**Kiến thức buổi 1:   
*1.Cấu trúc chương trình C cơ bản***

***2.Các kiểu dữ liệu cơ bản***

***3.Toán tử gán và phép tính toán thông thường***

*1. Cấu trúc một chương trình C cơ bản*Một chương trình C bao gồm những phần sau đây:

* Các lệnh tiền xử lý
* Các hàm
* Các biến
* Các lệnh và biểu thức
* Các comment

Đầu tiên hãy xem đoạn code đơn giản mà sẽ in ra màn hình hai từ "Hello World":

#include <stdio.h>

int main()

{

/\* Day la chuong trinh C dau tien \*/

printf("Hello, World! \n");

return 0;

}

+Dòng đầu tiên của chương trình *#include <stdio.h>* là lệnh tiền xử lý, nhắc nhở bộ biên dịch C thêm tệp stdio.h trước khi biên dịch.  
Một số thư viện chuẩn:

stdio.h Xuất, nhập với màn hình, file, bàn phím,…

ctype.h Kiểm tra các lớp ký tự (chữ số, chữ cái,…)

string.h Xử lý chuỗi và bộ nhớ

memory.h Cấp phát và quản lý bộ nhớ động

math.h Một số hàm toán học

stdlib.h Chuyển đổi dữ liệu số-chuỗi, cấp phát bộ nhớ,…

time.h Các hàm về thời gian

+Dòng tiếp theo *int main()* là hàm main, nơi chương trình bắt đầu.

+Dòng tiếp theo /\*...\*/ hoặc // là dòng comment được bỏ qua bởi bộ biên dịch compiler và được dùng để thêm các chú thích cho chương trình. Đây được gọi là phần comment của chương trình.

+Dòng tiếp theo *printf(...)* là một hàm chức năng khác của ngôn ngữ C , in ra thông điệp "Hello, World!" hiển thị trên màn hình.

+Dòng tiếp theo **return 0;** kết thúc hàm chính và trả về giá trị 0.

*\*\* Nhập xuất cơ bản*

Thư viện nhập xuất chuẩn trong C là <stdio.h>, do đó các lập trình viên cần khai báo thư viện này trước khi thực hiện việc nhập xuất dữ liệu bằng chỉ thị #include.

<stdio.h> cung cấp cho chúng ta hai hàm nhập và xuất dữ liệu có định dạng:

* scanf(): Hàm nhập

cú pháp: scanf(<định dạng kỹ thuật>, <danh sách tham chiếu>);

ví dụ: int a;

scanf(“%d”, &a);

%d: đặc tả kiểu dữ liệu đã khai báo với biến a

&a: con trỏ trỏ tới biến a

* printf(): Hàm xuất

cú pháp: printf(<"chuỗi định dạng">, <danh sách tham số>)

ví dụ: printf(“%d”, a);

*2. Các kiểu dữ liệu cơ bản:*

| **Loại dữ liệu** | **Kiểu dữ liệu** | **Đặc tả** | **Số ô nhớ** | **Miền giá trị** |
| --- | --- | --- | --- | --- |
| Boolean | bool |  | 1 byte | 0 hoặc 1. Trong đó 0 => FALSE và 1 => TRUE |
| Ký tự | char | %c | 1 byte | -128 ... 127 hoặc 0 ... 255 |
| Số nguyên | int | %d | 4 byte | -2147483648 ... 2147483647 |
| - | long | %ld | 4 byte | -215 ... 215 - 1 |
| Số thực | float | %f | 4 byte | ±10-37 ... ±10+38 |
| - | double | %lf | 8 byte | ±10-307 ... ±10+308 |

**Một số kí tự điều khiển:**

* \n : Xuống dòng
* \t : Tab ngang (tạo khoảng trắng giống như khi bạn ấn phím Tab trên bàn phím trong soạn thảo văn bản)
* \r : Nhảy về đầu hàng
* \a : Kêu Bip
* \\ : In ra dấu \
* \” : In ra dấu “
* \’ : In ra dấu ‘
* %% : In ra dấu %
* v.v.v

### Biến, hằng:

Biến, hằng giống như những chiếc hộp có tên riêng mà chúng ta có thể để dữ liệu vào và mang ra mỗi khi chúng ta cần sử dụng. Biến khác hằng ở chỗ giá trị lưu trữ trong biến có thể thay đổi trong quá trình thực hiện chương trình; còn giá thị của hằng thì được định nghĩa ngay từ đầu chương trình, không thay đổi trong suốt quá trình thực hiện chương trình.

#### Các quy tắc đặt tên biến, hằng:

Tên biến, tên hằng:

* **Chỉ** chứa các chữ cái, chữ số và kí tự gạch dưới ( \_ ) trong [**bảng mã ASCII**](https://vi.wikipedia.org/wiki/ASCII).
* **Phải** bắt đầu bằng một chữ cái hoặc kí tự gạch dưới.
* **Không** được trùng với các từ khóa trong ngôn ngữ C.
* **Phân biệt hoa thường.** C là ngôn ngữ phân biệt hoa thường, do vậy biến **aBc** và **abc** là khác nhau trong C.

Danh sách các từ khóa trong ngôn ngữ C:

|  |  |  |  |
| --- | --- | --- | --- |
| break | enum | register | typedef |
| case | extern | return | union |
| char | float | short | unsigned |
| const | for | signed | void |
| continue | goto | sizeof | volatile |
| default | if | static | while |
| do | int | struct | \_Packed |
| double |  |  |  |

**Ví dụ:**

* **Tên đúng:**abc, Abc, abc123, \_abc
* **Tên sai:** 1abc (tên sai do bắt đầu bằng số), b@c (tên sai do chứa kí tự @ không được phép), case (tên sai do trùng với từ khóa).

#### Khai báo biến:

Vị trí khai báo: Biến thường được khai báo ở đầu chương trình, đầu hàm hoặc khối lệnh.

Có 2 loại biến theo vị trí khai báo:

* Biến toàn cục: biến khai báo ở ngoài các hàm, các khối lệnh. Chu trình sống của chúng từ lúc bắt đầu chương trình tới khi kết thúc chương trình.
* Biến cục bộ: biến khai báo bên trong các hàm, các khối lệnh. Chu trình sống từ khi bắt đầu khối lệnh tới khi khối lệnh được thực hiện xong.

#### Cú pháp khai báo biến: <kiểu dữ liệu><danh sách biến>;

#### 

Ví dụ chương trình sau:

#include<sidio.h>

int main()

{

int a; // Khai báo biến a với kiễu dữ liệu là kiểu nguyên  
 scanf(“%d”, &a); // sử dụng hàm scanf nhập a   
 // %d: đặc tả của kiểu int đã khai báo với biến a

printf(“%d”, a); // sử dụng hàm prinf xuất a ra màn hình

return 0;

}

#### Khai báo hằng:

Khai báo hằng thực hiện ở đầu chương trình.

**Cách 1: Sử dụng bộ tiền xử lí:**

**#define <tên hằng> <giá trị của hằng>;**

**Cách 2:**

**const** **<**kiểu dữ liệu**>** **<**tên hằng**>** **=** **<**giá trị của hằng**>**;

*3.Toán tử gán và phép tính toán thông thường*

*a. Toán tử là gì?*

Toán tử là một biểu tượng mà nói với trình biên dịch thực hiện một phép toán học nhất định hoặc thao tác hợp lý.

*b. Toán hạng là gì?*

Là các biến tham gia trong toán tử

Ví dụ: a+b

a,b là toán hạng

‘+’ là toán tử

*c. Các loại toán tử*

+Toán tử số học

+Toán tử gán

\*\*Buổi 2 sẽ được học: {

+Toán tử quan hệ

+Toán tử logic

+Toán tử điều kiện }

1. Toán tử số học

Cho biến a=40, b=20, ta có các toán tử số học sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Toán tử | Giải thích | Code | Kết quả |
| + | cộng 2 toán hạng | a+b | 60 |
| - | trừ 2 toán hạng | a-b | 20 |
| \* | nhân 2 toán hạng | a\*b | 800 |
| / | chia 2 toán hạng | a/b | 2 |
| % | chia lấy dư 2 toán hạng | a%b | 0 |
| ++ | tăng toán hạng lên 1 đơn vị | a++ hoặc ++a | 41 |
| -- | giảm toán hạng đi 1 đơn vị | a—hoặc --a | 39 |

Ví dụ:

a=15, b=10, in ra a/b, ++a, b++

Kết quả 1 16 10

Note:

-Vì cùng kiểu số nguyên nên 15/10=1, muốn cho giá trị đúng là 1.5 thì phải ép kiểu

Muốn lấy x chữ số thập phân sau dấu phây thì dùng lệnh %.xf

-Nếu không thực hiện phép toán thì ++a giống a++

Nếu thực hiện phép toán thì có sự khác nhau:

++a: tăng giá trị a lên 1 đơn vị rồi mới thao tác toán tử với a

a++: thao tác toán tử với a rồi mới tăng giá trị a lên 1 đơn vị

--a và a— tương tự

1. Toán tử gán mở rộng

|  |  |  |
| --- | --- | --- |
| Toán tử | Code | Giải thích |
| += | a+=b | a=a+b |
| -= | a-=b | a=a-b |
| \*= | a\*=b | a=a\*b |
| /= | a/=b | a=a/b |
| %= | a%=b | a=a%b |

%c : Ký tự đơn

%s : Chuỗi

%d : Số nguyên hệ 10 có dấu

%f : Số chấm động (VD 5.54 khi in sẽ ra 5.540000)

%e : Số chấm động (ký hiệu có số mũ)

%g : Số chấm động (VD 5.54 khi in sẽ in ra 5.54)

%x : Số nguyên hex không dấu (hệ 16)

%o : Số nguyên bát phân không dấu (hệ 8)

l : Tiền tố dùng kèm với %d, %x, %o để chỉ số nguyên dài (ví dụ%ld)

%5c : Xuất ký tự có bề rộng 5

%5d : Số nguyên có bề rộng 5

%20s : Xuất chuỗi có bề rộng 20

%5.3f : Xuất số thực có bề rộng 5 trong đó có 3 số sau dấu phẩy

%-5d : Số nguyên có bề rộng 5 nhưng căn lề trái