

§1 CÀI ĐẶT CODE BLOCK VÀ CHƯƠNG TRÌNH ĐẦU TIÊN

I. Cài đặt Code Block

Như các bạn đã biết, CodeBlocks là một trong những IDE được sử dụng phổ biến để lập trình các ứng dụng viết bằng C và C++.

1. Tải Code Block

Đầu tiên, các bạn cần vào trang chủ của CodeBlocks để thực hiện tải về, vào link tải này <http://www.codeblocks.org/downloads/binaries> để thực hiện download trực tiếp CodeBlocks IDE bản đầy đủ cho Windows, bao gồm cả trình biên dịch GCC/G++ để biên dịch chương trình.

Việc cài đặt thì các bạn cứ để như mặc định là được, không có vấn đề gì cả. <http://www.codeblocks.org/downloads/binaries>



Windows XP / Vista / 7 / 8.x / 10:

File	Date	Download from
codeblocks-16.01-setup.exe	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01-setup-nonadmin.exe	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01-nosetup.zip	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01mingw-setup.exe	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01mingw-nosetup.zip	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01mingw_fortran-setup.exe	28 Jan 2016	Sourceforge.net or FossHub

2. Cài đặt Code Blocks

- Click đúp vào file vừa tải về để bắt đầu quá trình cài đặt
- Click next liên tiếp, màn hình thông báo cài đặt mặc định vào C:\Program Files\CodeBlocks
- Chọn cài FULL để có đầy đủ các tính năng
- Mở chương trình lên sau khi cài đặt xong.

3. Khởi động Code Blocks

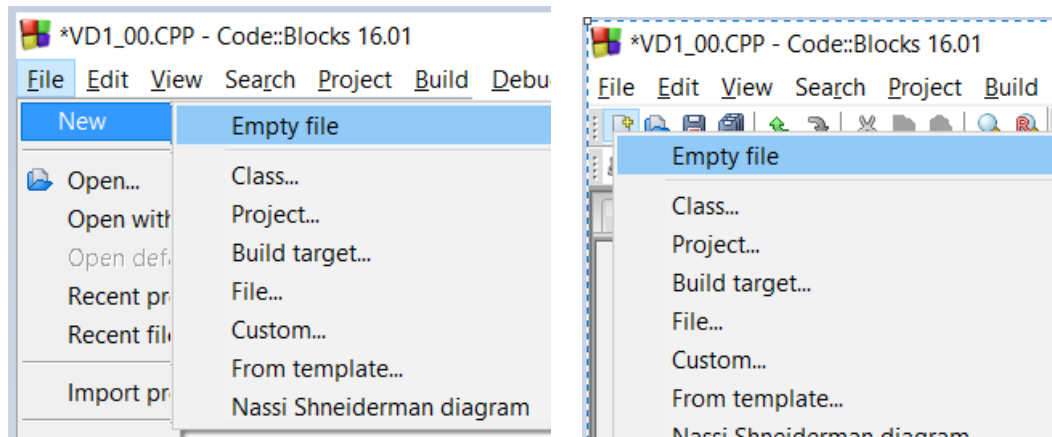
Kích đúp vào biểu tượng Code Block trên màn hình Desktop.

Lần đầu khởi động có 1 cửa sổ hiển thị lên yêu cầu bạn chọn Trình biên dịch mặc định của Code::Blocks.



Để tạo một file trống ta thực hiện:

- File \ New \ Empty file (Ctrl-Shift-N)
- Kích chọn biểu tượng New và chọn Empty file



4. Khái niệm thư viện và hàm main trong C++

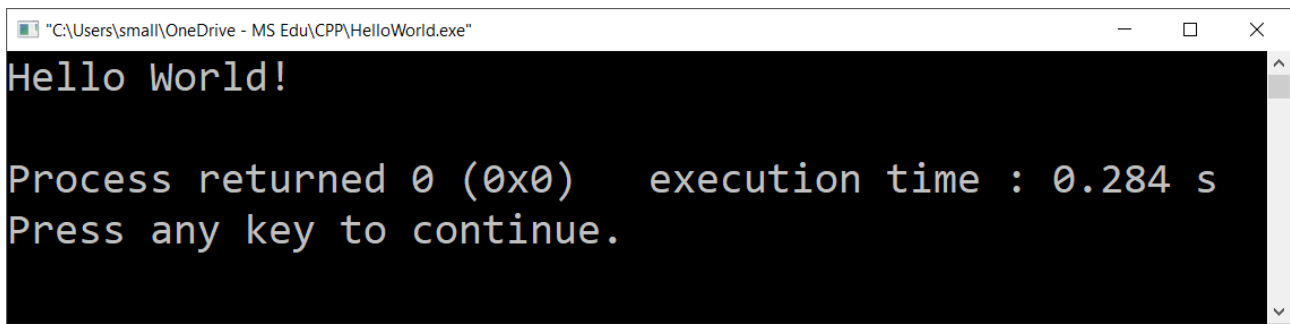
Để bắt đầu tìm hiểu C++ thì bắt buộc bạn phải hiểu hai khái niệm thư viện và hàm main vì trong chương trình đầu tiên mà bạn học sẽ phải khai báo sử dụng thư viện và viết những đoạn code chính trong hàm main.

Trước khi vào tìm hiểu thì ta sẽ viết một chương trình Hello World trước.

Đầu tiên bạn tạo một file HelloWorld.cpp sau đó nhập đoạn code sau:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout<<"Hello World!"<<endl;
7  }
8
```

Sau đó bạn nhấn **F9** thì chương trình sẽ được chạy và bạn sẽ thấy kết quả như hình dưới đây:



```
"C:\Users\small\OneDrive - MS Edu\CPP\HelloWorld.exe"
Hello World!
Process returned 0 (0x0) execution time : 0.284 s
Press any key to continue.
```

Chương trình trên có các phần sau:

Ngôn ngữ C++ định nghĩa một số header, mà chứa thông tin cần thiết và hữu ích cho chương trình của bạn. Với chương trình này, header là **<iostream>** là thư viện chứa các câu lệnh **cin**, **cout**.

Dòng **using namespace std**; nói cho compiler (trình biên dịch) sử dụng std namespace. Namespace là phần bổ sung gần đây cho C++.

Dòng **int main()** là hàm main, tại đây việc thực thi chương trình bắt đầu.

Dòng tiếp theo **cout << "Hello World"**; để in dòng chữ "Hello World" trên màn hình.


Dòng tiếp theo **return 0**; kết thúc hàm main() và làm nó trả về giá trị 0 tới tiến trình đang gọi.

5. Hướng dẫn sử dụng trang <http://lequydon.ntucoder.net>

Tạo tài khoản

- Đăng nhập trang web chấm bài: <http://lequydon.ntucoder.net>
- Kích chọn Đăng kí ở góc trên bên phải
- Nhập các thông tin như hình dưới

Xin vui lòng nhập thông tin (*: cần nhập)

Tài khoản	<input type="text" value="Nhập tên tài khoản"/>	*
Email	<input type="text" value="nhập email thật (chưa có thì phải tạo trước)"/>	*
Mật khẩu	<input type="password" value="Nhập mật khẩu (nhớ mật khẩu để đăng nhập)"/>	*
Xác nhận mật khẩu	<input type="password" value="Nhập lại mật khẩu"/>	*
Họ tên - Lớp - MSSV	<input type="text" value="Nhập họ tên, lớp và trường mình đang học"/>	*
Ngôn ngữ sử dụng	<input type="text" value="Chọn ngôn ngữ là GNU C++"/>	*
		
Mã bảo vệ	<input type="text" value="Nhập mã bảo vệ ở trên"/>	*
	<input type="button" value="Đăng ký"/>	← Kích chọn Đăng ký

Nộp bài

- Chọn bài cần làm

Huy -

§2 CÁC THÀNH PHẦN CỦA NGÔN NGỮ LẬP TRÌNH C++

1. Dấu chấm phẩy và khối (block) trong C++

Trong C++, dấu chấm phẩy là ký tự kết thúc lệnh (statement terminator). Nghĩa là, mỗi lệnh đơn phải kết thúc bởi một dấu chấm phẩy. Nó chỉ dẫn sự kết thúc của một thực thể logic.

Dưới đây là ví dụ về 3 lệnh khác nhau:

```
int n;  
cin>>n;  
cout<<2*n<<endl;
```

Một khối (block) là một tập hợp các lệnh được kết nối một cách logic, mà được bao quanh bởi các dấu ngoặc móc mở và đóng. Ví dụ:

```
{  
    int n;  
    cin>>n;  
    cout<<2*n<<endl;  
}
```

Hai dạng sau là tương đương, C++ không quan tâm bạn đặt bao nhiêu lệnh trên một dòng. Ví dụ:

```
int n;  
cin>>n;  
cout<<2*n<<endl;
```

Là giống với:

```
int n; cin>>n; cout<<2*n<<endl;
```

2. Cặp dấu ghi chú thích

Khi viết chương trình đôi lúc ta cần phải có vài lời ghi chú về 1 đoạn chương trình nào đó để dễ nhớ và dễ điều chỉnh sau này; nhất là phần nội dung ghi chú phải không thuộc về chương trình (khi biên dịch phần này bị bỏ qua). Trong ngôn ngữ lập trình C/C++, nội dung chú thích phải được viết trong cặp dấu /* và */. Nếu chú thích nằm trên một dòng thì dùng dấu // nằm trước dòng cần làm chú thích

/ Đây là chú thích nằm trên 1 hay nhiều dòng*/*

//Đây là chú thích nằm trên 1 dòng

3. Từ khóa

Từ khóa là các từ dành riêng (reserved words) của C mà người lập trình có thể sử dụng nó trong chương trình tùy theo ý nghĩa của từng từ. Ta không được dùng từ khóa để đặt cho các tên của riêng mình

Các từ khóa của C++:

asm	auto	break	case	cdecl	char
class	const	continue	_cs	default	delete
do	double	_ds	else	enum	_es
extern	_export	far	_fastcall	float	for
friend	goto	huge	if	inline	int
interrupt	_loadds	long	near	new	operator
pascal	private	protected	public	register	return
_saveregs	_seg	short	signed	sizeof	_ss
static	struct	switch	template	this	typedef
union	unsigned	virtual	void	volatile	while

4. Tên (định danh)

Tên hay còn gọi là định danh (identifier) được dùng để đặt cho chương trình, hằng, kiểu, biến, chương trình con... Tên có hai loại là tên chuẩn và tên do người lập trình đặt.

Tên chuẩn là tên do C đặt sẵn như: printf, scanf, sin, cos...

Tên do người lập trình tự đặt để dùng trong chương trình của mình. Sử dụng bộ chữ cái, chữ số và dấu gạch dưới (_) để đặt tên, nhưng phải tuân thủ quy tắc:

- Bắt đầu bằng một chữ cái hoặc dấu gạch dưới.
- Không có khoảng trống ở giữa tên.
- Không được trùng với từ khóa.
- Độ dài tối đa của tên là không giới hạn, tuy nhiên chỉ có 31 ký tự đầu tiên là có ý nghĩa.
- Không cấm việc đặt tên trùng với tên chuẩn nhưng khi đó ý nghĩa của tên chuẩn không còn giá trị nữa.

Ví dụ: tên do người lập trình đặt: Chieu_dai, Chieu_Rong, Chu_Vi, Dien_Tich

Các tên sau đây không hợp lệ:

- Do Dai → sử dụng dấu cách
- 12A2 → bắt đầu là chữ số
- num-odd → sử dụng dấu gạch ngang
- int → trùng với từ khóa
- f(x) → có dấu ngoặc tròn

§3 KIỂU DỮ LIỆU TRONG C++ ???

I. Kiểu dữ liệu trong C/C++

Trong khi làm việc với bất kỳ ngôn ngữ lập trình nào, bạn cần sử dụng các kiểu biến đa dạng để lưu giữ thông tin. Các biến, không gì khác ngoài các vị trí bộ nhớ được dành riêng để lưu giá trị. Nghĩa là, khi bạn tạo một biến, bạn dành riêng một số không gian trong bộ nhớ cho biến đó.

Bạn có thể thích lưu thông tin của các kiểu dữ liệu (Data Type) đa dạng như Character, Wide Character, integer, floating-point, double floating point, Boolean,... Dựa trên kiểu dữ liệu của một biến, hệ thống sẽ cấp phát bộ nhớ và quyết định những gì có thể được lưu giữ trong bộ nhớ dành riêng đó.

1. Kiểu ký tự - char

Kiểu char dùng để lưu trữ 1 ký tự trong bảng mã ASCII.

Bảng mã ASCII (American Standard Code for Information Interchange)

0		20	¶	40	<	60	<	80	P	100	d	120	x
1	☺	21	☹	41	>	61	=	81	Q	101	e	121	y
2	☹	22	—	42	*~	62	>	82	R	102	f	122	z
3	♥	23	↑	43	+	63	?	83	S	103	g	123	{
4	♦	24	↑	44	, -	64	@	84	T	104	h	124	
5	♣	25	↓	45	-	65	A	85	U	105	i	125	}
6	♠	26	→	46	.	66	B	86	V	106	j	126	~
7		27	←	47	/	67	C	87	W	107	k	127	Δ
8		28	⌞	48	0	68	D	88	X	108	l	128	Ç
9		29	↔	49	1	69	E	89	Y	109	m	129	ü
10		30	▲	50	2	70	F	90	Z	110	n	130	é
11	♂	31	▼	51	3	71	G	91	[111	o	131	â
12	♀	32	!	52	4	72	H	92	\	112	p	132	ä
13		33	"	53	5	73	I	93] ^	113	q	133	å
14	☾	34	#	54	6	74	J	94	⌘	114	r	134	æ
15	*	35	\$	55	7	75	K	95	˘	115	s	135	ç
16	▶	36	%	56	8	76	L	96	˘	116	t	136	ë
17	◀	37	&	57	9	77	M	97	a	117	u	137	è
18	↑	38	'	58	:	78	N	98	b	118	v	138	ì
19	!!	39	,	59	;	79	O	99	c	119	w	139	

Kiểu dữ liệu	Kích thước	Miền giá trị
unsigned char	1 byte	$0 \div 255$
char	1 byte	$-128 \div 127$

Người ta dùng dấu nháy đơn cho ký tự và dấu nháy kép cho chuỗi.

Ví dụ:

Ký tự: 'a', '@', '~'

Chuỗi: "a", "abc"

2. Kiểu nguyên

Kiểu số nguyên trong C++ được dùng để lưu trữ số nguyên. Đây là kiểu dữ liệu cơ bản nhất trong các ngôn ngữ lập trình.

Kiểu dữ liệu	Kích thước	Miền giá trị
int	4 byte	$-2147483648 \div 2147483647$
long long	8 byte	$-9223372036854775808 \div 9223372036854775807$

3. Kiểu thực

Kiểu số thực trong C++ được dùng để lưu trữ số thực.

Kiểu dữ liệu	Kích thước	Miền giá trị
float	4 byte	$\pm 10^{-37} \div \pm 10^{+38}$
double	8 byte	$\pm 10^{-307} \div \pm 10^{+308}$

II. Hằng và biến

1. Hằng (Constant)

Là đại lượng không đổi trong suốt quá trình thực thi của chương trình.

Hằng có thể là một chuỗi ký tự, một ký tự, một con số xác định. Chúng có thể được biểu diễn hay định dạng (Format) với nhiều dạng thức khác nhau.

Cách khai báo hằng: Khai báo hằng bạn có thể khai báo ở bất kỳ đâu trong chương trình, khai báo ở đâu thì từ đó hằng sẽ được xác định.

Cú pháp: `#define ten_hang gia_tri` (Không có dấu chấm phẩy ở cuối)

Ví dụ:

```
#define MAX 1000;
```

```
#define ten "Small";
```

Chú ý chúng ta không dùng cấu trúc const như một số người dùng vì khi dùng const trong một số trường hợp ta vẫn thay đổi được giá trị của hằng.

2. Biến

Biến là một đại lượng được người lập trình định nghĩa và được đặt tên thông qua việc khai báo biến. Biến dùng để chứa dữ liệu trong quá trình thực hiện chương trình và giá trị của biến có thể bị thay đổi trong quá trình này. Cách đặt tên biến giống như cách đặt tên đã nói trong phần trên.

Mỗi biến thuộc về một kiểu dữ liệu xác định và có giá trị thuộc kiểu đó.

Cú pháp: **kiểu_dữ_liệu** **tên_biến**;

Ví dụ:

```
char ch='a';  
int a,b=8;  
long long s;
```

Giải thích ví dụ:

Biến **ch** kiểu ký tự chứa ký tự **'a'**

Biến **a** kiểu số nguyên chưa có giá trị; biến **b** kiểu nguyên chứa giá trị là 8

Biến **s** kiểu số nguyên lớn chưa có giá trị

III. Nhập, xuất dữ liệu

1. Header *iostream*

File này định nghĩa các đối tượng **cin**, **cout**, **cerr** và **clog**, tương ứng với Standard Input Stream (Luồng đầu vào chuẩn), Standard Output Stream (Luồng đầu ra chuẩn), Un-buffered Standard Error Stream (Luồng lỗi chuẩn không được đệm) và Buffered Standard Error Stream (Luồng lỗi chuẩn được đệm)

2. *Standard Output Stream (cout)*

Đối tượng tiền định nghĩa **cout** là một minh họa của lớp **iostream**. Đối tượng **cout** được xem như "được kết nối tới" thiết bị đầu ra chuẩn, thường là màn hình. Đối tượng **cout** được sử dụng kết hợp với toán tử chèn luồng (insertion operator), được viết là **<<**, xem ví dụ dưới đây:

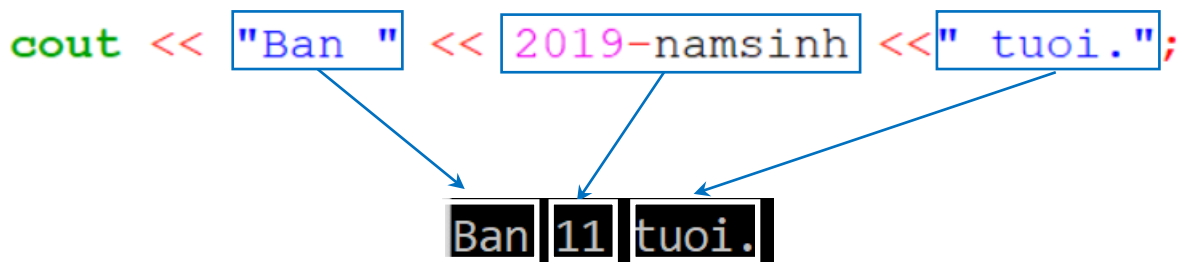
```
#include <iostream>
using namespace std;
int main()
{
    int namsinh;
    cout<<"Ban sinh nam bao nhieu: ";
    cin>>namsinh;
    cout << "Ban " << 2019-namsinh <<" tuoi.";
    return 0;
}
```

"C:\Users\small\OneDrive - MS Edu\CPP\test.exe"

Ban sinh nam bao nhieu: 2008
Ban 11 tuoi.
Process returned 0 (0x0) execution time : 10.959
Press any key to continue.

Bộ biên dịch C++ cũng quyết định kiểu dữ liệu của biến để được output và chọn toán tử chèn luồng thích hợp để hiển thị giá trị. Toán tử << được nạp chồng (overload) tới dữ liệu đầu ra của kiểu integer, float, double, string, và giá trị con trỏ có sẵn.

Toán tử chèn luồng << có thể được sử dụng nhiều hơn một lần trong một lệnh và endl được sử dụng để thêm một dòng mới tại cuối dòng đó.



3. Standard Input Stream (cin)

Đối tượng tiền định nghĩa cin là một minh họa của lớp istream. Đối tượng cin được xem như đính kèm với thiết bị đầu vào chuẩn, mà thường là bàn phím. Đối tượng cin được sử dụng kết hợp với toán tử trích luồng (extraction operator), viết là >>, như trong ví dụ ở trên:

4. Thư viện định dạng output

iomanip viết tắt của cụm từ iostream manipulator là một thư viện thuộc namespace std, nó định nghĩa một số hàm giúp lập trình viên có thể định dạng output.

Trong bài này, chúng ta chỉ làm quen với một số hàm đơn giản thường xuyên được sử dụng.

Hàm `setw(int n)`

setw là một hàm cho phép giới hạn độ rộng của một giá trị được xuất lên màn hình.

Cách sử dụng:

```
#include <iostream>           // cout, endl
#include <iomanip>             // setw
using namespace std;

int main () {
    cout << setw(10) << 0.1277 << endl;
    cout << setw(7) << 77 << endl;
    cout << setw(4) << 123456789 << endl;
    return 0;
}
```

```
"C:\Users\small\OneDrive - MS Edu\CPP\test.exe"
0.1277
77
123456789
```

Hàm `setprecision(int n)`

Cũng tương tự như hàm **setw**, hàm **setprecision** cũng nhận vào một giá trị số nguyên, nhưng mục đích của hàm này là định dạng số lượng chữ số trong phần thập phân của kiểu số thực.

Cách sử dụng:

```

#include <iostream>           // cout, fixed
#include <iomanip>             // setprecision
using namespace std;
int main () {
    double f =3.14159;
    cout << setprecision(5) << f << '\n';
    cout << setprecision(9) << f << '\n';
    cout << fixed;
    cout << setprecision(5) << f << '\n';
    cout << setprecision(9) << f << '\n';
    return 0;
}

```

"C:\Users\small\OneDrive - MS Edu\CPP\test.exe"

```

3.1416
3.14159
3.14159
3.141590000

```

Hãy quan sát và nhận xét kết quả trước/sau khi sử dụng hàm `cout<<fixed;`