CHƯƠNG 1

TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH C++

Giảng viên: Nguyễn Chiến Thắng
Email: thangnc.haui@gmail.com

Nội dung

1. Khái niệm chương trình, lập trình, ngôn ngữ lập trình
2. Các khái niệm cơ bản
3. Hằng, biến, các kiểu dữ liệu
4. Biểu thức và các toán tử
5. Các hàm và các lệnh nhập xuất trong C++

1.1. Khái niệm về lập trình

- Chương trình: là một chuỗi các lệnh điều khiển máy tính giải quyết một bài toán.
- Lập trình: Việc xây dựng các chương trình điều khiển máy tính.
- Ngôn ngữ lập trình: Là ngôn ngữ để viết các chương trình điều khiển máy tính.
- Kỹ thuật lập trình: Vận dụng kiến thức và sử dụng các công cụ lập trình để xây dựng chương trình.

1.2. Các khái niệm cơ bản trong C++

1. Bộ ký tự sử dụng trong C++
2. Các từ khóa
3. Định danh
4. Cấu trúc chương trình C++
5. Câu lệnh và lời giải thích
6. Lưu đồ chương trình
7. Một số chương trình đơn giản

1.2.1. Bộ ký tự sử dụng trong C++

1. Bộ chữ cái: a, b, …, z, A, B, …, Z
2. Chữ số thập phân: 0, 1, 2, …, 9
3. Dấu câu: . , : ; ? !
4. Các phép toán: + - * / = < >
5. Các dấu ngoặc: ( ) { } [ ] ' "
6. Ký tự đặc biệt: | \ ~ & ^ % $ # @ ! _
7. Dấu cách, dấu tab …

1.2.2. Các từ khóa

- Các từ có sẵn trong C++, các từ khóa kết hợp với mã lệnh tạo nên cấu trúc của chương trình.
- Từ khóa khai báo dữ liệu: const, typedef, struct, void, char, int, long, float, double, unsiged, …
- Từ khóa lệnh rẽ nhánh, lệnh nhảy: if…else…, switch…case…default…, goto
- Từ khóa lệnh lặp: for…, do…, while…
- Từ khóa toán tử: break, continue, return, new, sizeof, delete

Chú ý: Tất cả các từ khóa đều viết thường

1.2.3. Định danh

- Tên của các hằng, biến, kiểu dữ liệu, hàm v.v… trong chương trình được gọi là một định danh.
- Ví dụ: sqrt, a, x, b1, b2, hocsinh là các định danh.
- Định danh nhằm để phân biệt các đối tượng trong chương trình với nhau.

1.2.3. Định danh - Qui tắc định danh

- Định danh chỉ được viết bằng các chữ cái, chữ số và dấu gạch nối _.
- Ký tự đầu tiên phải là chữ cái.
- Định danh phân biệt chữ hoa/thường
- Không được có 2 định danh trùng nhau trong một chương trình.
- Định danh không được trùng với từ khóa
- Định danh nên ngắn gọn, dễ nhớ, có ý nghĩa
- Các định danh đúng: hoc_sinh, a1, x2, sinhvien
- Các định danh sai: hoc sinh, 1a2, a$3, sinh-vien

1.2.4. Cấu trúc một chương trình C++

```
1. #include <tệp_tiêu_đề>
2. #define Tên_hằng Giá_trị
3. …
4. Khai báo các kiểu dữ liệu
5. Khai báo các hằng
6. Khai báo biến toàn cục
7. Khai báo hàm/nguyên mẫu hàm
8. int main()
9. {
10.     //Các lệnh thực thi chương trình
11.}
```

Cấu trúc một chương trình C++ (tt)

```
1.  #include <tệp_tiêu_đề>
2.  #define Tên_hằng Giá_trị
3.  …
4.  Khai báo các kiểu dữ liệu
5.  Khai báo các hằng
6.  Khai báo biến toàn cục
7.  Khai báo hàm/nguyên mẫu hàm
8.  int main()
9.  {
10.     //Các lệnh thực thi chương
    trình
11. }
```

- Dòng 1, khai báo thư viện (chỉ thị tiền xử lý): Các hàm, hằng chuẩn của C++ đặt trong các thư viện này.
- Các thư viện hay dùng: conio.h, stdio.h, iostream.h, math.h, ctype.h, iomanip.h, string.h, ofstream.h, ifstream.h, malloc.h, v.v…
- Dòng 2: khai báo hằng tượng trưng
- Các dòng 3, 4, 5, 6, 7: là các khai báo cần thiết
- Các dòng 8, 9, 10, 11: Bắt đầu và kết thúc hàm main(), chứa các lệnh thực thi chương trình

1.2.5. Câu lệnh và lời giải thích

- Mỗi câu lệnh trong chương trình yêu cầu máy tính thực hiện một thao tác.
- Các câu lệnh được ngăn cách nhau bởi dấu chấm phẩy (;)
- Có thể viết một câu lệnh trên nhiều dòng hoặc viết nhiều lệnh trên một dòng.
- Ví dụ:

```
s = a + b;
cout<<"Tong 2 so la "<<s;
```

Câu lệnh và lời giải thích (tt)

- Lời giải thích giúp người đọc dễ hiểu chương trình
- Có thể ghi lời giải thích trên một dòng hoặc nhiều dòng
- Ví dụ:

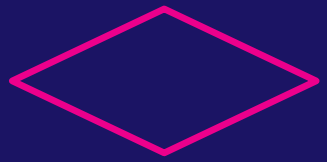
```
// Giải thích trên một dòng
/* Giải thích
        trên
   nhiều dòng */
```

1.2.6. Lưu đồ thuật toán – Chương trình

Các ký hiệu dùng trong lưu đồ

Bắt đầu hoặc kết thúc chương trình

Hướng thực thi chương trình



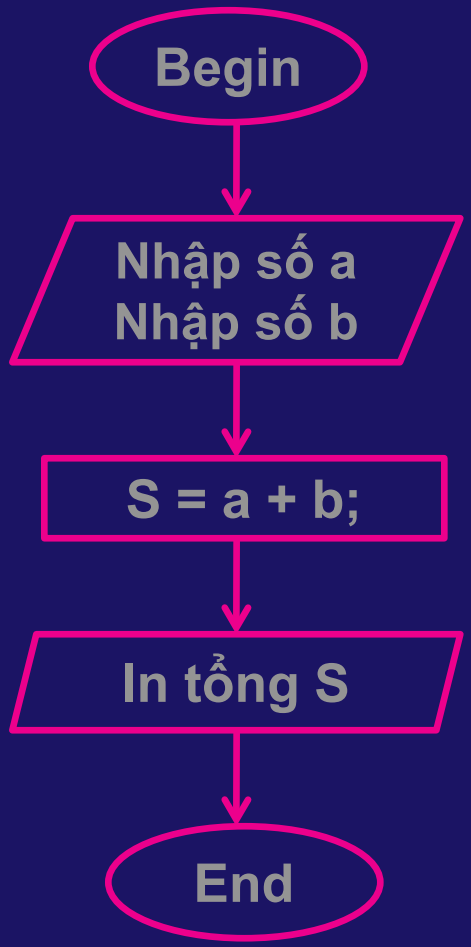
Kiểm tra điều kiện trong chương trình

Thực thi câu lệnh trong chương trình

Nhập/hiển thị dữ liệu của chương trình

Lưu đồ thuật toán – Chương trình (tt)

Ví dụ: Chương trình cộng 2 số




```
#include<iostream>
using namespace std;
int main(){
int a,b,s;
cout<<"Nhap so a: ";
cin>>a;
cout<<"Nhap so b: ";
cin>>b;
s = a + b;
cout<<"Tong la s: "<<s;
getch();
return 0;
}
```

1.2.7. Một số chương trình C++ đơn giản

1. Chương trình cộng 2 số nguyên
2. Chương trình tính diện tích hình thang
3. Một người mua hàng mua n tập vở, giá mỗi tập vở là k đồng, khách hàng được giảm giá là r %. Viết chương trình khởi gán các giá trị cho n, k, r. Tính và in ra màn hình số tiền khách hàng phải trả.

1.3. Hằng, biến, biểu thức và các kiểu dữ liệu

- Hằng và biến
- Các kiểu dữ liệu cơ sở
- Sự chuyển đổi kiểu
- Biểu thức và các toán tử
- Biểu thức điều kiện
- Các hàm toán học trong thư viện math.h

1.3.1. Hằng và biến

- Dữ liệu chứa trong bộ nhớ máy tính có thể là các biến hay các hằng.
- Hằng: là một đại lượng có giá trị không thay đổi
- Biến: là một đại lượng có thể thay đổi được giá trị

Biến và hằng được lưu trữ trong bộ nhớ và thường được đặt tên.

- Kiểu dữ liệu:
  - Là một tập các giá trị dữ liệu mà một biến hoặc 1 hằng thuộc kiểu đó có thể nhận được.
  - Trên mỗi kiểu dữ liệu có một số phép toán được định nghĩa để thao tác dữ liệu.

Hằng và biến (tt)

a. Khai báo hằng

Bằng chỉ thị #define

```
#define tên_hằng giá trị
```

Ví dụ:

```
#define MAX 100
```

Bằng từ khóa const

```
const <Kiểu_DL> tên_hằng = giá_trị;
```

Ví dụ:

```
const int MAX = 100;
```

Hằng và biến (tt)

b. Khai báo biến

<Kiểu_dữ_liệu> <danh_sách_biến>;

Các biến trong danh sách cách nhau bởi dấu phẩy

Ví dụ:

```
int a, b, m, n;
float x, x1, y, z;
```

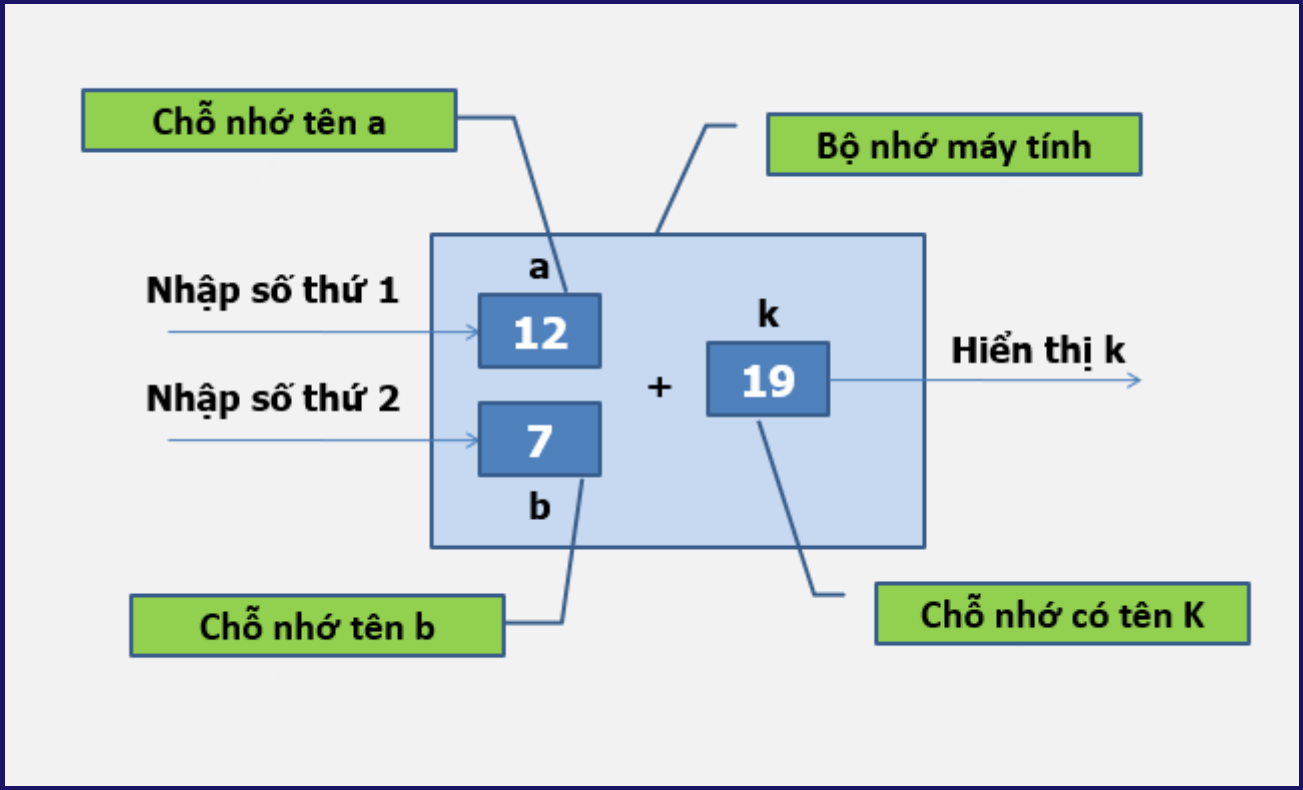
c. Địa chỉ của biến:

- Mỗi biến trong bộ nhớ có địa chỉ là số thứ tự của ô nhớ chứa biến.
- Toán tử & được sử dụng để trả về địa chỉ của biến
- Ví dụ: &x là địa chỉ của biến x trong bộ nhớ

Hằng và biến (tt)

- Biến là tên chỗ nhớ chứa dữ liệu
- Ví dụ nhập 2 số a và b. Tính k = a + b.
- Khi đó: 2 số nhập vào được lưu trong bộ nhớ máy tính với 2 chỗ nhớ có tên là a và b.
- Số k cũng được lưu trong bộ nhớ với chỗ nhớ có tên k
- Trong chương trình các chỗ nhớ có tên này cần được khai báo và chúng được gọi là các biến.

Hằng và biến - Mô tả biến



Hằng và biến - Mô tả biến

```
#include<iostream>
using namespace std;
int main()
{
      int k, a, b;
      cout<<"Nhap so a:"; cin>>a;
      cout<<"Nhap so b:"; cin>>b;
      k = a + b;
      cout<<"Tong la k:"<<k;
      return 0;
}
```

1.3.2. Các kiểu dữ liệu cơ sở

Tên kiểu DL	Miền giá trị	KT
int	-32768 ⇨ +32767	2 bytes
unsigned int	0 ⇨ 65535	2 bytes
long	– 2.147.483.648 ⇨ 2.147.483.647	4 bytes
unsigned long	0 ⇨ 4.294.967.295	4 bytes
float	$3.4*(10^{-38})$ ⇨ $3.4*(10^{38})$	4 bytes
double	$1.7*(10^{-308})$ ⇨ $1.7*(10^{308})$	8 bytes
long double	$3.4*(10^{-4932})$ ⇨ $1.1*(10^{4932})$	10 bytes
char	-128 ⇨ +127	1 byte
unsigned	0 ⇨ 255	1 byte
void	Không có giá trị nào	
bool	true, false	1 bit

1.3.3. Sự chuyển đổi kiểu dữ liệu

- Chuyển đổi kiểu tự động: Khi thực hiện một phép toán với 2 toán hạng có kiểu DL khác nhau, máy tính tự chuyển kiểu DL của toán hạng có kích thước nhỏ hơn về kiểu DL có kích thước lớn hơn
- Ví dụ: int + float = float + float = float
- Ép kiểu: Khi thực hiện phép chia với 2 số nguyên cho kết quả là một số nguyên
- Ví dụ: 10/4 = 2
- Muốn kết quả được một số thực ta cần ép kiểu
- Ví dụ: (float)10/4 = 2.5

1.3.4. Các loại toán tử

Toán tử số học

STT	Toán tử	Cách viết	Kết quả
1	Cộng	+	Tổng 2 số nguyên/thực
2	Trừ	-	Hiệu 2 số nguyên/thực
3	Nhân	*	Tích 2 số nguyên/thực
4	Chia	/	Phần nguyên phép chia nếu 2 toán hạng đều có kiểu nguyên
5	Đồng dư	%	Chỉ thực hiện với 2 số nguyên, kết quả là phần dư của phép chia
6	Tăng	++	Làm tăng giá trị của biến nguyên lên 1 đơn vị
7	Giảm	- -	Làm giảm giá trị của biến nguyên đi 1 đơn vị
8	Đổi dấu	-	Trả về giá trị đối của toán hạng

Các loại toán tử (tt)

Toán tử so sánh

Stt	Toán tử	Cách viết	Kết quả
1	Lớn hơn	>	true/false
2	Nhỏ hơn	<	true/false
3	Lớn hơn hoặc bằng	>=	true/false
4	Nhỏ hơn hoặc bằng	<=	true/false
5	Bằng	==	true/false
6	Không bằng	!=	true/false

Các loại toán tử (tt)

Toán tử logic

Stt	Toán tử	Cách viết	Kết quả
1	Và	&&	true/false
2	Hoặc	||	true/false
3	Phủ định	!	true/false

Toán tử gán

Stt	Toán tử	Cách viết
1	Gán	=

1.3.5. Các hàm toán học trong thư viện math.h

STT	Tên hàm	Cách viết
1	Sin(x)	sin(x)
2	Cos(x)	cos(x)
3	$\sqrt{x}$	sqrt(x)
4	e^x	exp(x)
5	Ln(x)	log(x)
6	$Log_{10}(x)$	log10(x)
7	|x| (x nguyên)	abs(x)
8	|x| (x thực)	fabs(x)
9	x^y	pow(x,y)
10	Phần nguyên dưới	floor(x)
11	Phần nguyên trên	ceil(x)

1.4. Nhập/xuất trong C++

Ngoài các hàm nhập xuất chuẩn như trong C: getch(), getche(), getchar(), putchar(), gets(), puts(), scanf(), printf() C++ còn có thêm các lệnh nhập xuất trong thư viện iostream.

1.4.1. Các lệnh Nhập/xuất trong thư viện iostream.h

- Lệnh xuất ra màn hình:

```
cout << <Nội dung>;
```

  - <Nội dung> có thể là chuỗi, biểu thức, biến, hằng số
  - <<: là toán tử xuất

- Ví dụ

```
cout<<"tong hai so la "<<a+b;
```

1.4.1. Các lệnh Nhập/xuất trong thư viện iostream.h

Toán tử định dạng xuất:

```
cout.width(int n) ;
//Định số vị trí tối thiểu cho nội dung được xuất ra màn hình.
cout.fill(char ch);
//Chỉ định ký tự ch được điền vào vị trí trống
cout.precision(int n);
//Chỉ định n vị trí thập phân
```

Ví dụ:

```
cout.width(9);
cout.fill('0');
cout.precision(2);
cout<<a;
```

Nếu a = 123.4523 nó sẽ được in ra màn hình là: 000123.45

Nhập/xuất trong thư viện iostream.h (tt)

Ví dụ: Chương trình tính giá trị biểu thức

```
#include <math.h>
#include <iostream>
using namespace std;
int main()
{
	float x, F;
	cout<<"Nhap so thuc x ";  cin>>x;
	cout.width(8); cout.precision(2);
	cout<<"Gia tri F("<<x<<") =";
	cout<<sin(x)*sin(x)+fabs(x)+exp(log(x));
	return 0;
}
```

Nhập/xuất trong thư viện iostream.h (tt)

- Các hàm định dạng xuất
  - setw(int n) – tương tự như cout.width(int n).
  - setfill(char ch) – tương tự như cout.fill(char ch).
  - setprecision(int n) – tương tự như cout.precision(int n).

  Ví dụ: cout<<setw(9)<<setfill('0')<<setprecision(2)<<a;
- Lệnh nhập dữ liệu: cin>>bien_1>>bien_2>>…>>bien_n;
- >>: được gọi là toán tử nhập

  Ví dụ: cin>>a>>b>>c;

  //dữ liệu nhập cho các biến cách nhau dấu cách.

1.4.2. Các hàm Nhập/xuất ký tự, chuỗi ký tự

- Hàm gets()
- Hàm puts()
- Hàm getch()
- Hàm getche()
- Hàm getchar()
- Hàm putchar()
- Hàm fflush(stdin)

Nhập/xuất dữ liệu trong C++ (tt)

Chương trình nhập/xuất thông tin nhân sự

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main(){
        char HoTen[30]; char Que[50];
        int NgayCong;
        cout<<"Nhap ho ten: "; fflush(stdin); gets(HoTen);
        cout<<"Nhap que quan: "; fflush(stdin); gets(Que);
        cout<<"Nhap ngay cong: "; cin>>NgayCong;
        long Luong = (long) NgayCong*50000;
        cout<<"Thong tin vua nhap la: "<<endl;
        cout<<"Ho ten:"<<HoTen<<endl<<"Que:"<<Que<<endl;
        cout<<"Ngay cong:"<<NgayCong<<endl;
        cout<<"Luong:"<<Luong;
        return 0;
}
```

Thank you!