

Ngôn ngữ lập trình C

B4: Khai báo Biến và Hằng số



PHENIKAA
UNIVERSITY

Khoa Công nghệ thông tin

CHỦ ĐỀ

- **Khai báo**

- Biến là gì?

- Các kiểu dữ liệu cơ bản

- Hằng số

- **Bài tập thực hành**

BIẾN LÀ GÌ?

- **Biến (variable):**

- Là đại lượng mà giá trị có thể thay đổi trong chương trình.
- Hằng và biến được sử dụng để lưu trữ dữ liệu, và phải thuộc một kiểu dữ liệu nào đó.
- Tên biến và hằng được đặt theo quy tắc đặt tên cho định danh.

Nhắc lại quy tắc đặt tên

- **Định danh / tên (identifier):**
 - Là một dãy các kí tự dùng để gọi tên các đối tượng trong chương trình.
 - Các đối tượng trong chương trình gồm có biến, hằng, hàm, kiểu dữ liệu, ...
 - Có thể được đặt tên:
 - *Bởi ngôn ngữ lập trình (đó chính là các từ khóa)*
 - *Hoặc do người lập trình đặt.*

Nhắc lại quy tắc đặt tên

– Quy tắc đặt tên:

- *Chỉ được gồm có: chữ cái, chữ số và dấu gạch dưới “_” (underscore).*
- *Bắt đầu của định danh phải là chữ cái hoặc dấu gạch dưới, không được bắt đầu định danh bằng chữ số.*
- *Định danh do người lập trình đặt không được trùng với từ khóa.*
- *Phân biệt chữ hoa chữ thường.*

Nhắc lại quy tắc đặt tên

- Ví dụ định danh/tên hợp lệ:
i, x, y, a, b, _function, _MY_CONSTANT, PI, gia_tri_1
- Ví dụ về định danh/tên không hợp lệ:

1_a, 3d, 55x	bắt đầu bằng chữ số
so luong, ti le	có kí tự không hợp lệ (dấu cách – <i>space</i>) trong tên
int, char	trùng với từ khóa của ngôn ngữ C

Nhắc lại quy tắc đặt tên

- Cách thức đặt định danh/tên:
 - Hằng số: chữ hoa
 - Các biến, hàm hay cấu trúc: Bằng chữ thường.
 - Nếu tên gồm nhiều từ thì ta nên phân cách các từ bằng dấu gạch dưới.
- Ví dụ:

Định danh	Loại đối tượng
HANG_SO_1, _CONSTANT_2	hằng
a, b, i, j, count	biến
nhap_du_lieu, tim_kiem, xu_li	hàm
sinh_vien, mat_hang	cấu trúc

Đặc điểm của Biến

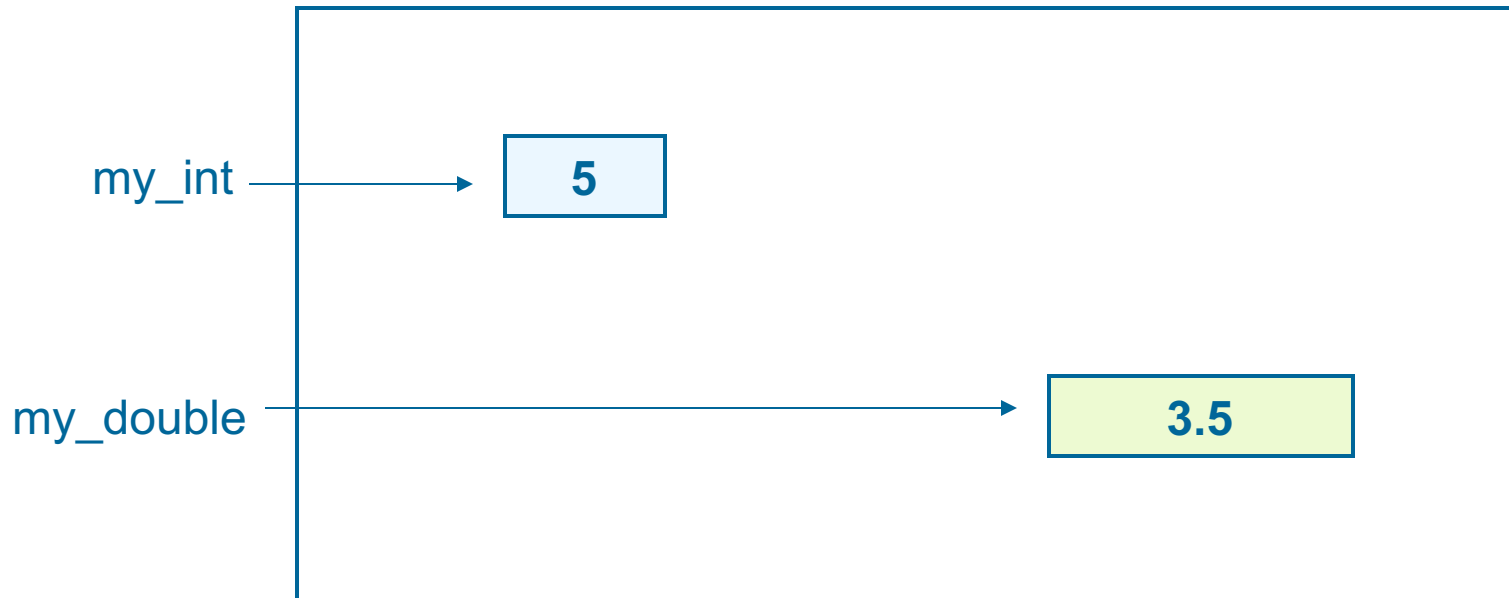
- Tên của một vùng không gian lưu trữ trên bộ nhớ:

```
int nRow = 0;
```

- Kiểu biến: kích thước và ý nghĩa của đối tượng lưu trữ. Ví dụ: kiểu `int` là đối tượng số nguyên sẽ chiếm 4 byte trên bộ nhớ.
- Phạm vi của biến:
 - Cục bộ/Địa phương: trong một khối chương trình con;
 - Toàn cục: trong toàn bộ chương trình;
 - Đối số của một hàm số;
- Thời gian sống của biến:
 - Tự động sinh ra và giải phóng trong từng khối chương trình con;
 - Toàn cục: tồn tại đến hết chương trình;

Biến được lưu trữ trên bộ nhớ

```
int my_int = 5;  
double my_double = 3.5;
```



Khai báo, Định nghĩa và Khởi tạo

- Khai báo biến là đặt chỗ một không gian lưu trữ hay còn gọi là định nghĩa một biến. Ví dụ:

```
int j;
```

- Trong định nghĩa biến còn có thể thiết lập một giá trị ban đầu cho biến hay còn được gọi là khởi tạo giá trị. Ví dụ:

```
int j = 0;
```

- Khai báo dạng chỉ định ngụ ý mà không đặt chỗ trong bộ nhớ lưu trữ (dùng từ khóa *extern*). Ví dụ:

```
extern int j;
```

- Sử dụng biến. Ví dụ:

```
printf("%d + %d = %d\n", a, b, c);
```

Ví dụ khai báo biến

- `int i;`
- `char c;`
- `float f1, f2;`
- `float f1=7.0, f2 = 5.2;`
- `unsigned int ui = 0;`

Ví dụ 4.1

```
1. #include <stdio.h>
2.
3. void main()
4. {
5.     int a, b, c;
6.     printf("The first number: ");
7.     scanf("%d", &a);
8.     printf("The second number: ");
9.     scanf("%d", &b);
10.    c = a + b;
11.    printf("%d + %d = %d\n", a, b, c);
12.}
```

12	c
7	b
5	a

The first number: 5

The second number:7

5 + 7 = 12

Cú pháp khai báo biến

- <kiểu dữ liệu> <danh sách biến>;

- Ví dụ:

```
int i;  
float x, y, z;  
char c;
```

- Gán giá trị cho biến

```
<tên biến> = <giá trị>;
```

ví dụ:

```
i = 4;  
x = 5.4;  
y = z = 1.2;
```

Hằng số

- **Hằng số:** có giá trị không thay đổi trong suốt chương trình.

- Cú pháp khai báo hằng số:

- Cách 1: sử dụng bộ tiền xử lý

```
#define <tên hằng> <giá trị của hằng>
```

- Cách 2:

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

- Ví dụ:

```
#define TRUE 1
```

```
#define FALSE 0
```

```
const float PI = 3.14;
```

Các kiểu dữ liệu cơ bản

- Kích thước và giới hạn (có thể thay đổi tùy từng hệ máy; CUNIX trong bảng dưới đây):

Kiểu	Kích thước biến (bits)	Khoảng giá trị
char	8	-128...127
short	16	-32,768...32,767
int	32	-2,147,483,648...2,147,483,647
long	32	-2,147,483,648...2,147,483,647
float	32	10^{-38} ... 3×10^{38}
double	64	2×10^{-308} ... 10^{308}

- float có 6 bits dành cho độ chính xác
- double có 15 bits dành cho độ chính xác

Các kiểu dữ liệu cơ bản

- Đối với dữ liệu kiểu unsigned:

<i>Kiểu</i>	<i>Kích thước biến (bits)</i>	<i>Khoảng giá trị</i>
unsigned char	8	0...255
unsigned short	16	0...65,535
unsigned int	32	0...4,294,967,295
unsigned long	32	0...4,294,967,295

- Xem file `/usr/include/limits.h`

Ví dụ 4.2

```
int d,m,y,x;  
char ch1,ch2;  
float f;  
scanf("%d", &x);  
  
scanf("%2d%2d%4d", &d,&m,&y);  
  
scanf("%d/%d/%d", &d,&m,&y);  
  
scanf("%c%c", &ch1,&ch2);  
  
scanf("%f", &f);
```

Kết quả

```
4  
// x=4  
22062007  
// d=22, m=6, y=2007  
22/06/2007  
// d=22, m=6, y=2007  
Ab  
// ch1='A', ch2='b'  
2.3  
// f=2.300000
```

Ví dụ 4.3

- Đọc ký tự và chuỗi

```
1 #include <stdio.h>
2
3 void main()
4 {
5     char x, y[9];
6
7     printf( "Enter a string: " );
8     scanf( "%c%s", &x, y );
9
10    printf( "The input was:\n" );
11    printf( "the character \"%c\" ", x );
12    printf( "and the string \"%s\"\n", y );
13
14 }
```

```
Enter a string: Sunday
The input was:
the character "S" and the string "unday"
```

BÀI TẬP 4.1

- Tìm và giải thích lỗi của chương trình sau, sau đó thực hiện sửa lỗi để cho chương trình chạy được và in ra chữ *Nice*.

```
1  #include<stdio.h>
2  int main()
3  {
4      int j = 4.5;
5      unsigned float i = 1.23;
6      double j = 6.7;
7      k = 10;
8      printf("Nice\n");
9      return 5;
10 }
```

Ví dụ 4.4

Nhập chương trình dưới đây, sau đó đọc và phân tích về các ký tự sử dụng trong chương trình.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int month1, day1, year1, month2, day2, year2;
6
7     printf( "Enter a date in the form mm-dd-yyyy: " );
8     scanf( "%d%c%d%c%d", &month1, &day1, &year1 );
9     printf( "month = %d  day = %d  year = %d\n\n",
10         month1, day1, year1 );
10    printf( "Enter a date in the form mm/dd/yyyy: " );
14    scanf( "%d%c%d%c%d", &month2, &day2, &year2 );
15    printf( "month = %d  day = %d  year = %d\n",
16        month2, day2, year2 );
17
18    return 0;
19 }
```

```
Enter a date in the form mm-dd-yyyy: 11-18-2000
month = 11  day = 18  year = 2000
```

```
Enter a date in the form mm/dd/yyyy: 11/18/2000
month = 11  day = 18  year = 2000
```

BÀI TẬP 4.2

- Tìm, giải thích và sửa lỗi cho chương trình sau:

```
1  #include<stdio.h>
2  #define NUM 30;
3  int main()
4  {
5      const int a;
6      int i = f;
7      float f = a = 10;
8      NUM = i+2;
9      printf("i = %d, f = %.2f", i, f);
10     return 0;
11 }
```

BÀI TẬP 4.3

- Viết chương trình tính chu vi và diện tích hình chữ nhật khi biết độ dài 2 cạnh.

BÀI TẬP 4.4

- Viết chương trình thanh toán tiền mua sách. Người dùng nhập vào: tên sách (title), giá sách (price), số lượng mua (quantity), Mã sách (ISBN) và tính ra số tiền phải trả, bao gồm VAT cố định 5%. Hiện thị ra bảng như sau:

BK Bookseller				
Qty	ISBN	Title	Price	Total
<hr/>				
VAT				
You pay:				

BÀI TẬP 4.5

- Viết chương trình gán giá trị dương có hai chữ số cho một biến số nguyên và hiển thị tổng các chữ số của nó. Ví dụ: nếu giá trị được gán là 35 thì chương trình sẽ hiển thị 8. Không sử dụng toán tử chia lấy dư %.

BÀI TẬP 4.6

- Viết chương trình nhập các dữ liệu cần thiết để có thể in ra nhật ký sử dụng phòng thực hành của giảng viên Việt Nhật như sau
- STT Ngày Tháng Tên lớp Tên môn học Bắt đầu Kết thúc
- 23 10/03/2011 5C C Programming 8:30 11:15