

HVIT CLAN JSC

C Programming

Hàm trong C

Nội dung slide: Viết hàm trong C

Thuyết minh: Nguyễn Đồng Khánh

HVIT CLAN 2020



NỘI DUNG CHÍNH (3 nội dung)



Khai báo hàm



Hàm void



Không dùng đệ quy!

Phần 1

Khai báo hàm



Khai báo hàm

Hàm: Là một đoạn mã được đặt tên, có thể có tham số truyền vào.

Kiểu Tên_Hàm (**kiểu** tham_số1, **kiểu** tham_số2...)

```
{  
    //Công việc  
    return giá_trị;  
}
```

```
4 int Tong(int a, int b)  
5 {  
6     int c = a + b;  
7     return c;  
8 }
```

Chú ý:

- Khi gặp câu lệnh **return**, hàm sẽ dừng lại, lấy **giá_trị** đằng sau câu lệnh return gán cho **Tên_Hàm**, người ta gọi việc này là **hàm được trả về giá trị**
- **Kiểu** của **tham_số** truyền vào cho hàm không liên quan gì đến **giá_trị** trả về cho hàm

Gọi Hàm:

- Gọi bởi tên, kèm theo tham số truyền phù hợp
- Có thể gọi hàm lồng với hàm
- Có thể truyền tham số bởi các biến số

```
7 void main()  
8 {  
9     int x = Tong(3, 4);  
10    int y = Tong(3, Tong(4, 5));  
11    printf("%d ", Tong(x, y));  
12 }
```

Phần 2

Hàm void



Hàm void

Hàm void:

Khi mà hàm không phải trả lại một giá trị cụ thể sau khi thực hiện công việc.
Ví dụ như in ra một nội dung nào đó. Khi đó, hàm sẽ nên là hàm void.

void Tên_Hàm (*kiểu tham_số1*, *kiểu tham_số2...*)

```
{  
    //Công việc  
    //có thể có return;  
}
```

Chú ý: Hàm void vẫn có thể có câu lệnh **return** !

Người ta sử dụng câu lệnh return trong hàm void nhằm ngắt chuỗi hành động của hàm hoặc phục vụ mục đích cụ thể.

```
1  #include <stdio.h>  
2  void Hien(int n)  
3  {  
4      if (n < 10) return;  
5      for (int i = 0; i < n; i++)  
6      {  
7          printf("_");  
8      }  
9      printf("\n");  
10 }  
11 void main()  
12 {  
13     Hien(3);  
14     Hien(10);  
15     Hien(20);
```

Phần 3

Đệ quy



Đệ quy rất nguy hiểm và chậm

Cùng tìm hiểu ví dụ về hàm Hiên thị số từ 1 đến n không dùng vòng lặp

~~Hien(1)~~

~~ln(1)~~

~~Hien(2)~~

~~ln(2)~~

~~Hien(3)~~

~~ln(3)~~

~~Hien(4)~~

~~Hien(1)~~

~~Hien(2)~~

~~ln(1)~~

~~Hien(3)~~

~~ln(2)~~

~~Hien(4)~~

~~ln(3)~~

TRÂN TRỌNG CẢM ƠN

