

# IT001 – NHẬP MÔN LẬP TRÌNH

## CÁC KIỂU DỮ LIỆU CƠ BẢN



1. Cấu trúc một chương trình
2. Bộ từ vựng trong C++
3. Các kiểu dữ liệu cơ sở
4. Biến
5. Hằng
6. Bài tập minh họa

# 1. Cấu trúc chương trình C/C++



# Tiền xử lý

Khai báo thư viện và macro

Khai báo biến, hàm ...

Khai báo hàm được sử dụng trong CT chính

```
void main() {
```

Thân hàm chính

Chương trình chính

Định nghĩa hàm (thân hàm)

Định nghĩa thân hàm đã khai báo

```
#include <iostream>
int main ()
{
    std::cout<<"Xin chào";
    return 0;
}
```



## 2. Bộ từ vựng trong C++

1. Ký tự
2. Từ khóa
3. Tên
4. Hằng ký tự
5. Hằng chuỗi
6. Dấu chấm phẩy
7. Câu chú thích



## 2.1. Ký tự

### Bộ chữ cái 26 ký tự Latin

A, B, C, ..., Z  
a, b, c, ..., z

### Bộ chữ số thập phân

0, 1, 2, ..., 9

### Các ký hiệu toán học

+ - \* / = < > ( )

### Các ký tự đặc biệt

. , : ; [ ] % \ # \$ " ' \_ và khoảng trắng

### Lưu ý:

Khi viết chương trình C++ chỉ sử dụng các ký tự này



## 2.2. Từ khóa

- **Không** thể sử dụng từ khóa để đặt tên cho biến, hàm, tên chương trình con.
- Viết bằng chữ thường

### **Một số từ khóa thông dụng:**

const, enum, signed, struct, typedef, unsigned  
char, double, float, int, long, short, void  
case, default, else, if, switch  
do, for, while  
break, continue, goto, return



## 2.3. Tên/ Định danh (Identifier)

- Một dãy ký tự dùng chỉ **tên** hằng, biến, kiểu dữ liệu, hàm
- Được tạo thành từ các chữ cái, chữ số, dấu gạch nối \_
- Quy ước đặt tên:
  - **Không trùng** với các từ khóa
  - Ký tự đầu tiên là chữ cái hoặc \_
  - Tối đa 255 ký tự
  - Không được sử dụng khoảng trắng ở giữa các ký tự
  - Phân biệt chữ **hoa** chữ **thường**
- Một số ví dụ sai khi đặt tên:

1abc

default

hello world

@mail

case

X-1

x&y

f(x)

ho-ten



## 2. Bộ từ vựng trong C

### 2.4. Hằng ký tự:

'A', 'a'

### 2.5. Hằng chuỗi:

"Xin Chao"

### 2.6. Dấu chấm phẩy:

- Dùng để phân cách các câu lệnh
- Các câu lệnh có thể viết trên 1 dòng, tuy nhiên luôn được phân cách bằng dấu chấm phẩy ;

```
cout<<"Xin chao";return 0;
```





## 2.7. Câu chú thích

- Dùng để mô tả hoặc ghi chú trong source code
- Giúp dễ dàng đọc code sau này
- Có 2 cách để viết chú thích trong C++

**Cách 1:** Viết chú thích trong `/*chú thích*/`

```
/* Đây là câu chú thích 1  
   Đây là câu chú thích 2*/
```

**Cách 2:** Viết chú thích sau `//` chú thích

```
// Đây là câu chú thích 1  
// Đây là câu chú thích 2
```

- Nên sử dụng nhất quán 1 cách. Cách 2 được sử dụng phổ biến hơn
- Khi biên dịch source code thì các phần comments sẽ không được biên dịch



### 3. Các kiểu dữ liệu cơ sở

1. Kiểu số nguyên
2. Kiểu số thực
3. Kiểu luận lý/logic
4. Kiểu void
5. Kiểu kí tự
6. Typedef
7. Enum

## 3.1 Kiểu số nguyên



Kiểu dữ liệu	Kích thước	Phạm vi
short	2 bytes	[-32.768, 32.767]
unsigned short	2 bytes	[0, 65.535]
int	4 bytes	[-2.147.483.648, 2.147.483.647]
unsigned	4 bytes	[0, 4.294.967.295]
long	4 bytes	[-2.147.483.648, 2.147.483.647]
unsigned long	4 bytes	[0, 4.294.967.295]
long long	8 bytes	[-9.223.372.036.854.775.807, 9.223.372.036.854.775.807]
unsigned long long	8 bytes	[0, 18.446.744.073.709.551.615]

**int vs long ???**

OS	arch	size
Windows	IA-32	4 bytes
Windows	Intel 64	4 bytes
Windows	IA-64	4 bytes
Linux	IA-32	4 bytes
Linux	Intel 64	8 bytes
Linux	IA-64	8 bytes
Mac OS X	IA-32	4 bytes
Mac OS X	Intel 64	8 bytes

## 3.1 Kiểu số nguyên



### Chương trình kiểm tra kích thước kiểu dữ liệu

```
#include <iostream>
int main() {
    std::cout << sizeof(char) << " bytes\n";
    std::cout << sizeof(int) << " bytes\n";
    std::cout << sizeof(short) << " bytes\n";
    std::cout << sizeof(long) << " bytes\n";
    return 0;
}
```



## 3.2. Kiểu số thực

Các cách biểu diễn số thực:

### 1. Dạng thập phân:

45.0      -256.45      +122.8      .34      15.

### 2. Dạng khoa học:

$$1.257E + 01 = 1.257 * 10^1 = 12.57$$

$$1257.0E - 02 = 1257 * 10^{-2} = 12.57$$

Kiểu dữ liệu	Kích thước	Phạm vi
float	4 bytes	[3.4E-38, 3.4E+38](~7 chữ số)
double	8 bytes	[1.7E-308, 1.7E+308](~15 chữ số)
long double	8 bytes	[1.7E-308, 1.7E+308](~15 chữ số)

**double vs long double ???**

[https://en.wikipedia.org/wiki/Long\\_double](https://en.wikipedia.org/wiki/Long_double)

## 3.3. Kiểu luận lý/logic



Kiểu dữ liệu	Kích thước	Giá trị
bool	1 bytes	<b>false</b> : giá trị 0 <b>true</b> : giá trị khác 0

```
bool isTrue1 = 1;  
bool isFalse1 = 0;  
bool isTrue2 = true;  
bool isFalse2 = false;
```



## 3.4. Kiểu void

- Kiểu dữ liệu rỗng không chứa gì cả
- Có 2 cách sử dụng:

**Cách 1:** Giá trị trả về cho hàm. Khi không cần giá trị trả về

```
void swap(int &a, int &b)
{
    int c = b;
    b = a;
    a = b;
}
```

**Cách 2:** Một con trỏ chung không trỏ về bất kì giá trị nào

```
int a = 10;
float b = 5;
void *c;
c = &a;
c = &b;
```

## 3.5. Kiểu ký tự



Kiểu dữ liệu	Kích thước	Giá trị
char	1 byte	[-128, 127] hoặc [0, 255]
unsigned char	1 byte	[0, 255]

- Biểu diễn thông qua bảng mã **ASCII**.

- Ví dụ:

Ký tự	ASCII
0, 1, ..., 9	48, 49, ..., 57
A, B, ..., Z	65, 66, ..., 90
a, b, ..., z	97, 98, ..., 122
Enter, ESC, Space	13, 27, 32
", +, -, *, /	34, 43, 45, 42, 47
<, =, >, @	60, 61, 62, 64

<https://en.wikipedia.org/wiki/ASCII>





## 3.5 Kiểu ký tự

- Các ký tự có mã nhỏ hơn thì nhỏ hơn

```
#include <iostream>
int main()
{
    char a = 'a';
    char b = 'b';
    std::cout<<(a<b);
    return 0;
}
```

1

- Phân loại 256 ký tự thành 3 nhóm:

- Ký tự điều khiển: 0 → 31
- Ký tự văn bản: 32 → 126
- Ký tự đồ họa: 127 → 255

## 3.5. Kiểu ký tự



Chương trình xuất mã ASCII cho 1 ký tự nhập từ bàn phím

```
#include <iostream>
int main()
{
    char ascii;
    while(1)
    {
        std::cout<<"Nhập kí tự : ";
        std::cin>>ascii;
        std::cout<<"Mã ASCII là: "<<(int)ascii<<std::endl;
    }
    return 0;
}
```



## 3.6. Typedef

**typedef** dùng để đặt tên mới cho một kiểu dữ liệu có sẵn

Cấu trúc: `typedef kiểu_có_sẵn tên_mới;`

Ví dụ:

```
typedef int songuyen;  
int a;  
songuyen b;
```



## 3.7. Enum

Enum là kiểu dữ liệu giúp hỗ trợ định nghĩa những giá trị liệt kê.

Cấu trúc: `enum` tên\_danh\_sách {danh\_sách\_các\_tên};

Ví dụ:

```
#include <iostream>
enum gioi_tinh
{
    nam = 1,
    nu = 2,
    khac = 3,
};

int main()
{
    gioi_tinh sv_gt = nam;
    std::cout<<sv_gt;
    return 0;
}
```

### Tại sao nên sử dụng enum?

1. Nhất quán
2. Source code rõ ràng
3. Dễ nâng cấp, sửa chữa, bảo trì



## 4. Biến

1. Giới thiệu chung về biến
2. Cú pháp khai báo biến
3. Địa chỉ của biến
4. Biến cục bộ (local variable)
5. Biến toàn cục (global variable)
6. Khởi tạo biến cục bộ và toàn cục



## 4.1. Giới thiệu chung

- Biến là một ô nhớ hoặc 1 vùng nhớ dùng để chứa dữ liệu trong quá trình thực hiện chương trình
- Mỗi biến có một kiểu dữ liệu cụ thể, kích thước của biến phụ thuộc vào kiểu dữ liệu.
- Giá trị của biến có thể được thay đổi
- Quy cách đặt tên biến:
  - **Không trùng** với các từ khóa, hoặc tên hàm.
  - Ký tự đầu tiên là chữ cái hoặc \_
  - Không được sử dụng khoảng trắng ở giữa các ký tự
  - *Nên sử dụng tất cả chữ thường với dấu \_ giữa các từ.*

```
int so_nguyen;  
float so_thuc;
```

<https://google.github.io/styleguide/cppguide.html>



## 4.2. Cú pháp khai báo biến

### - Cú pháp khai báo biến:

- Cách 1: `kiểu_dữ_liệu tên_biến_1, tên_biến_2;`
- Cách 2: `kiểu_dữ_liệu tên_biến_1;`  
`kiểu_dữ_liệu tên_biến_2;`

```
int    i, j, k;  
char   c, ch;  
float  f, salary;  
double d;
```

### - Khai báo và khởi tạo:

`kiểu_dữ_liệu tên_biến_1 = giá_trị;`

```
int    d = 3;  
char   x = 'x';  
float  f = 2.1;
```



## 4.2. Cú pháp khai báo biến

### Ví dụ:

Viết chương trình nhập vào 3 số a, b và c.

Cho biết a, b, c có tạo thành 3 cạnh của tam giác không ?

### Cần khai báo bao nhiêu biến?

```
int a;  
int b;  
int c;
```

```
int a, b, c;
```





## 4.3. Địa chỉ của biến

RAM được tạo nên bởi nhiều ô nhớ.

Mỗi ô nhớ có kích thước 1 byte.

Mỗi ô nhớ có địa chỉ duy nhất và được đánh số từ 0 trở đi.

Mỗi 1 biến khi được khai báo sẽ được cấp 1 vùng nhớ với địa chỉ duy nhất để lưu trữ biến đó.

Để truy cập vào địa chỉ của một biến ta sử dụng câu lệnh. **&**

```
#include <iostream>
int main()
{
    int a = 5;
    std::cout<<"Gia tri cua a: "<<a<<'\\n';
    std::cout<<"Dia chi cua a: "<<&a<<'\\n';
    return 0;
}
```

008FF82C  
a 5

Gia tri cua a: 5  
Dia chi cua a: 008FF82C



## 4.4. Biến cục bộ

- Biến được **định nghĩa** trong một hàm hoặc 1 block được gọi là biến cục bộ
- Biến cục bộ chỉ được sử dụng bên trong hàm hoặc block
- Các hàm bên ngoài khác sẽ không truy cập được biến cục bộ

```
#include <iostream>
int main()
{
    int a = 2, b = 3;
    int c;
    c = a + b;
    std::cout<<c;
    return 0;
}
```

Biến a, b, c là các biến cục bộ bên trong hàm main



## 4.5. Biến toàn cục

- Biến toàn cục được định nghĩa bên ngoài các hàm, và thường được định nghĩa ở phần đầu của source code file.
- Biến toàn cục sẽ giữ giá trị của biến xuyên suốt chương trình.
- Tất cả các hàm đều có thể truy cập biến toàn cục

### Ví dụ 1

```
#include <iostream>
int g;
int main()
{
    int a = 2, b = 3;
    g = a + b;
    std::cout<<g;
    return 0;
}
```

g = ???

### Ví dụ 2

```
#include <iostream>
int g = 20;
int main()
{
    int a = 2, b = 3;
    g = a + b;
    std::cout<<g;
    return 0;
}
```

g = ???



## 4.6. Khởi tạo biến toàn cục và cục bộ

- Khi biến cục bộ được định nghĩa, giá trị của biến sẽ không được khởi tạo.

→ **Ta phải gán giá trị cho biến cục bộ để khởi tạo**

- Biến toàn cục sẽ được **tự động khởi tạo** giá trị

Kiểu dữ liệu	Giá trị khởi tạo
int	0
char	'\0'
float	0
double	0
pointer	NULL



## 5. Hằng (constant)

Hằng đại diện cho một giá trị không đổi trong suốt quá trình thực thi của chương trình.

Không thể gán lại giá trị cho hằng

### Có 4 loại hằng:

1. Hằng số nguyên:  
Có thể viết dưới dạng hệ 10:  $212_{10}$ , hệ 8:  $0213_8$ , hệ 16:  $0x4b_{16}$
2. Hằng số thực:  
 $102.0$ ,  $-223.1$ ,  $1234.56e - 3$  ( $123.56 * 10^{-3}$ )
3. Hằng luận lý (đúng hoặc sai)  
`true` hoặc `false`
4. Hằng ký tự  
`"C:\user\username\local"`



## 5. Hằng

Cách định nghĩa hằng trong C++: Có 2 cách

### Cách 1: Sử dụng **#define**

Câu lệnh: `#define tên_hằng giá_trị`

Lưu ý: Không có ký tự **;**

Ví dụ:

```
#include <iostream>
#define PI 3.14
int main()
{
    int r = 2;
    std::cout<<2*r*PI;
    return 0;
}
```

6.28



## 5. Hằng

Cách định nghĩa hằng trong C++: Có 2 cách

### Cách 2: Sử dụng **const**

Câu lệnh: `const kiểu_giá_trị tên_hằng = giá_trị_hằng;`

Ví dụ:

```
#include <iostream>
int main()
{
    const int PI = 3.14;
    int r = 2;
    std::cout<<r*PI;
    getch();
    return 0;
}
```

6.28



**Bài tập về nhà:**  
**Câu lệnh `#define` và `const`  
khác nhau thế nào?**





## 6. Bài tập minh họa

1. Cho biết năm sinh của một người và tính tuổi của người đó.
2. Cho 2 số  $a, b$ . Tính tổng, hiệu, tích và thương của hai số đó.
3. Cho biết tên sản phẩm, số lượng và đơn giá. Tính tiền và thuế giá trị gia tăng phải trả, biết:
  - a. tiền = số lượng \* đơn giá
  - b. thuế giá trị gia tăng = 10% tiền
4. Cho biết điểm thi và hệ số 3 môn Toán, Lý, Hóa của một sinh viên. Tính điểm trung bình của sinh viên đó.
5. Cho biết bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.



Cho biết năm sinh của một người và tính tuổi của người đó?

```
#include <iostream>
int main()
{
    int namsinh = 1998;
    int tuoi = 0;
    tuoi = 2016-namsinh;
    return 0;
}
```

## Bài tập 2



Cho 2 số a, b. Tính tổng, hiệu, tích và thương của hai số đó

```
#include <iostream>
int main()
{
    int a = 10;
    int b = 3;
    int tong = 0;
    int hieu = 0;
    int tich = 0;
    float thuong = 0;
    tong = a + b;
    hieu = a - b;
    tich = a * b;
    thuong = (float)a / b;
    return 0;
}
```

# Bài tập 3



1. Cho biết tên sản phẩm, số lượng và đơn giá. Tính tiền và thuế giá trị gia tăng phải trả, biết:
  - a. tiền = số lượng \* đơn giá
  - b. thuế giá trị gia tăng = 10% tiền

```
#include <iostream>
int main()
{
    int Soluong = 10;
    int Dongia = 500;
    int Tien = 0;
    float Vat = 0;
    Tien = Soluong * Dongia;
    Vat = Tien * 0.1;
    return 0;
}
```



Cho biết điểm thi và hệ số 3 môn Toán, Lý, Hóa của một sinh viên. Tính điểm trung bình của sinh viên đó.

```
#include <iostream>
int main()
{
    float toan = 6.5;
    float hstoan = 2.0;
    float ly = 7.0;
    float hsly = 1.0;
    float hoa = 7.5;
    float hshoa = 1.0;
    float Dtb = 0;
    Dtb = (toan * hstoan + ly * hsly + hoa * hshoa) / (hstoan + hsly + hshoa);
    return 0;
}
```

# Bài tập 5



Cho biết bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.

```
#include <iostream>
#define PI 3.14
int main()
{
    float r = 6.5;
    float chuvi = 0;
    float dientich = 0;
    chuvi = 2 * PI * r;
    dientich = PI * r * r;
    return 0;
}
```

## 5. Bài tập về nhà



1. Cho số xe (gồm 5 chữ số) của bạn. Cho biết số xe của bạn được mấy nút?
2. Cho 1 ký tự chữ thường. In ra ký tự chữ hoa tương ứng.
3. Cho 3 số nguyên. Cho biết số lớn nhất và nhỏ nhất?
4. Cho số thực  $x$ . Tính giá trị các biểu thức sau:

a)

$$y_1 = 4(x^2 + 10x\sqrt{x} + 3x + 1)$$

b)

$$y_2 = \frac{\sin(\pi x^2) + \sqrt{x^2 + 1}}{e^{2x} + \cos\left(\frac{\pi}{4}x\right)}$$

5. Viết chương trình cho 2 giờ (giờ, phút, giây) và thực hiện cộng, trừ 2 giờ này.