NHẬP MÔN LẬP TRÌNH

CHƯƠNG 1.2: GIỚI THIỆU VỀ C++

ThS. Nguyễn Thị Ngọc Diễm
diemntn@uit.edu.vn

Nội dung

1. Giới thiệu tổng quan về ngôn ngữ C++
2. Giới thiệu môi trường, công cụ hỗ trợ việc lập trình
3. Quy tắc soạn thảo mã nguồn
4. Quy trình tổng quát viết, dịch, chạy thử chương trình
5. Giới thiệu sơ lược về cấu trúc chương trình
6. Một số ví dụ minh họa về chương trình C++ và chạy thử
7. Một số quy tắc cần nhớ khi viết chương trình

Ví dụ về 1 đoạn chương trình C++

```
#include<iostream>
using namespace std;

int main() {
	int a, b;

	cout << "GIAI PHUONG TRINH BAC 1 CO DANG ax+b=0." << endl;
	cout << "Nhap he so a va b: ";  cin >> a >> b;

	if (a != 0) {
		cout << "Phuong trinh co 1 nghiem: "<< float(-b)/a << endl;
	} else {
		if (b == 0)
			cout << "Phuong trinh co Vo so nghiem" << endl;
		else
			cout << "Phuong trinh co Vo nghiem" << endl;
	}

	return 0;
}
```

1. Giới thiệu tổng quan về ngôn ngữ C++

- Là ngôn ngữ lập trình hỗ trợ lập trình thủ tục, dữ liệu trừu trượng, lập trình hướng đối tượng, và lập trình đa hình.
- C++ được coi như là ngôn ngữ bậc trung (middle-level), nó kết hợp các đặc điểm và tính năng của ngôn ngữ bậc cao và bậc thấp.
- C++ được phát triển bởi Bjarne Stroustrup năm 1979 tại Bell Labs ở Murray Hill, New Jersey, như là một bản nâng cao của ngôn ngữ C và với tên gọi đầu tiên là “C với các Lớp”, nhưng sau đó được đổi tên thành C++ vào năm 1983.
- Và C++ còn là ngôn ngữ được cải tiến từ C, và bất kỳ chương trình C nào cũng là một chương trình C++.
- Hiện tại tiêu chuẩn mới nhất của ngôn ngữ C++ là C++17.
  - Nhiều ngôn ngữ ảnh hưởng bởi C++: C#, Java.

Tiêu chuẩn ngôn ngữ C++

Year	*C++ Standard*	*Informal name*
1998	ISO/IEC 14882:1998	C++98
2003	ISO/IEC 14882:2003	C++03
2007	ISO/IEC TR 19768:2007	C++07/TR1
2011	ISO/IEC 14882:2011	C++11, C++0x
2014	ISO/IEC 14882:2014	C++14, C++1y
2017	ISO/IEC 14882:2017	C++17, C++1z
2020	to be determined	C++20

Bộ từ khóa của C++

- Từ khóa (keyword)
  - Các từ dành riêng trong ngôn ngữ.
  - Không thể sử dụng từ khóa để đặt tên cho biến, hàm, tên chương trình con.

Bộ từ khóa của C++

- http://en.cppreference.com/w/cpp/keyword

alignas	const	for	protected	true
alignof	constexpr	friend	public	try
and	const_cast	goto	register	typedef
and_eq	continue	if	reinterpret_cast	typeid
asm	decltype	inline	requires	typename
auto	default	int	return	union
bitand	delete	long	short	unsigned
bitor	do	mutable	signed	using
bool	double	namespace	sizeof	virtual
break	dynamic_cast	new	static	void
case	else	noexcept	static_assert	volatile
catch	enum	not	static_cast	wchar_t
char	explicit	not_eq	struct	while
char16_t	export	nullptr	switch	xor
char32_t	extern	operator	template	xor_eq
class	false	or	this	…
compl	float	or_eq	thread_local	
concept		private	throw	

Các kiểu dữ liệu

Data Types

Type	Length	Range
unsigned char	8 bits	0 to 255
char	8 bits	-128 to 127
enum	16 bits	-32,768 to 32,767
unsigned int	16 bits	0 to 65,535
short int	16 bits	-32,768 to 32,767
int	16 bits	-32,768 to 32,767
unsigned long	32 bits	0 to 4,294,967,295
long	32 bits	-2,147,483,648 to 2,147,483,647
float	32 bits	3.4 * (10**-38) to 3.4 * (10**+38)
double	64 bits	1.7 * (10**-308) to 1.7 * (10**+308)
long double	80 bits	3.4 * (10**-4932) to 1.1 * (10**+4932)

Các toán tử

[ ]	( )	.	->	++	--	&
*	+	-	~	!	sizeof	/
%	<<	>>	<	>	<=	>=
==	!=	^	|	&&	||	?:
=	*=	/=	%=	+=	-=	<<=
>>=	&=	^=	|=	,	#	##

The following operators are specific to C++:

:: .* ->*

The operators # and ## are used only by the preprocessor.

Depending on context, the same operator can have more than one meaning. For example, the ampersand (&) can be interpreted as

- a bitwise AND (A & B)
- an address operator (&A)
- in C++, a reference modifier

In the first case, the & is a unary operator; in the second, the & is a binary operator.

- **Môi trường phát triển tích hợp IDE (Integrated Development Environment)**
  - Trình soạn thảo (SOURCE CODE EDITOR).
  - Trình Biên dịch/ Thông dịch (COMPILE/ INTERPERTER).
  - Công cụ xây dựng tự động (AUTO BUILD).
  - Trình gỡ lỗi hỗ trợ dò tìm lỗi (Trình DEBUGER).
  - …

- Một số IDE hỗ trợ lập trình C++: Code:Blocks, Dev C++, Eclipse, NetBeans, Microsoft Visual Studio, …

- Todo: Thêm hình ảnh minh họa Visual Studio hoặc Code:Blocks

- Ví dụ Tên/Định danh (Identifier)
  - ? Tìm các tên không hợp lệ:

    Ma-Lop, GiaiPhuongTrinh, MãSinhViên, 1A, Bai_Tap_1 , Giai Phuong Trinh, if, f(x)

    ⇒Tên không hợp lệ:

Ma-Lop	Sử dụng dấu gạch ngang
MãSinhViên	Có dấu
1A	Ký tự đầu là số
Giai Phuong Trinh	Sử dụng khoảng trắng
if	Trùng tư khóa
f(x)	Sử dụng ký tự ngoặc tròn

3. Quy tắc soạn thảo mã nguồn

- ? Các bộ tên sau đây có giống nhau không?
  - A, a
  - BaiTap, baitap, BAITAP, bAItaP, …

=> Vì C++ có tính chất case sentitive nên các tên trên là phân biệt

3. Quy tắc soạn thảo mã nguồn

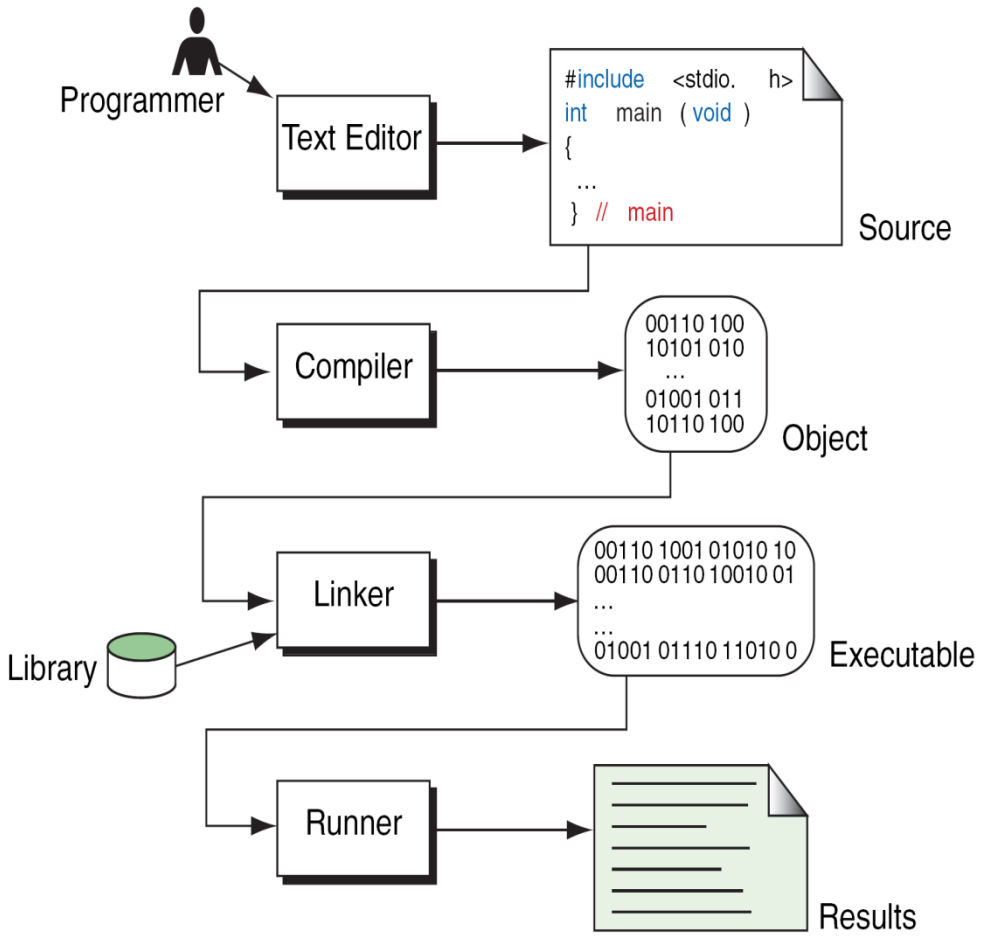
- Dấu chấm phẩy ;
  - Dùng để phân cách các câu lệnh.
  - Ví dụ: cout << “Hello World!”;
- Câu chú thích
  - Đặt giữa cặp dấu /* */ hoặc // (C++)
  - Ví dụ: /*Ho & Ten: NVA*/, // MSSV: 0712078
- Hằng số, hằng ký tự và hằng chuỗi
  - Hằng số: 1, 200, …
  - Hằng ký tự: ‘A’, ‘a’, …
  - Hằng chuỗi: “Hello World!”, “Nguyen Van A”
  - Chú ý: ‘A’ khác “A”

3. Quy tắc soạn thảo mã nguồn

- Code canh lề ngay ngắn sẽ dễ đọc, dễ nắm bắt cấu trúc của chương trình từ đó dễ phát hiện và sửa lỗi, ngăn ngừa việc tạo ra chương trình lỗi. Có nhiều kiểu canh lề khác nhau, trong đó kinh điển nhất là kiểu K&R, được hỗ trợ bởi hầu hết các IDE hiện đại.

4. Qui trình tổng quát viết, dịch, chạy thử chương trình

- **Writing** source code as an C++ file.
  - e.g. "*hello.cpp*" file
- **Preprocessing**
  - Processes the source code for compilation.
- **Compilation**
  - Checks the grammatical rules (syntax).
  - Source code is converted to object code in machine language (e.g. "*hello.obj*" file)
- **Linking**
  - Combines object code and libraries to create an executable (e.g. "*hello.exe*" file).
  - Library: common functions (input, output, math, etc).



5. Giới thiệu sơ lược về cấu trúc chương trình

```
# Tiền xử lý
```

```
Khai báo biến, hàm, ...
```

```
int main() {

        Thân hàm chính

}
```

```
Định nghĩa hàm
```

Ví dụ 1

```
/* Chương trình tính chu vi và diện tích hình tròn */
#include<iostream>
using namespace std;
#define PI 3.14

int main() {
    float BanKinh, ChuVi, DienTich;

    BanKinh = 20;
    ChuVi = BanKinh * 2 * PI;
    DienTich = BanKinh * BanKinh * PI;

    cout << "Chu vi hinh tron: " << ChuVi << endl;
    cout << "Dien tich hinh tron: " << DienTich << endl;
}
```

#Tiền xử lý

Thân hàm chính

6. Một số ví dụ minh họa về chương trình C++ và chạy thử

Ví dụ 1

Xuất đơn giản như “Hello World”

```
#include <iostream>
using namespace std;
int main() {
   cout << "Hello, world!\n";
}
```

Ví dụ 2

Chương trình có nhập xuất dữ liệu và tính toán xử lý đơn giản như “Nhập độ dài 2 cạnh của hình chữ nhật, xuất diện tích của hình”

```
/* Minh hoa chuong trinh tinh dien tich hinh chu nhat */

#include <iostream>
using namespace std;
int main() {
    int chieu_dai,chieu_rong;
    cout << "Nhap chieu dai = ";
    cin >> chieu_dai;
    cout << "Nhap chieu rong = ";
    cin >> chieu_rong;
    // Tinh dien tich hinh chu nhat
    int dien_tich = chieu_dai*chieu_rong;
    // In ket qua ra man hinh
    cout << "Dien tich HCN  = " << dien_tich;
}
```

Ví dụ 3: Cách 1

Chương trình phức tạp hơn, có sử dụng vòng lặp: kiểm tra một số nguyên n có phải là số nguyên tố không

```
/* Chuong trinh minh hoa kiem tra so nguyen to */
#include <iostream>
using namespace std;
// Ham kiem tra so nguyen n co phai la so nguyen to (true)
hay khong (false).
bool kiemtralasonguyento(int n) {
   if( n<2) return false;
   for(int i=2; i<n; i++)
         if(n%i==0) return false;

   return true;
}
int main() {
     int n;
     cout << "Nhap vao so nguyen n = ";
     cin >> n;
     if (kiemtralasonguyento(n)== true)
     cout << "So " << n << " la so nguyen to !";
     else
     cout << "So " << n << " khong phai la so nguyen to !";
}
```

Ví dụ 3: Cách 2

Chương trình phức tạp hơn, có sử dụng vòng lặp: kiểm tra một số nguyên n có phải là số nguyên tố không

```
/* Chuong trinh minh hoa kiem tra so nguyen to */
#include <iostream>
using namespace std;
// Ham kiem tra so nguyen n co phai la so nguyen to (true)
hay khong (false).
bool kiemtralasonguyento(int n);
int main() {
     int n;
     cout << "Nhap vao so nguyen n = ";
     cin >> n;
     if (kiemtralasonguyento(n)== true)
     cout << "So " << n << " la so nguyen to !";
     else
     cout << "So " << n << " khong phai la so nguyen to !";
}
bool kiemtralasonguyento(int n) {
   if( n<2) return false;
   for(int i=2; i<n; i++)
         if(n%i==0) return false;

   return true;
}
```

7. Một số quy tắc cần nhớ khi viết chương trình

- Chương trình nên được tách thành nhiều đơn thể (mô-đun), mỗi đơn thể thực hiện một công việc và càng độc lập với nhau.
- Cách trình bày chương trình càng nhất quán sẽ càng dễ đọc và dễ hiểu (định hướng về phong cách lập trình).
- Mỗi câu lệnh có thể viết trên một hay nhiều dòng nhưng phải được kết thúc bằng dấu ;
- Quy tắc viết lời giải thích, lời giải thích không có tác dụng với sự làm việc của chương trình trên máy tính, chỉ có tác dụng với người đọc
- Sử dụng các hàm chuẩn: sử dụng #include
- Hàm chính main

Bài tập lý thuyết

1. Tên (định danh) nào sau đây đặt không hợp lệ, tại sao?

   - Tin hoc co SO A, 1BaiTapKHO, for
   - THucHaNH, NhapMon_L@pTrinH

2. Câu ghi chú dùng để làm gì? Cách sử dụng ra sao? Cho ví dụ minh họa.

3. Trình bày cấu trúc của một chương trình C. Giải thích ý nghĩa của từng phần trong cấu trúc.

Chúc các em học tốt!