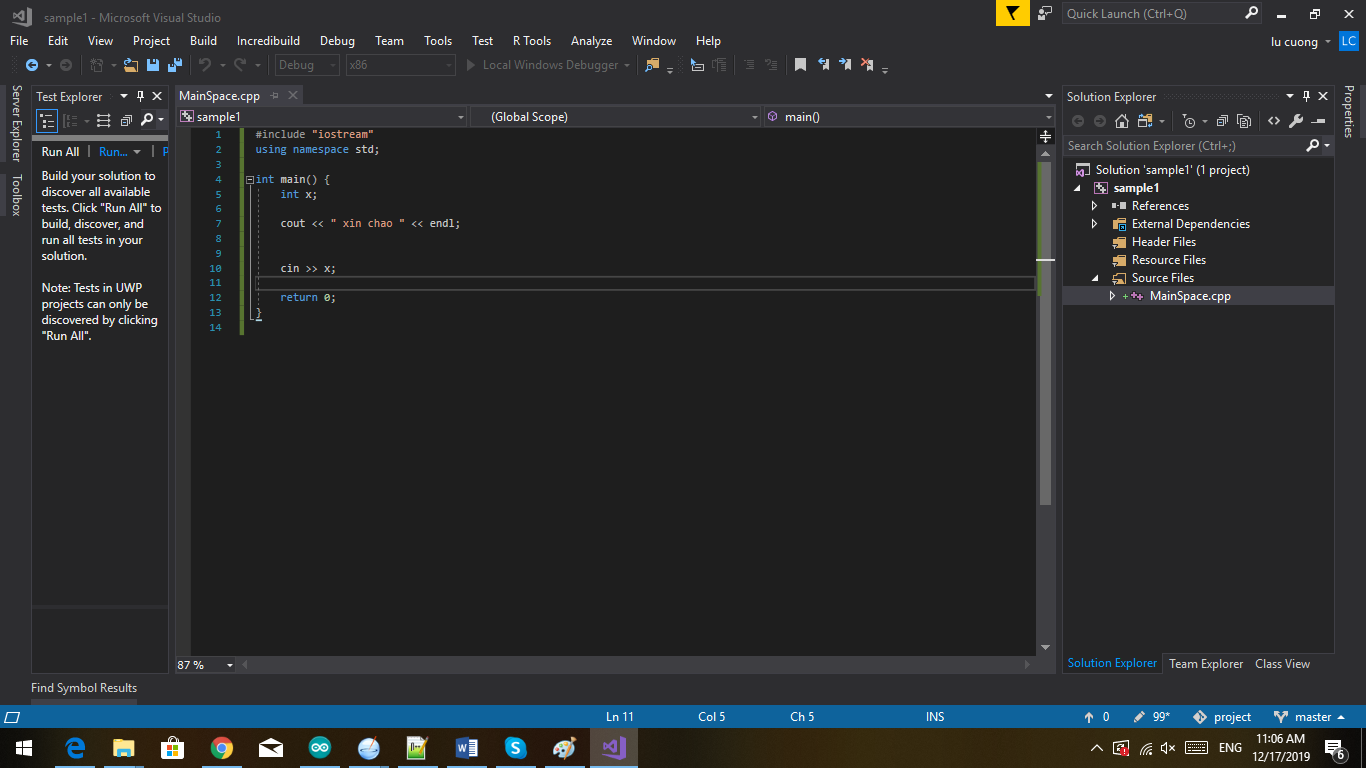
Bạn chọn mục



Nào chúng ta code thôi , visual hỗ trợ gợi ý chúng ta rồi đó, mỗi lần gõ visual sẽ gợi ý từ khóa cho chúng ta biết, thật tiện lợi không nào!



Chúng ta bắt đầu thử đoạn code để khời động tư duy nhé



Các bạn them đoạn code như sau nhé

#include "iostream"

using namespace std;

int main() {

int x;

cout << " xin chao " << endl;

cin >> x;

return 0;

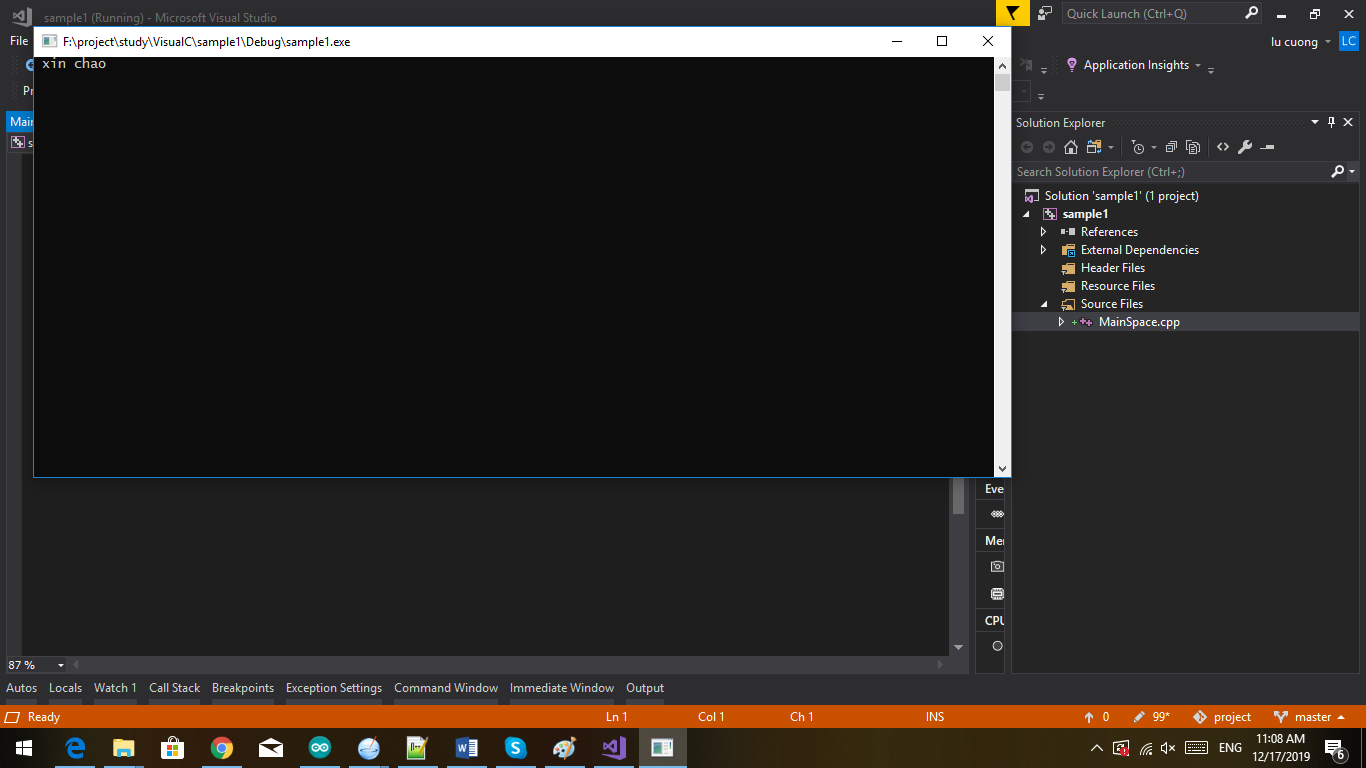
}

Các bạn chú ý phải khai báo từ khóa

using namespace std;

Thì các lệnh trên có ý nghĩa

Nhấn nút debug chúng ta được kết quả



Chú ý nhé các bạn phải có lệnh

cin >> x;

thì màn hình đen sẽ giữ lại nếu không sẽ tắt hết đó

nào chúng ta bắt đầu những cú pháp cơ bản nảo

Nếu bạn có thời gian thì đọc kỹ nếu không thì nên đọc lướt thôi nhé, tôi sẽ hướng dẫn chi tiết sau nhé

# Compilers

The essential tools needed to follow these tutorials are a computer and a compiler toolchain able to compile C++ code and build the programs to run on it.  
  
C++ is a language that has evolved much over the years, and these tutorials explain many features added recently to the language. Therefore, in order to properly follow the tutorials, a recent compiler is needed. It shall support (even if only partially) the features introduced by the 2011 standard.  
  
Many compiler vendors support the new features at different degrees. See the bottom of this page for some compilers that are known to support the features needed. Some of them are free!  
  
If for some reason, you need to use some older compiler, you can access an older version of these tutorials [here](http://www.cplusplus.com/doc/oldtutorial/) (no longer updated).

Các công cụ thiết yếu cần thiết để làm theo các hướng dẫn này là một máy tính và một công cụ biên dịch có thể biên dịch mã C ++ và xây dựng các chương trình để chạy trên nó.

C ++ là một ngôn ngữ đã phát triển nhiều năm qua và những hướng dẫn này giải thích nhiều tính năng được thêm vào gần đây cho ngôn ngữ này. Do đó, để làm theo đúng hướng dẫn, cần có một trình biên dịch gần đây. Nó sẽ hỗ trợ (ngay cả khi chỉ một phần) các tính năng được giới thiệu bởi tiêu chuẩn 2011.

Nhiều nhà cung cấp trình biên dịch hỗ trợ các tính năng mới ở các mức độ khác nhau. Xem dưới cùng của trang này để biết một số trình biên dịch được biết là hỗ trợ các tính năng cần thiết. Một số trong số họ là miễn phí!

Nếu vì lý do nào đó, bạn cần sử dụng một số trình biên dịch cũ hơn, bạn có thể truy cập phiên bản cũ hơn của các hướng dẫn này tại đây (không còn cập nhật nữa).

Console programs are programs that use text to communicate with the user and the environment, such as printing text to the screen or reading input from a keyboard.  
  
Console programs are easy to interact with, and generally have a predictable behavior that is identical across all platforms. They are also simple to implement and thus are very useful to learn the basics of a programming language: The examples in these tutorials are all console programs.  
  
The way to compile console programs depends on the particular tool you are using.  
  
The easiest way for beginners to compile C++ programs is by using an Integrated Development Environment (IDE). An IDE generally integrates several development tools, including a text editor and tools to compile programs directly from it.  
  
Here you have instructions on how to compile and run console programs using different free Integrated Development Interfaces (IDEs):

Các chương trình bảng điều khiển là các chương trình sử dụng văn bản để giao tiếp với người dùng và môi trường, chẳng hạn như in văn bản ra màn hình hoặc đọc đầu vào từ bàn phím.

Các chương trình điều khiển rất dễ tương tác và thường có hành vi có thể dự đoán giống hệt nhau trên tất cả các nền tảng. Chúng cũng đơn giản để thực hiện và do đó rất hữu ích để tìm hiểu những điều cơ bản của ngôn ngữ lập trình: Các ví dụ trong các hướng dẫn này là tất cả các chương trình điều khiển.

Cách để biên dịch các chương trình giao diện điều khiển phụ thuộc vào công cụ cụ thể bạn đang sử dụng.

Cách dễ nhất cho người mới bắt đầu biên dịch chương trình C ++ là sử dụng Môi trường phát triển tích hợp (IDE). Một IDE thường tích hợp một số công cụ phát triển, bao gồm trình soạn thảo văn bản và các công cụ để biên dịch các chương trình trực tiếp từ nó.

Tại đây, bạn có hướng dẫn về cách biên dịch và chạy các chương trình bảng điều khiển bằng cách sử dụng các Giao diện phát triển tích hợp (IDE) miễn phí khác nhau:

|  |  |  |
| --- | --- | --- |
| **IDE** | **Platform** | **Console programs** |
| **Code::blocks** | Windows/Linux/MacOS | [Compile console programs using Code::blocks](http://www.cplusplus.com/doc/tutorial/introduction/codeblocks/) |
| **Visual Studio Express** | Windows | [Compile console programs using VS Express 2013](http://www.cplusplus.com/doc/tutorial/introduction/visualstudio/) |
| **Dev-C++** | Windows | [Compile console programs using Dev-C++](http://www.cplusplus.com/doc/tutorial/introduction/devcpp/) |

If you happen to have a Linux or Mac environment with development features, you should be able to compile any of the examples directly from a terminal just by including C++11 flags in the command for the compiler:  
Nếu bạn tình cờ có môi trường Linux hoặc Mac với các tính năng phát triển, bạn sẽ có thể biên dịch bất kỳ ví dụ nào trực tiếp từ thiết bị đầu cuối chỉ bằng cách đưa cờ C ++ 11 vào lệnh cho trình biên dịch:

|  |  |  |
| --- | --- | --- |
| **Compiler** | **Platform** | **Command** |
| **GCC** | Linux, among others... | g++ -std=c++0x example.cpp -o example\_program |
| **Clang** | OS X, among others... | clang++ -std=c++11 -stdlib=libc++ example.cpp -o example\_program |

Cấu trúc chương trình

Cách tốt nhất để học một ngôn ngữ lập trình là viết chương trình. Thông thường, người mới bắt đầu viết chương trình là một chương trình có tên "Hello World", đơn giản chỉ in "Hello World" lên màn hình máy tính của bạn. Mặc dù nó rất đơn giản, nhưng nó chứa tất cả các thành phần cơ bản mà các chương trình C ++ có:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 | // my first program in C++  #include <iostream>  int main()  {  std::cout << "Hello World!";  } | Hello World! | [Edit & Run](http://www.cplusplus.com/doc/tutorial/program_structure/) |

Bảng bên trái ở trên hiển thị mã C ++ cho chương trình này. Bảng bên phải hiển thị kết quả khi chương trình được thực hiện bởi máy tính. Các số màu xám ở bên trái của bảng là số dòng để thảo luận về chương trình và nghiên cứu lỗi dễ dàng hơn. Họ không phải là một phần của chương trình.

Hãy xem xét từng dòng chương trình này:

Dòng 1: // chương trình đầu tiên của tôi trong C ++

Hai dấu gạch chéo chỉ ra rằng phần còn lại của dòng là một bình luận được chèn bởi lập trình viên nhưng không có ảnh hưởng đến hành vi của chương trình. Các lập trình viên sử dụng chúng để bao gồm các giải thích hoặc quan sát ngắn liên quan đến mã hoặc chương trình. Trong trường hợp này, nó là một mô tả giới thiệu ngắn gọn về chương trình.

Dòng 2: #include <iostream>

Các dòng bắt đầu bằng dấu băm (#) là các lệnh được đọc và diễn giải bởi cái được gọi là bộ tiền xử lý. Chúng là những dòng đặc biệt được giải thích trước khi quá trình biên dịch chương trình bắt đầu. Trong trường hợp này, lệnh #include <iostream>, hướng dẫn bộ tiền xử lý bao gồm một phần của mã C ++ tiêu chuẩn, được gọi là tiêu đề iostream, cho phép thực hiện các hoạt động đầu vào và đầu ra tiêu chuẩn, như viết đầu ra của chương trình này (Hello World ) đến màn hình.

Dòng 3: Một dòng trống.

Dòng trống không có hiệu lực trên một chương trình. Họ chỉ đơn giản là cải thiện khả năng đọc của mã.

Dòng 4: int main ()

Dòng này khởi tạo khai báo của một hàm. Về cơ bản, một hàm là một nhóm các câu lệnh mã được đặt tên: trong trường hợp này, điều này đặt tên "chính" cho nhóm các câu lệnh mã theo sau. Các hàm sẽ được thảo luận chi tiết trong chương sau, nhưng về cơ bản, định nghĩa của chúng được giới thiệu với sự nối tiếp của một loại (int), tên (chính) và một cặp dấu ngoặc đơn (()), bao gồm tùy chọn bao gồm các tham số.

Hàm có tên main là một hàm đặc biệt trong tất cả các chương trình C ++; nó là hàm được gọi khi chương trình được chạy. Việc thực thi tất cả các chương trình C ++ bắt đầu bằng hàm chính, bất kể hàm đó thực sự nằm ở đâu trong mã.

Dòng 5 và 7: {và}

Dấu ngoặc mở ({) ở dòng 5 biểu thị điểm bắt đầu của định nghĩa hàm chính và dấu ngoặc nhọn (}) ở dòng 7, biểu thị kết thúc của nó. Tất cả mọi thứ giữa các dấu ngoặc này là phần thân của hàm xác định điều gì sẽ xảy ra khi hàm chính được gọi. Tất cả các hàm sử dụng dấu ngoặc nhọn để chỉ sự bắt đầu và kết thúc định nghĩa của chúng.

Dòng 6: std :: cout << "Xin chào thế giới!";

Dòng này là một tuyên bố C ++. Một tuyên bố là một biểu thức thực sự có thể tạo ra một số hiệu ứng. Nó là thịt của một chương trình, chỉ định hành vi thực tế của nó. Các câu lệnh được thi hành theo cùng thứ tự xuất hiện trong thân hàm.

Tuyên bố này có ba phần: Đầu tiên, std :: cout, xác định thiết bị đầu ra ký tự tiêu chuẩn (thông thường, đây là màn hình máy tính). Thứ hai, toán tử chèn (<<), chỉ ra rằng những gì tiếp theo được chèn vào std :: cout. Cuối cùng, một câu trong dấu ngoặc kép ("Xin chào thế giới!"), Là nội dung được chèn vào đầu ra tiêu chuẩn.

Lưu ý rằng câu lệnh kết thúc bằng dấu chấm phẩy (;). Nhân vật này đánh dấu sự kết thúc của tuyên bố, giống như giai đoạn kết thúc một câu bằng tiếng Anh. Tất cả các câu lệnh C ++ phải kết thúc bằng ký tự dấu chấm phẩy. Một trong những lỗi cú pháp phổ biến nhất trong C ++ là quên kết thúc câu lệnh bằng dấu chấm phẩy.

Bạn có thể nhận thấy rằng không phải tất cả các dòng của chương trình này thực hiện các hành động khi mã được thực thi. Có một dòng chứa một nhận xét (bắt đầu bằng //). Có một dòng với một lệnh cho bộ tiền xử lý (bắt đầu bằng #). Có một dòng xác định một chức năng (trong trường hợp này là chức năng chính). Và cuối cùng, một dòng có câu lệnh kết thúc bằng dấu chấm phẩy (chèn vào cout), nằm trong khối được giới hạn bởi dấu ngoặc ({}) của hàm chính.

Chương trình đã được cấu trúc theo các dòng khác nhau và được thụt lề đúng cách, để dễ hiểu hơn cho người đọc nó. Nhưng C ++ không có các quy tắc nghiêm ngặt về thụt đầu dòng hoặc về cách phân chia các hướng dẫn trong các dòng khác nhau. Ví dụ, thay vì

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 | int main ()  {  std::cout << " Hello World!";  } | [Edit & Run](http://www.cplusplus.com/doc/tutorial/program_structure/) |

We could have written:

|  |  |  |
| --- | --- | --- |
|  | int main () { std::cout << "Hello World!"; } |  |

<http://www.cplusplus.com/doc/tutorial/program_structure/>