

HVIT CLAN JSC

C Programming

D1.P2 : Thao tác với các kiểu dữ liệu

Nội dung slide: Thao tác với các kiểu dữ liệu

Thuyết minh: Nguyễn Đồng Khánh

HVIT CLAN 2020



NỘI DUNG CHÍNH (7 nội dung)



**Các kiểu dữ liệu
cơ bản**



Khai báo biến



Phép gán dữ liệu



**Các phép toán cơ
bản**



**Cách Debug trên
Visual Studio**



**Một ví dụ phép
gán cơ bản hay**



**Thư viện
tiêu chuẩn**

Phần 1

Các kiểu dữ liệu cơ bản



A. Kiểu dữ liệu trong C

Kiểu dữ liệu **cơ bản** trong C: được hình dung như **một tập hợp các giá trị**

Càng đi sâu vào lập trình, hiểu biết về kiểu dữ liệu càng sâu sắc

Kiểu số nguyên



short , **int**, *long int*, *unsigned short*, *unsigned int*...

Kiểu số thực



float, **double**, *long double*

Kiểu ký tự



char, *unsigned char*

B. Kiểu số nguyên int

Kiểu int trong C: là kiểu số nguyên, có giá trị 4byte

4byte thì sao? Nghĩa là 32 bit, mà mỗi 1 bit thì biểu diễn được 2 số phân biệt 0,1.
N bit thì biểu diễn được 2^N số phân biệt, cụ thể ở đây, sẽ có 2^{32} số phân biệt thể hiện bởi kiểu int



Nếu một số không nằm trong đoạn giá trị và cho vào tập này thì sao? Ngôn ngữ C sẽ quản lý theo kiểu quay vòng giá trị.

Ví dụ giá trị **2.147.483.649** vượt ra ngoài 2 giá trị, vậy nếu cho vào tập int, sẽ quay vòng về đầu 2 giá trị, trở thành số **-2.147.483.647**

C. Kiểu số thực double

Kiểu double trong C: là kiểu số thực, có giá trị 8byte

Vấn đề về độ lớn của tập hợp:

short < int < long < long long int ...

float < double < long double

D. Kiểu ký tự char

Kiểu ký tự đại diện cho bảng mã ASCII

Kiểu char trong C: là kiểu ký tự, có độ lớn là 1byte, biểu diễn các mẫu tự hiển thị, điều khiển và mở rộng của bàn phím. Một ký tự là một mẫu tự nằm trong dấu 2 dấu ‘ ‘

Ví dụ:

ký tự 'a' ,
ký tự '@' ,
ký tự '1'

[illegible]

Phần 2

Khai báo biến



A. Cách khai báo biến

Khai báo một biến

Tên Kiểu Dữ Liệu **Tên Biến ;**

Khai báo nhiều biến

Tên Kiểu Dữ Liệu **Tên Biến1, Tên Biến2... ;**

Ví dụ:

khai báo biến a là số nguyên:

int **a;**

Khai báo biến b,c là kiểu thực:

double **b,c;**

Khai báo biến d là kiểu ký tự:

char **c;**

```
1 void main()  
2 {  
3     int a;  
4     double b,c;  
5     char d;  
6 }
```

B. Cách đặt tên biến

```
1 void main()
2 {
3     //Tên biến được viết thường và cần sát nghĩa với hoàn cảnh dùng
4     int count, temp;
5     double price, salary;
6     char key;
7     //Nếu tên có nhiều hơn 1 từ, sử dụng dấu "_" để phân tách
8     int time_count;
9     double monthly_salary;
10    char key_start;
11 }
```

```
1 void main()
2 {
3     //Phong cách bố trí mã nguồn theo kiểu Nhật
4     int          count, temp;          /* Lý giải về sử dụng count, temp */
5     double       price, salary;       /* Lý giải về sử dụng price, salary */
6     char         key;                 /* Lý giải về sử dụng key */
7
8 }
```

Thử viết lại ví dụ

(Xem Hướng Dẫn Trên Visual Studio)

Phần 3

Phép gán dữ liệu



A. Phép gán

Gán giá trị cho một biến

Tên Biến = Giá Trị;

Gán biểu thức cho một biến

Tên Biến = Biểu thức;

Gán ngay khi khai báo biến

Kiểu dữ liệu Tên Biến = Giá trị hoặc Biểu thức;

Khi một biến được gán bởi một giá trị: bất kể biến đó có giá trị là gì trước đó, sau khi được gán, biến sẽ nhận lấy giá trị này

Khi một biến được gán bởi một biểu thức: chương trình thực hiện tính giá trị biểu thức trước, sau đó lấy kết quả tính được đem gán cho biến.

B. Một số ví dụ

```
1 void main()  
2 {  
3     //Hãy lý giải các giá trị nhận được của biến value  
4     int value;  
5     value = 10;  
6     value = 10 + value;  
7     value = 0;  
8     //Hãy lý giải các giá trị các biến number1,number2,number3  
9     int number1 = 10, number2 = 20;  
10    int number3 = number1 + number2;  
11 }
```

Thử viết lại ví dụ

(Xem Hướng Dẫn Trên Visual Studio)

Phần 4

Các phép toán
cơ bản



A. Các toán tử thông dụng

Toán tử	Ví dụ	Kết quả
+	$A + B$	Tính tổng biến A và B
-	$A - B$	Tính hiệu A và B
*	$A * B$	Tính tích A và B
/	A / B	Nếu số A hoặc B là số thực thì thực hiện phép chia thông thường Nếu cả A và B là số nguyên thì thực hiện phép chia lấy nguyên
%	$A \% B$	Chia lấy dư với điều kiện cả A và B là số nguyên
++	$A++$ $++A$	Tăng A lên 1 giá trị, được thực hiện sau cùng trên cùng câu lệnh Tăng A lên 1 giá trị, được thực hiện trước tiên trên cùng câu lệnh
--	$A--$ $--A$	Giảm A xuống 1 giá trị, được thực hiện sau cùng trên cùng câu lệnh Giảm A xuống 1 giá trị, được thực hiện trước tiên trên cùng câu lệnh
+=	$A += B$	Tăng A lên B giá trị, là cách viết tắt của $A = A + B$
-=	$A -= B$	Giảm A xuống B giá trị, là cách viết tắt của $A = A - B$

B. Một số ví dụ

```
1 void main()  
2 {  
3     //Kết thúc dòng số 6, value1 và value2 có giá trị bao nhiêu?  
4     int value1 = 10, value2 = 20;  
5     value1 = ++value1 + value2 * (value2 / value1);  
6     value2 += 10;  
7     //Vì sao phải nhân 1.0 vào value2  
8     double division_param = value1 / (value2 * 1.0);  
9 }
```



Phần 5

Cách Debug trên
Visual Studio

A. Các bước debug trong VS

(Xem Hướng Dẫn Trên Visual Studio)

Phần 6

Một ví dụ phép gán
cơ bản hay



A. Ví dụ đảo giá trị và ý nghĩa

Yêu cầu: Cho biến a có giá trị là 10, b có giá trị là 20. Hãy đổi giá trị 2 biến này với nhau.

Cách 1: dùng thêm biến phụ c

```
1 void main()  
2 {  
3     int a = 10, b = 20, c;  
4     c = a; //a "nhờ" c giữ hộ giá trị  
5     a = b; //sau đó a nhận giá trị của b  
6     b = c; //b nhận giá trị cũ của a đang lưu ở c  
7 }
```

*Ưu nhược
điểm từng
cách là gì?*

Cách 2: không sử dụng biến phụ c

```
1 void main()  
2 {  
3     int a = 10, b = 20;  
4     a = a + b; //a = 30  
5     b = a - b; //b = 10  
6     a = a - b; //a = 20  
7 }
```

*Ta rút ra
được ý
nghĩa gì?*

Phần 7

Các thư viện tiêu
chuẩn thường
dùng




A. Các hàm tính toán trong math.h

Hàm	Ví dụ	Kết quả
sin,cos,tan	sin(x)	Tính sin, cos, tang cho biến x
log,log10	log(x)	Tính logarit và logarit cơ số 10 của x
abs	abs(x)	Tính trị tuyệt đối của x
sqrt	sqrt(x)	Tính căn bậc 2 của x
pow	pow(x,y)	Tính x mũ y
ceil	ceil(x)	Tìm số nguyên nhỏ nhất mà vẫn lớn hơn hoặc bằng x
floor	floor(x)	Tìm số nguyên lớn nhất mà vẫn nhỏ hơn hoặc bằng x
fmod	fmod(x,y)	Tính số dư trong phép chia thực của x/y
exp	exp(x)	Tính e mũ x

Chú ý: biến x, y truyền vào cho hàm cần có kiểu double

B. ví dụ viết biểu thức

$$5y^2 + \sqrt[3]{3y}$$



```
1  #include <math.h>
2  void main()
3  {
4      double value,y;
5      value = 5 * y * y + pow(3 * y, (1.0 / 3));
6  }
```

C. Printf của stdio.h và getch của conio.h

printf("nội dung hiển thị")

→ Thư viện *stdio.h*.

Sử dụng các chỉ lệnh *\t: tab, \n: newline*

printf("%kiểu", biến)

→ Sử dụng các chỉ lệnh *%d: int %lf: double %c: char*

_getch()

→ Thư viện *conio.h*

Chương trình chờ nhập một ký tự để trả lại ký tự đó, thường được sử dụng để xem kết quả thực hiện của lệnh printf

C. Một số ví dụ printf, getch

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     printf("noi dung can hien thi ra man hinh");
6     _getch();
7 }
```

noi dung can hien thi ra man hinh

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     printf("noi dung\n\n can hien thi\n\t\t ra man hinh");
6     _getch();
7 }
```

noi dung
can hien thi
ra man hinh

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     int a = 10;
6     double b = 10.5;
7     char c = 'x';
8     printf("so nguyen %d so thuc %lf ky tu %c", a, b, c);
9     _getch();
10 }
```

so nguyen 10 so thuc 10.500000 ky tu x

TRÂN TRỌNG CẢM ƠN

