CHƯƠNG 2 : CÁC KIỂU DỮ LIỆU CƠ BẢN TRONG C++

2.1 Các yếu tố cơ bản của ngôn ngữ C++

2.1.1 Ký hiệu cơ sở :

Ngôn ngữ C++ được xây dựng từ bộ ký hiệu cơ sở sau :

-Bộ 26 chữ cái La-Tinh viết thường (nhỏ): a,b,...,z.

-Bộ 26 chữ cái La-Tinh viết hoa (lớn) : A,B,...,Z.

-Bộ 10 chữ số hệ thập phân : 0,1,...,9.

-Bộ dấu các toán tử số học : + - * /

-Bộ dấu các toán tử so sánh : < > =

- Ký tự gạch nối : _ (Khác dấu trừ -).

-Các ký hiệu khác : ' " ; , . : [ ] # $ & { } % ! . . . .

Đặc biệt có khoảng trắng dùng để ngăn cách các từ (phím Space).

Các ký hiệu cơ sở đều có trên bàn phím.

2.1.2. Các từ

a. Từ khóa (Key Word)

b. Tên hoặc danh hiệu (identifier)

a. Từ khóa (Key Word):

Là những từ có ý nghĩa hoàn toàn xác định, chúng thường được dùng để khai báo các kiểu dữ liệu , để viết các toán tử, và các câu lệnh. Sau đây là các từ khóa thường dùng:

- a1. Các từ khóa khai báo kiểu dữ liệu:

  char, int, long, float, double, struct, class, void, unsigned, union, enum, typedef, . . .

- a2. Các từ khóa liên quan đến cấu trúc lệnh:

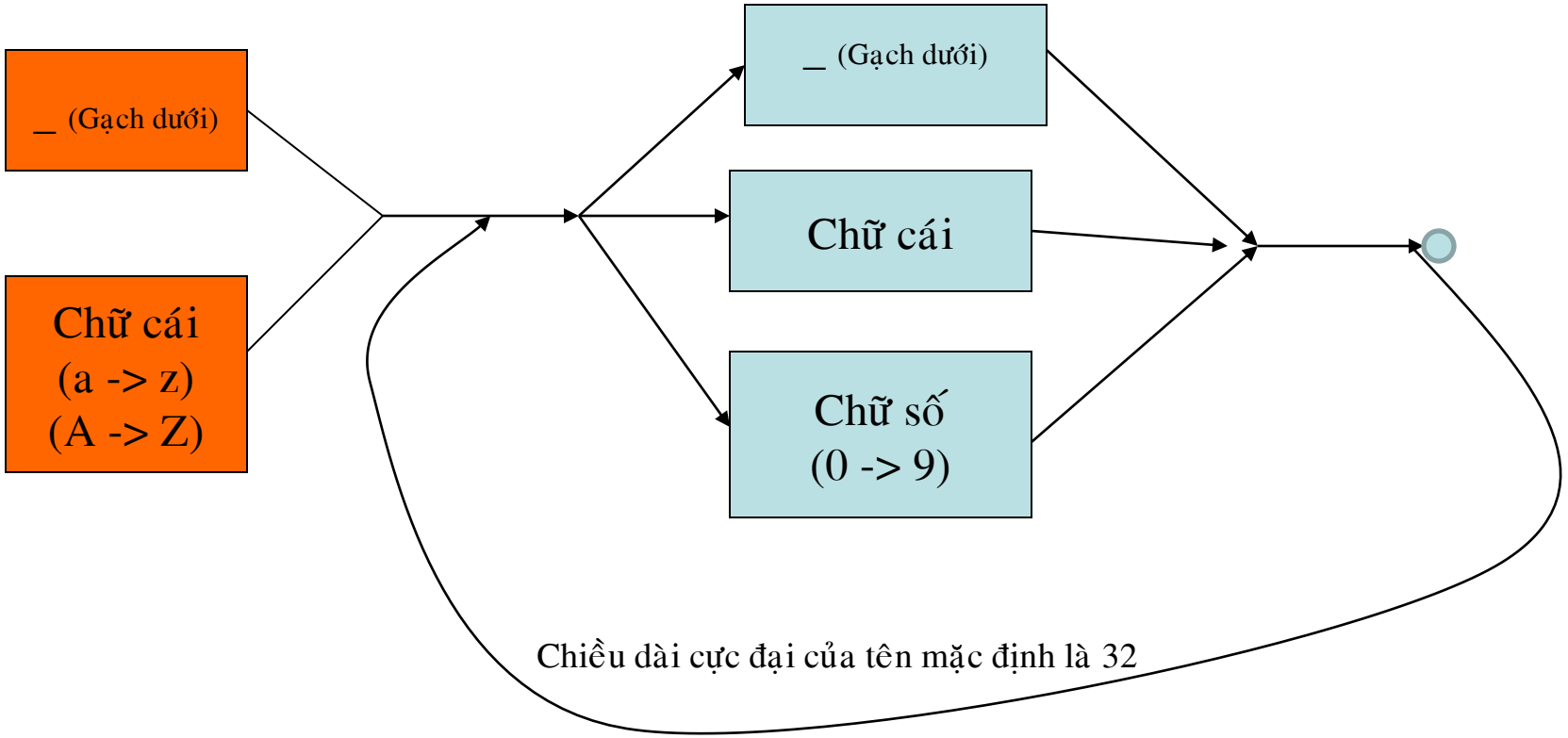
  if, else, switch, case, default, while, for, do..while, break, continue, . . .

- Ghi chú :

  Các từ khóa chỉ được viết thường, chẳng hạn không được viết STRUCT mà phải viết struct.

b. Tên hoặc danh hiệu (identifier)

- Là từ do người sử dụng tự đặt để giải quyết bài toán của mình. Từ tự đặt dùng để đặt tên cho hằng, biến, hàm, tên kiểu dữ liệu mới,...
- Tên được đặt theo quy tắc : phải bắt đầu bằng một chữ cái hoặc dấu gạch nối, sau đó là các chữ cái, chữ số hoặc dấu gạch nối , và không được trùng với từ khóa.
- C/C++ phân biệt chữ thường, chữ hoa.



2.2 Các kiểu dữ liệu cơ bản trong C++

2.2.1. Kiểu ký tự

a. Ký tự 8 bit:

Một giá trị ký tự có kiểu dữ liệu khai báo bằng từ khóa char, được lưu trử trong 8 bit và biểu diễn thông qua bảng mã ASCII .

Chẳng hạn :

Ký tự	Mã ASCII (hệ 10)
0	48
1	49
A	65
a	97
. . . .	

Có các kiểu ký tự 8 bit tương ứng với các từ khóa :

signed char (như char, có dấu)

unsigned char (Không dấu).

Sau đây là bảng kích thước, phạm vi biểu diễn của các kiểu ký tự :

KIỂU	Phạm vi biểu diễn	Kích thước	Số ký tự
char	-128 $\rightarrow$ 127	1 byte	256
signed char	-128 $\rightarrow$ 127	1 byte	256
unsigned char	0 $\rightarrow$ 255	1 byte	256

Ghi chú:

b. Ký tự UNICODE

Các ký tự Unicode trong C/C++ được định nghĩa bởi :

wchar_t

Mỗi ký tự Unicode rộng 16 bit.

c. Kiểu TCHAR :

Dùng chung cho char và wchar_t (cho ký tự 8 bit hay unicode 16 bit).

2.2.2 Kiểu nguyên :

Trong C++ cho phép sử dụng các kiểu số nguyên được khai báo bởi từ khóa int, hoặc đi kèm theo int với các từ khóa long, short, unsigned.

Kích thước và phạm vi biểu diễn của chúng được cho trong bảng sau :

Kiểu	Phạm vi biểu diễn	Kích thước
int	$-2^{31} \rightarrow 2^{31} - 1$	32 bit (chiếm 1 từ của máy)
byte	$-128 \rightarrow 127$	8 bit
short int , short	$-2^{15} \rightarrow 2^{15} - 1$	16 bit
unsigned short int	$0 \rightarrow 2^{16} - 1$	16 bit
long int , long	$-2^{31} \rightarrow 2^{31} - 1$	32 bit
unsigned long int	$0 \rightarrow 2^{32} - 1$	32 bit
unsigned int	$0 \rightarrow 2^{32} - 1$	32 bit (chiếm 1 từ của máy)

2.2.3 Kiểu số thực :

C++ cho phép sử dụng 3 kích thước giá trị thực, tương ứng với 3 từ khóa : float, double và long double.

Kích thước và phạm vi biểu diễn của chúng được cho trong bảng sau :

Kiểu	Ý nghĩa	Phạm vi biểu diễn	Kích thước	Độ chính xác
float	Số thực chính xác đơn	-3.4E+38 → 3.4E+38	32 bit	6 số thập phân
double	Số thực chính xác kép	-1.7E+308 → 1.7E+308,0	64 bit	10 số thập phân
long double		Kích thước 96 bit hoặc 128 bit		

2.3 Các hằng

2.3.1 Định Nghĩa.

2.3.2 Các loại hằng.

2.3.1 Định Nghĩa :

Hằng là các đại lượng mà giá trị của nó không thay đổi trong quá trình tính toán. Ta thường dùng các ký tự hoa để biểu diễn các hằng ký hiệu.

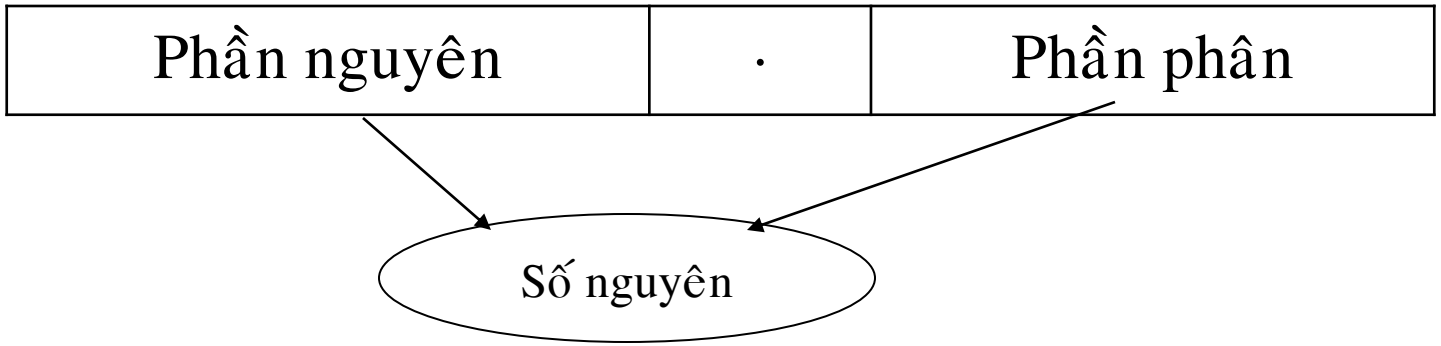
2.3.2 Các loại hằng.

- 2.3.2.1 Hằng số thực
- 2.3.2.2 Hằng nguyên
- 2.3.2.3 Hằng ký tự
- 2.3.2.4 Hằng chuỗi (xâu ký tự)
- 2.3.2.5 Định nghĩa một hằng

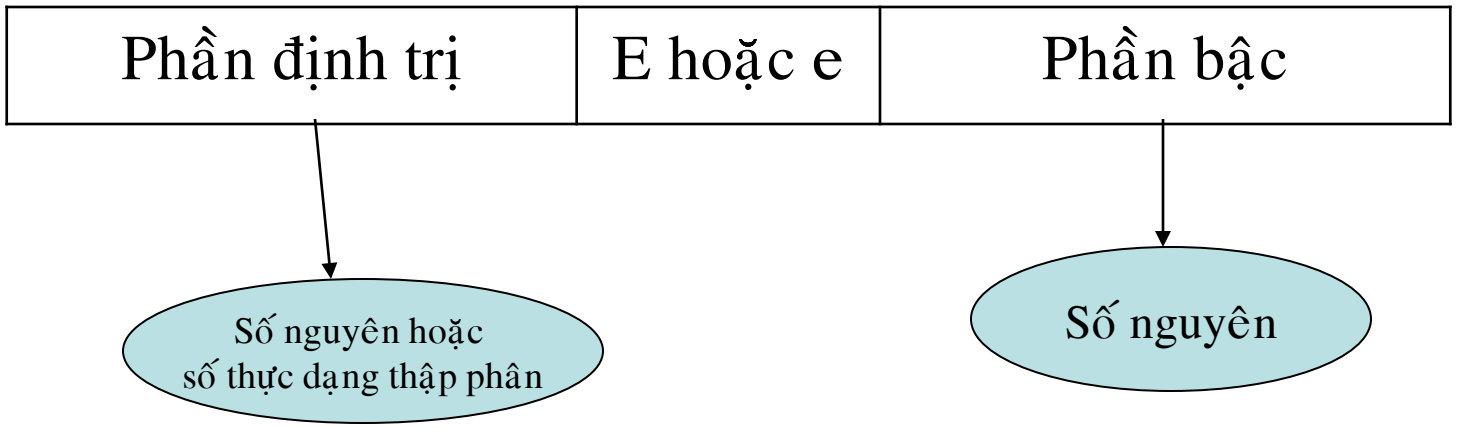
2.3.2.1 Hằng số thực

Giá trị được lấy là float và double. Viết theo 2 cách :

1. Dạng thập phân (dấu chấm cố định) :



2. Dạng khoa học hay dạng mũ (dấu chấm động):



2.3.2.2 Hằng nguyên

1. Hằng int :

   Là số nguyên có kiểu int .

2. Hằng unsigned :

   Biểu diễn : thêm u vào sau hằng int.

3. Hằng int he 8: (c_1, c_2, c_3 thuộc hệ 8)

   $0c_1c_2c_3$

4. Hằng int he 16: (c_1, c_2, c_3 thuộc hệ 16)

   $Xc_1c_2c_3$

2.3.2.3 Hằng ký tự.

a. Đối với ký tự 8 bit (char) :

- Là một ký tự được viết trong 2 dấu nháy đơn.

  Ta có thể viết ‘a’ , ‘A’, ‘3’, ‘+’ . . . Khi đó trình biên dịch của C++ sẽ lưu trử các hằng này bằng cách dùng các mã số ASCII (hệ 10) của nó, tương ứng là 97, 65, 51, . . .
- Ký tự còn có thể biểu diễn theo hệ 8, 16

b) Đối với ký tự unicode 16 bit :

Là một ký tự được viết trong 2 dấu nháy đơn, trước dấu nháy có tiếp đầu ngữ L.

Chẳng hạn , có thế viết : L’A’, hằng ký tự unicode này có giá trị 16 bit là 0x0041.

c. Ghi chú:

Đối với một số hằng ký tự đặc biệt, ta sử dụng cách viết sau (thêm dấu \) :

Cách viết	Ký tự
'\' '	'
'\" '	"
'\\'	\
'\n'	\n (chuyển dòng)
'\0'	\0 (NULL)
'\t'	Tab
'\b'	Backspace
'\r'	CR (Về đầu dòng)
'\f'	LF (sang trang)

2.3.2.4 Hằng chuỗi (xâu ký tự).

Là một dãy ký tự được bao trong 2 dấu nháy kép.

Ví dụ :

"Da Lat"

"" // Xâu rổng

Ghi chú :

- Xâu ký tự được lưu trử trong máy dưới dạng một mảng các ký tự. Trình biên dịch tự động thêm ký tự NULL ('\0')- được xem là dấu hiệu kết thúc xâu- vào cuối mỗi xâu.
- Hằng ký tự 'a' khác xâu ký tự "a".

2.3.2.5 Định nghĩa một hằng

1. Dùng chỉ thị #define (Có thể định nghĩa lại giá trị hằng) :

- Cách viết :

```
#define  TEN-HANG  GIA_TRI_HANG
//Không có dấu ; sau cùng
```

- TEN-HANG là một tên, đặt theo qui định đặt tên. Qui ước các ký tự trong TEN-HANG là các ký tự HOA.
- Tác dụng :

  TEN-HANG sẽ được thay thế bởi GIA_TRI_HANG cho phần còn lại của văn bản chương trình.

Ví dụ :

```
#define MAX 100      // Thay thế MAX bằng 100
```

2. Dùng từ khóa const

- Cú pháp :

```
const  KDL TEN-HANG = GIA_TRI_HANG;
          //  có dấu ; sau cùng
```

- Tác dụng :

TEN-HANG biểu thị bởi GIA_TRI_HANG

Ví dụ :

```
const   double  PI = 3.14159;
```

2.4 Biến

2.4.1 Khái niệm :

Biến là một phần bộ nhớ được đặt tên, được dùng, để giữ một giá trị mà có thể thay đổi trong chương trình.

Vậy biến gắn với tên và kiểu dữ liệu, có giá trị thay đổi trong quá trình tính toán.

- Tên biến là một tên, đặt theo qui ước đặt tên

2.4.2 Định nghĩa (Khai báo) biến

(Mỗi biến phải được định nghĩa trước khi sử dụng.)

- Cú pháp:

  - Trường hợp một biến:

    kdl b;

  - Trường hợp nhiều biến cùng một kiểu, chẳng hạn:

    kdl b1, b2, b3;

  Trong đó :

  – kdl là kiểu dữ liệu nào đó như char, int, double, . . .
  – b, b1, b2, b3,. . . là các từ tự đặt để chỉ tên của các biến.
  – Giữa kdl và tên biến phải cách nhau ít nhất 1 khoảng trắng.
  – Trong phần tên biến, nếu có nhiều biến thì giữa 2 biến phải phân cách bởi dấu phảy (,).

- Tác dụng:

  Định nghĩa biến sẽ cho CT biết tính chất của biến (kiểu gì, kích thước,..) và yêu cầu CT cấp phát vùng nhớ cho biến lưu trử dữ liệu.

- Ghi chú :

  - Biến của kiểu dữ liệu nào thì chỉ có thể nhận các giá trị của kiểu dữ liệu đó.

  - Tên biến có thể có nhiều từ, qui ước ký tự đầu của của các từ kể từ từ thứ hai trở đi sẽ viết HOA, các ký tự còn lại của tên biến đều viêt thường. Chẳng hạn, danhSach, namSinh, bien, ...

2.4.3 Khởi đầu cho các biến.

Nếu trong định nghĩa biến, ngay sau tên biến ta đặt dấu = (phép gán) và một giá trị dữ liệu tương ứng thì đó chính là cách vừa định nghĩa vừa khởi đầu cho 1 biến.

Ví dụ :

```
int a = 20;
double x = 12456.85, n;
float  x, y = 32.1;
```

Ghi chú :

Có thể thực hiện viêc khởi đầu cho biến nhờ lệnh gán.

2.4.4 Lấy địa chỉ cho biến .

Mỗi biến được cấp phát một vùng nhớ gồm một số byte liên tiếp. Số hiệu của byte đầu chính là địa chỉ của biến.

Để nhận địa chỉ biến ta dùng toán tử & với cú pháp :

&bien

2.5. Câu lệnh gán

- Dạng :

  b = bt;

  Trong đó b là biến. bt là một biểu thức (một công thức toán học nào đó). Trước tiên tính biểu thức bt và sau đó gán giá trị tính được cho biến b.

- Ví dụ :

  ```
  float x;
  x = 2.5;
  x = x*x; // khi đó x nhận giá trị là 6.25
  ```

- Ghi chú:

  Có thể có nhiều dấu gán liền nhau :

  ```
  m = n = 2;
  ```

  Có nghĩa là : n lấy giá trị 2, rồi m lấy giá trị của n, nên có giá trị bằng 2.

2.6. Từ khoá typedef

- Từ khóa typefef được dùng để đổi lại tên một kiểu dữ liệu đã có như char, int, float, mảng ... thành một tên mới.Qui ước ký tự đầu của mỗi từ trong tên kiểu dữ liệu mới sẽ viết HOA, các ký tự còn lại viết thường.
- Cách viết là đặt từ khóa typedef vào trước 1 khai báo thông thường :

Ví dụ :

a) typedef int *IntGia*;// Kiểu intgia thực chất là kiểu int
   IntGia x,y; // x,y là biến kiểu ingia (int)

b) typedef double DGia;// DGia thực chất là kiểu double
   DGia a,b; // a,b là biến kiểu dgia

c) Kiểu ký tự Unicode wchar_t định nghĩa trong WCHART như sau:
   typedef unsigned short wchar_t;
   . . .