

# Bài 1. C++ Các khái niệm cơ bản

**Mục tiêu:** Làm quen với C++ và môi trường lập trình. Sinh viên biết viết một chương trình C++ nhập xuất đơn giản với input/output chuẩn, biết cách dịch và chạy chương trình.

Ôn lại các khái niệm cơ bản như hằng, biến, kiểu dữ liệu, lệnh điều kiện, lệnh lặp...

*Sinh viên đã học C++ ở môn Tin học cơ sở 4 có thể bỏ qua các mục 1-2-3 và các bài tập dễ.*

## 1. Chuẩn bị môi trường lập trình

Ta sẽ sử dụng bộ biên dịch GNU GCC. Nếu bạn dùng hệ điều hành Linux/Unix thì máy bạn đã sẵn có bộ biên dịch này. Nếu bạn dùng Windows, bạn có thể cài Code::Blocks (bản kèm sẵn MinGW bên trong), hoặc còn có nhiều lựa chọn khác như Visual Code,.... tùy bạn quyết định. Bài thực hành này hướng dẫn làm quen với C++ qua Code::Blocks.

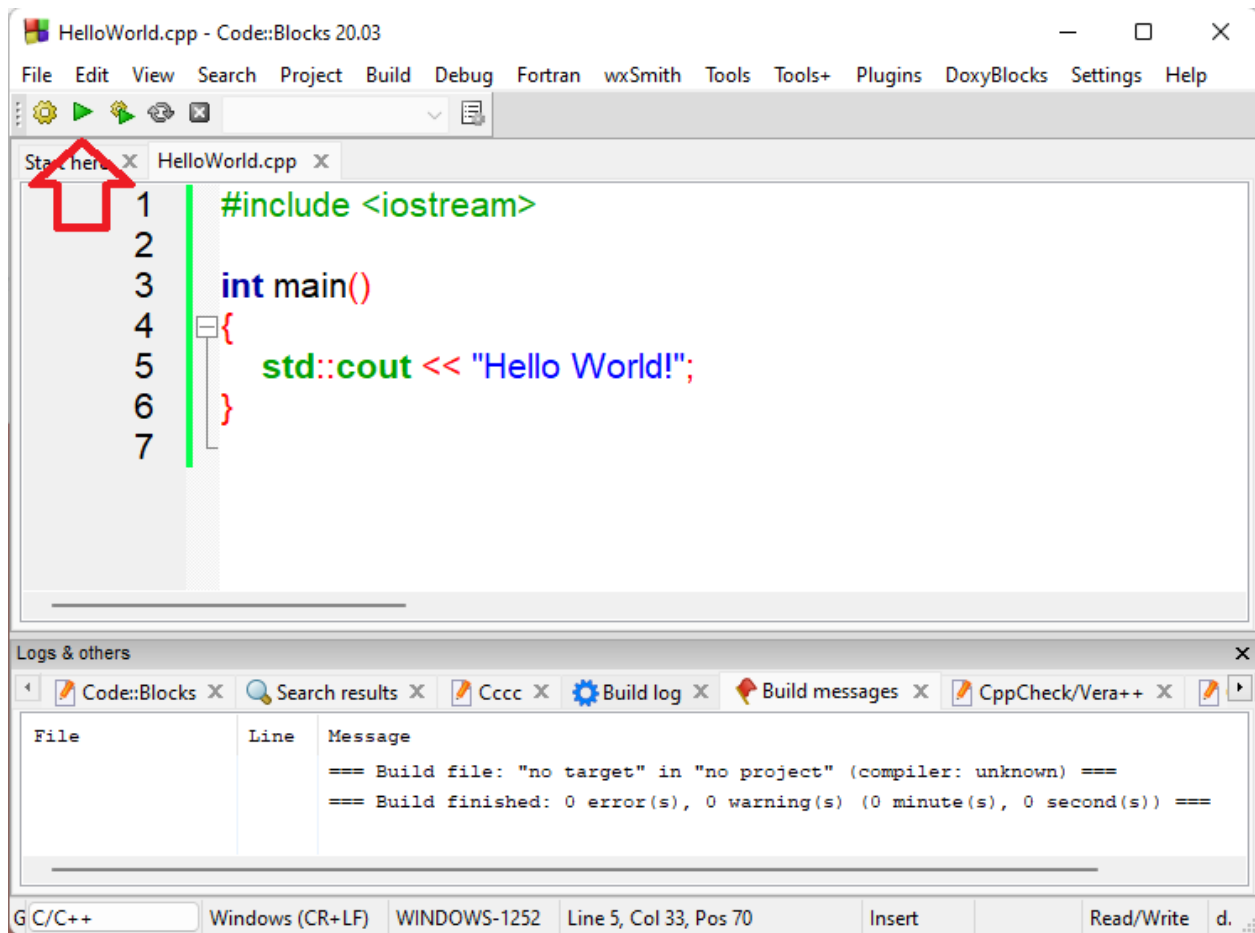
## 2. Chương trình đầu tiên

Chạy Code::Blocks.

Mở một file mới bằng **phím tắt Ctrl-Shift-N** hoặc menu **File|New Empty File**

Lưu file, chẳng hạn với tên HelloWorld.cpp. Ta sử dụng quy ước đặt tên cho các file mã nguồn C++ có phần mở rộng là cpp.

Gõ nội dung như dưới đây vào file. Chú ý gõ chính xác để thấy các màu cú pháp (syntax highlight) được hiển thị như trong hình



Để ý hình trên có một mũi tên màu đỏ chỉ tới ba nút trên thanh công cụ. Đó là ba chức năng bạn sẽ thường xuyên sử dụng. Kể từ trái sang phải, ba chức năng đó là:

1. Biên dịch (compile): Dịch từ file mã nguồn mà bạn đang soạn sang file chạy được. Nếu không thành công do có lỗi biên dịch thì bạn sẽ thấy thông báo lỗi ở tab **Build messages** ở đáy cửa sổ Code::Blocks. Nếu biên dịch thành công, bạn sẽ thấy đường dẫn đến file chạy được cũng tại tab đó. Phím tắt cho chức năng này là **F9**
2. Chạy chương trình (run). Phím tắt là **F10**
3. Dịch, sau đó chạy nếu dịch thành công. Phím tắt là **F11**.

Hãy thử ba chức năng trên cho chương trình HelloWorld.cpp bạn vừa viết. Nên tập dùng các phím tắt cho thạo thay vì dùng chuột để tăng tốc độ thao tác.

Hãy sửa cho chương trình HelloWorld sai cú pháp, chẳng hạn xóa dấu ngoặc } cuối cùng, sau đó dịch lại để xem thông báo lỗi tại tab **Build messages**. Mỗi dòng thông báo lỗi cho biết lỗi được phát hiện tại dòng nào và lỗi đó là gì.

Thử xóa dòng `#include <iostream>`, bấm F9 để dịch, bạn sẽ thấy thông báo lỗi ... [Error] 'cout' is not a member of 'std'. Đó là vì khi bỏ include iostream là thư viện input/output chứa cout, trình biên dịch sẽ không hiểu cout là gì. std là không gian tên của thư viện chuẩn C++, khi ta viết `std::cout` có nghĩa ta đang nói tới một thứ có tên là cout nằm trong không gian tên std thư viện chuẩn.

Sau khi lấy lại dòng include, hãy thử xóa std::, bấm F9 để dịch, sau đó đọc các dòng báo lỗi xem nó nói gì.

### 3. using namespace std

Hãy sửa chương trình thành nội dung như sau.

```
#include <iostream>

int main()
{
    std::string name;
    std::cout << "What's your name?";
    std::cin >> name;
    std::cout << "Hello " << name << "!";
}
```

Trong hàm main có 04 lệnh. Lệnh thứ nhất khai báo một biến kiểu *string* có tên là *name*. Lệnh thứ hai in ra màn hình mời nhập tên. Lệnh thứ ba đọc từ input chuẩn một chuỗi kí tự và ghi vào biến name. Lệnh thứ tư in ra màn hình lời chào với tên vừa nhập.

Hãy chạy thử.

Bạn có thấy lần nào cũng phải có std:: cho từng khái niệm trong thư viện chuẩn? Cách thông dụng để không phải lặp đi lặp lại std:: trong code là dùng khai báo using namespace std ở ngay sau các định hướng #include. Như sau:

```
#include <iostream>

using namespace std;

int main()
{
    string name;
    cout << "What's your name?";
    cin >> name;
    cout << "Hello " << name << "!";
}
```

Từ nay, ta sẽ dùng cách này cho chương trình gọn và dễ đọc.

## Bài tập

Sinh viên cần làm hết toàn bộ các bài dạng A, làm hoàn chỉnh tối thiểu 05 dạng B và 05 bài dạng C. Biết cách làm tất cả các bài thuộc các dạng B và C.

## A. Dạng bài tìm hiểu

1. Lệnh sau đây làm gì khi grade là một biến kiểu int? Nếu có lỗi cú pháp, hãy sửa lỗi.

```
bool isA = (90 <= grade <= 100);
```

2. Đoạn chương trình sau đây in ra kết quả gì?

```
double x = (double) (3/5);  
cout << x;
```

3. Tại sao đoạn code dưới đây không in ra  $4294967296 = 2^{32}$ ?

```
int x = 65536;  
long y = x * x;  
cout << y;
```

Gợi ý (bôi đen để nhìn thấy):

4. Biểu thức sau có giá trị gì?

```
(sqrt(2) * sqrt(2) == 2)
```

5. Viết một chương trình divideByZero để xem chuyện gì xảy ra khi bạn đem chia một giá trị kiểu int hoặc double cho 0 hoặc lấy đồng dư cho 0.

6. Thử xem trình biên dịch nói gì với một chương trình có hàm main chứa đoạn code sau

```
int a = 27 * "three";
```

hoặc

```
double x;  
cout << x;
```

Nếu chuyển dòng khai báo x ra ngoài hàm main thì chuyện gì xảy ra?

7. Đoạn code sau cho output gì?

```
int threeInt = 3;  
int fourInt = 4;  
double threeDouble = 3.0;  
double fourDouble = 4.0;  
cout << threeInt / fourInt << endl;  
cout << threeInt / fourDouble << endl;  
cout << threeDouble / fourInt << endl;  
cout << threeDouble / fourDouble << endl;
```

8. Chuyện gì xảy ra nếu ta khai báo hai biến trùng tên trong cùng một khối lệnh (block)?

Lấy ví dụ là chương trình có hàm main() với nội dung sau:

```

int arg1;
arg1 = -1;
int x, y, z;
char myDouble = '5';
char arg1 = 'A';
cout << arg1 << "\n";
return 0;

```

Gợi ý: Hãy thử dịch xem trình biên dịch báo lỗi gì.

9. Cùng câu hỏi trên với đoạn code sau:

```

int arg1;
arg1 = -1;
{
    char arg1 = 'A';
    cout << arg1 << "\n";
}
return 0;

```

Khi ở bên trong khối lệnh con, định danh arg1 chỉ tới biến arg1 nào?

10. Cùng câu hỏi trên với đoạn code sau:

```

int arg1;
arg1 = -1;
{
    char arg1 = 'A';
}
cout << arg1 << "\n";
return 0;

```

Khi ra khỏi khối lệnh con, định danh arg1 chỉ tới biến arg1 nào?

11. Chuyển đổi nhiệt độ. Dòng lệnh sau có nhiệm vụ đổi từ độ F (Fahrenheit) sang độ C (Celsius). Lệnh đó có gì sai?

```
double C = (F - 32) * (5 / 9);
```

12. Đoạn code sau in ra kết quả gì?

```

if (10 > 5);
else; {
    cout << "Here";
};

```

## B. Luyện tập biểu thức và các lệnh điều kiện

13. Viết một chương trình `distance.cpp` đọc hai số nguyên  $x$  và  $y$  từ bàn phím và in ra khoảng cách từ điểm  $(x, y)$  tới gốc tọa độ  $(0, 0)$ .
14. Đoán số lớn nhất. Xét trò chơi sau: Alice viết hai số nguyên trong khoảng từ 0 đến 100 vào 2 thẻ. Bob được chọn xem giá trị ghi trên một trong hai thẻ. Sau khi xem thẻ, Bob được chọn một trong hai thẻ. Nếu đó là thẻ có giá trị lớn hơn trong hai thẻ thì Bob thắng, nếu không thì Bob thua. Hãy tìm một chiến lược chơi cho Bob và viết chương trình tương ứng để Bob chắc chắn thắng quá nửa số lần đoán.
15. Từ Fibonacci. Viết chương trình `FibonacciWord.cpp` in ra các từ Fibonacci từ 0 đến 10. Biết rằng  $f(0) = "a"$ ,  $f(1) = "b"$ ,  $f(2) = "ba"$ ,  $f(3) = "bab"$ ,  $f(4) = "babba"$ , tổng quát là  $f(n) = f(n-1)$  tiếp theo là  $f(n-2)$ . Hãy dùng phép cộng đối với string (chẳng hạn  $f2 = f1 + f0$ ).
16. Viết một chương trình đọc ba số nguyên từ bàn phím và in ra "true" nếu cả ba số bằng nhau, nếu không thì in ra "false".
17. Viết chương trình `LeapYear` kiểm tra năm nhuận. Chương trình đọc vào một số nguyên từ bàn phím và in ra "true" nếu đó là một năm nhuận, nếu không thì in ra "false".
18. Viết chương trình `EuclidDistance` lần lượt đọc 4 số thực từ bàn phím  $x_1$ ,  $y_1$ ,  $x_2$ , và  $y_2$  in ra khoảng cách Euclid giữa hai điểm  $(x_1, y_1)$  và  $(x_2, y_2)$ . Gợi ý: include thư viện `cmath` để dùng hàm tính căn bậc hai `sqrt()` trong thư viện đó.
19. Viết một chương trình `Ordered` đọc ba số nguyên  $x$ ,  $y$ ,  $z$  từ bàn phím. Tạo một biến bool  $b$  và gán cho nó giá trị true nếu ba giá trị đã nhập theo thứ tự tăng dần hoặc giảm dần, false nếu không phải vậy. In ra giá trị của  $b$ .
20. Viết chương trình `Divisibility` đọc hai số nguyên từ bàn phím và in ra 'true' nếu cả hai đều chia hết cho 7 và false nếu không phải như vậy.
21. Viết chương trình `TriangleArea` tính diện tích tam giác. Chương trình nhập độ dài 3 cạnh tam giác  $a$ ,  $b$ ,  $c$  từ bàn phím và in ra diện tích của tam giác đó. Sử dụng công thức diện tích Heron:  $area = \sqrt{s(s-a)(s-b)(s-c)}$ , với  $s = (a + b + c) / 2$ .
22. Body mass index - BMI là tỷ lệ giữa cân nặng của một người (tính theo kilogram) với bình phương chiều cao của người đó (tính theo meter). Hãy viết chương trình BMI nhập hai giá trị cân nặng và chiều cao từ bàn phím và in chỉ số BMI ra màn hình.
23. Viết một chương trình `MaxWithoutIf` nhập hai số nguyên và in ra số lớn hơn trong hai số đó. Trong chương trình không được dùng lệnh if hay cấu trúc  $(a > b)? b : a$ ;. Gợi ý: dùng hàm `abs` trong thư viện `cstdlib`.
24. Viết chương trình `DayOfWeek` nhập ba số nguyên ngày, tháng, năm theo thứ tự đó và in ra thứ của ngày đó ('Monday', 'Tuesday', .... 'Sunday'). Hãy dùng lệnh `switch` trong chương trình.
25. Viết một chương trình nhập 5 số từ bàn phím sau đó in ra số đứng giữa (lớn hơn hai số và nhỏ hơn hai số) trong 5 số đó. Chú ý sao cho chương trình không bao giờ dùng quá 6 phép so sánh.

## C. Luyện tập vòng lặp

26. Thống kê đơn giản. Cho N số nguyên, hãy viết một chương trình BasicStatistics tính giá trị trung bình (double), giá trị lớn nhất và giá trị nhỏ nhất. Chương trình cần nhập N là số các số nguyên, sau đó lần lượt nhập từng số nguyên trong N số đó. Ví dụ

Input: đầu tiên là số N (3), tiếp theo là N số (2,1,3):

3  
2  
1  
2

Với input trên, output cần có là:

Mean: 2  
Max: 3  
Min: 1

27. Viết một chương trình lặp vô tận. Trong mỗi lần lặp, nó đọc vào một số nguyên N kiểu int, nếu N không âm và chia hết cho 5 thì chương trình in ra thương của phép chia N cho 5, nếu không thì in ra -1. Gợi ý: dùng toán tử (?:) và phép đồng dư (%)  
Sửa chương trình để nó kết thúc khi N = -1, trước khi kết thúc cần in ra lời chào 'Bye'.  
Gợi ý: dùng lệnh break.

28. Viết một chương trình dùng 01 vòng lặp và 04 điều kiện để in ra đoạn sau:

12 midnight  
1am  
2am  
...  
12 noon  
1pm  
...  
11pm

29. Viết chương trình nhập một số nguyên dương N và in ra một tam giác kích thước N.  
Chẳng hạn với N bằng 5 thì tam giác có dạng như sau:

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

30. Tương tự bài trên với dạng tam giác sau (với N bằng 5):

\*\*\*\*\*  
\*\*\*\*  
\*\*\*

\*\*

\*