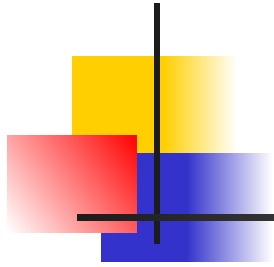




Biến và Kiểu Dữ Liệu

Chương 2



Mục Tiêu

- Hiểu được biến (variables)
- Phân biệt biến và hằng (constants)
- Liệt kê các kiểu dữ liệu khác nhau và sử dụng chúng trong chương trình C
- Hiểu và sử dụng các toán tử số học



Dữ liệu **15**

Bộ nhớ			
	15		
	Dữ liệu trong bộ nhớ		

Mỗi vị trí trong bộ nhớ là duy nhất

Biến cho phép cung cấp một tên có ý nghĩa cho mỗi vị trí nhớ



Ví dụ

```
BEGIN  
  DISPLAY 'Enter 2 numbers'  
INPUT A, B  
C = A + B  
  DISPLAY C  
END
```

- A, B và C là các biến trong đoạn mã giả trên
- Tên biến giúp chúng ta truy cập vào bộ nhớ mà không cần dùng địa chỉ của chúng
- Hệ điều hành đảm nhiệm việc cấp bộ nhớ còn trống cho những biến này
- Để tham chiếu đến một giá trị cụ thể trong bộ nhớ, chúng ta chỉ cần dùng tên của biến



Hằng

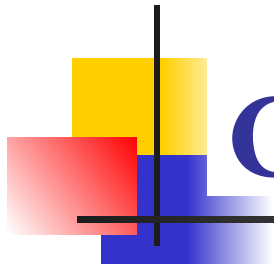
- Một **hằng** (constant) là một giá trị không bao giờ thay đổi
- Các ví dụ
 - 5 số / hằng số nguyên
 - 5.3 số / hằng số thực
 - 'Black' Hằng chuỗi
 - 'C' Hằng ký tự
- **Biến lưu giữ các giá trị hằng**



Định danh

- Tên của các biến (variables), các hàm (functions), các nhãn (labels) và các đối tượng khác nhau do người dùng định nghĩa gọi là định danh
- Ví dụ về các định danh đúng
 - arena
 - s_count
 - marks40
 - class_one
- Ví dụ về các định danh sai
 - 1sttest
 - oh!god
 - start... end
- Các định danh có thể có bất cứ chiều dài nào theo quy ước, nhưng số ký tự trong một biến được nhận diện bởi trình biên dịch thì thay đổi theo trình biên dịch
- Các định danh trong C có phân biệt chữ hoa và chữ thường

Không hợp lệ !



Các nguyên tắc đặt tên định danh

Tên biến phải bắt đầu bằng một ký tự alphabet

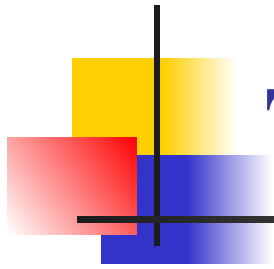
Theo sau ký tự đầu có thể là các ký tự chữ, số ...

Nên tránh đặt tên biến trùng tên các từ khoá

Tên biến nên mô tả được ý nghĩa của nó

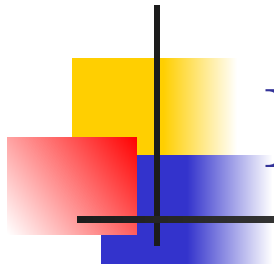
Tránh dùng các ký tự gây nhầm lẫn

Nên áp dụng các quy ước đặt tên biến chuẩn khi lập trình



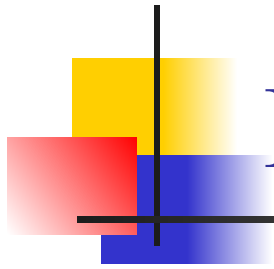
Từ khóa

- Từ khóa: Tất cả các ngôn ngữ dành một số từ nhất định cho mục đích riêng
- Những từ này có một ý nghĩa đặc biệt trong ngữ cảnh của một ngôn ngữ cụ thể
- Sẽ không có xung đột nếu từ khóa và tên biến khác nhau. Ví dụ từ *integer* cho tên biến thì hoàn toàn hợp lệ ngay cả khi mà từ khóa là `int`



Kiểu dữ liệu

- Các kiểu dữ liệu khác nhau được lưu trữ trong biến là:
 - Số (Numbers)
 - Số nguyên.
Ví dụ : 10 hay 178993455
 - Số thực.
Ví dụ, 15.22 hay 15463452.25
 - Số dương
 - Số âm
 - Tên. Ví dụ : John
 - Giá trị luận lý :
Ví dụ : Y hay N



Kiểu dữ liệu (tt.)

- Kiểu dữ liệu mô tả loại dữ liệu sẽ được lưu trong biến
- Tên biến đặt sau kiểu dữ liệu
- Ví dụ : tên biến “varName” đứng sau kiểu dữ liệu “int”

kiểu dữ liệu

tên biến

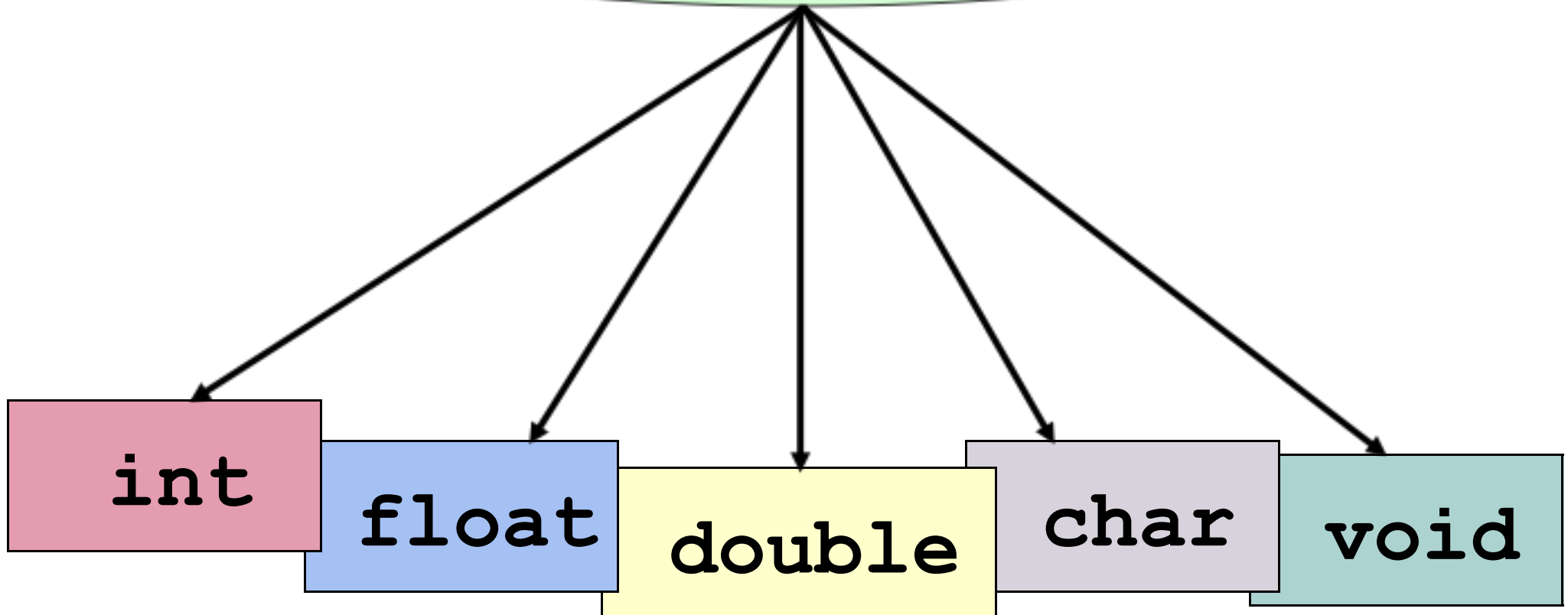
int

varName



Kiểu dữ liệu cơ bản

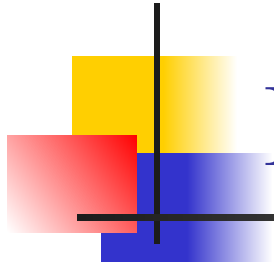
Kiểu dữ liệu cơ bản





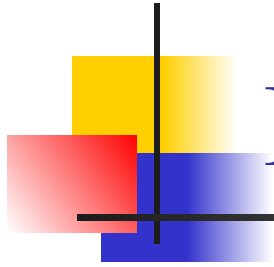
Kiểu số nguyên (int)

- Lưu trữ dữ liệu số
int num;
- Không thể lưu trữ bất cứ kiểu dữ liệu nào khác như “Alan” hoặc “abc”
- Chiếm 16 bits (2 bytes) bộ nhớ
- Biểu diễn các số nguyên trong phạm vi -32768 tới 32767
- Ví dụ : 12322, 0, -232



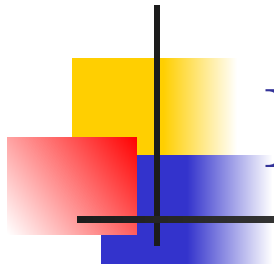
Kiểu số thực (float)

- Lưu trữ dữ liệu số chứa phần thập phân
float num;
- Có độ chính xác tới 6 con số
- Chiếm 32 bits (4 bytes) bộ nhớ
- Ví dụ : 23.05, 56.5, 32



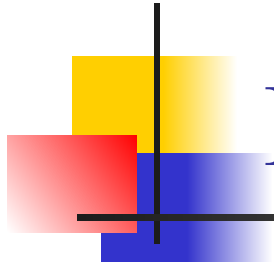
Kiểu số thực (double)

- Lưu trữ dữ liệu số chứa phần thập phân
double num;
- Có độ chính xác tới 10 con số
- Chiếm 64 bits (8 bytes) bộ nhớ
- Ví dụ : 23.05, 56.5, 32



Kiểu ký tự (char)

- Lưu trữ một ký tự đơn
char gender;
gender='M';
- Chiếm 8 bits (1 byte) bộ nhớ
- Ví dụ: 'a', 'm', '\$' '%', '1', '5'



Kiểu void

- Không lưu bất cứ dữ liệu gì
- Báo cho trình biên dịch không có giá trị trả

về



Những kiểu dữ liệu dẫn xuất

Bộ bổ từ (Modifiers)
kiểu dữ liệu

+

Kiểu dữ liệu
cơ bản

=

Kiểu dữ liệu dẫn xuất

unsigned
d

+

int

=

unsigned int
(chỉ là số dương)

short
t

+

int

=

short int
(chiếm ít bộ nhớ hơn int)

long
g

+

int/double
e

=

Long int /longdouble
(chiếm nhiều bộ nhớ hơn
int/double)



Các kiểu dữ liệu signed và unsigned

- Kiểu unsigned chỉ rõ rằng một biến chỉ có thể nhận giá trị dương
 unsigned int varNum;
 varNum=23123;
- varNum được cấp phát 2 bytes
- Bỏ từ unsigned có thể được dùng với kiểu dữ liệu int và float
- Kiểu unsigned int hỗ trợ dữ liệu trong phạm vi từ 0 đến 65535



Những kiểu dữ liệu long (dài) và short (ngắn)

- **short int** chiếm giữ 8 bits (1 byte)
 - Cho phép số có phạm vi từ -128 tới 127
- **long int** chiếm giữ 32 bits (4 bytes)
 - -2,147,483,648 và 2,147,483,647
- **long double** chiếm 128 bits (16 bytes)



Kiểu dữ liệu & phạm vi giá trị

Kiểu	Dung lượng tính bằng bit	Phạm vi
char	8	-128 tới 127
Unsigned char	8	0 tới 255
signed char	8	-128 tới 127
int	16	-32,768 tới 32,767
unsigned int	16	0 tới 65,535
signed int	16	Giống như kiểu int
short int	16	Giống như kiểu int
unsigned short int	16	0 tới 65, 535



Kiểu dữ liệu & phạm vi giá trị (tt.)

Kiểu	Dung lượng tính bằng bit	Phạm vi
signed short int	16	Giống như kiểu short int
long int	32	-2,147,483,648 tới 2,147,483,647
signed long int	32	0 tới 4,294,967,295
unsigned long int	32	Giống như kiểu long int
float	32	6 con số thập phân
double	64	10 con số thập phân
long double	128	10 con số thập phân



Ví dụ về cách khai báo biến

```
main ()
```

```
{
```

```
    char abc;    /*abc of type character */
```

```
    int xyz;      /*xyz of type integer */
```

```
    float length; /*length of type float */
```

```
    double area;  /*area of type double */
```

```
    long liteyrs; /*liteyrs of type long int */
```

```
    short arm;    /*arm of type short integer*/
```

```
}
```