

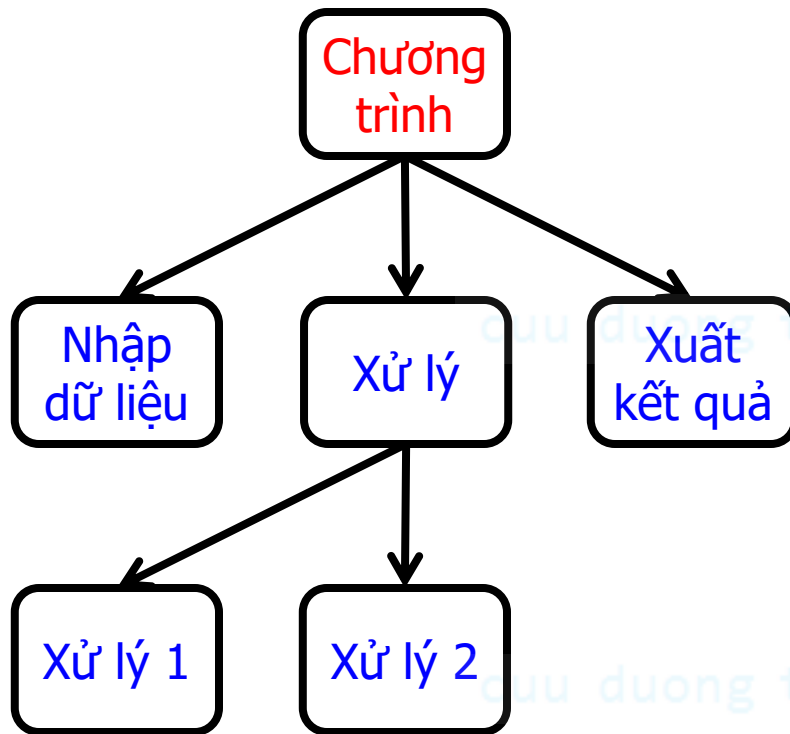
# Hàm & Kỹ thuật tổ chức chương trình

## Phần a: Giới thiệu

### **Nhập môn lập trình**

Trình bày: Nguyễn Sơn Hoàng Quốc  
Email: [nshquoc@fit.hcmus.edu.vn](mailto:nshquoc@fit.hcmus.edu.vn)

# Tiếp cận top-down

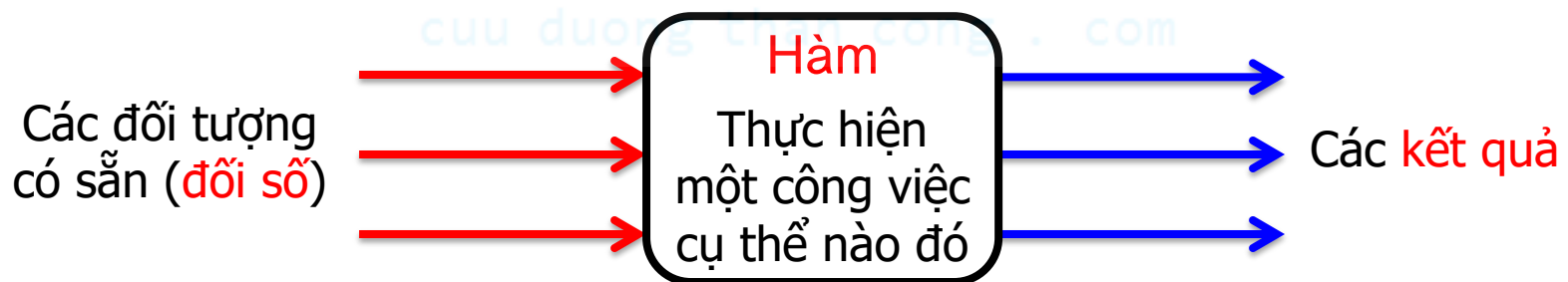


Tiếp cận top-down  
trong lập trình cấu trúc

**Chương trình lớn** được **chia** thành các **chương trình con** nhỏ hơn nhằm dễ dàng phân chia và kiểm tra công việc hay sử dụng lại những bộ phận đã hoàn tất.

# Đặc điểm

- **Hàm** có các đặc điểm sau:
  - Có một **tên duy nhất**.
  - Là một thành phần **độc lập**.
  - Thực hiện một **công việc cụ thể**.
  - Có thể **nhận** các **đối số**.
  - Có thể **trả về kết quả** cho chương trình gọi nó.



# Nguyên mẫu hàm

```
return-type function_name(param-type param_name,  
..., param-type param_name);
```

- Trong đó:
  - **return-type**: **kiểu** của giá trị hàm sẽ **trả về**, nếu không trả về gì cả thì kiểu trả về sẽ là **void**.
  - **function\_name**: **tên của hàm**, thể hiện công việc hàm sẽ làm, nên bắt đầu bằng một động từ.
  - **param-name**, **param-type**: **tên** và **kiểu** tương ứng của tham số hình thức (formal parameter).
  - Được **kết thúc** bằng dấu chấm phẩy ;

# Định nghĩa hàm

```
return-type function_name(param-type param_name,  
                           ..., param-type param_name)  
{  
    // statements here...  
}
```

- Trong đó:
  - Dòng đầu là **tiêu đề hàm** (giống nguyên mẫu hàm nhưng **không có ;** và **bắt buộc** phải có **tên tham số**).
  - Tiếp theo là **thân hàm** (đặt trong {}) chứa các câu lệnh hàm sẽ thực hiện (phải có ít nhất một lệnh **return** nếu kiểu trả về không phải là **void**)

## Phân biệt một cách tương đối

- **Hàm có sẵn** (trong **ngôn ngữ** hoặc do một hãng **phần mềm viết** để bán hoặc cho) như:
  - **Hàm xuất, nhập thông tin**: printf(), scanf(), ...
  - **Hàm toán học**: sqrt(), pow(), abs(), sin(), ...
- **Hàm** do người **lập trình viết thêm** như:
  - **Hàm xuất, nhập thông tin**: Nhập số dương...
  - **Hàm toán học**: Tính căn bậc 3, tính căn bậc n, tính giai thừa, giải phương trình bậc 1, bậc 2, bậc 4 đối xứng, ...

# Ví dụ về hàm có sẵn

```
1. void main()  
2. {  
3.     double a = 7, b = 5;  
4.     double z = 9;  
5.     printf("a = %d\n", a);  
6.     printf("b = ");  
7.     scanf("%lf", &b);  
8.     z = pow(b, a);  
9. }
```

Đổi số



Biến nhận giá trị  
trả về của hàm



## Ví dụ về hàm tự viết thêm

- Hàm tính  $\sqrt[3]{x}$  ( $x \in \mathbb{R}$ ) chưa có trong thư viện `math.h`

$x \rightarrow$  hàm `sqrt(3)`  $\rightarrow \sqrt[3]{x}$

- Lưu ý:

$$\sqrt[3]{x} = \begin{cases} 0 & \text{nếu } x = 0 \\ \text{pow}(x, 1.0/3) & \text{nếu } x > 0 \\ -\text{pow}(-x, 1.0/3) & \text{nếu } x < 0 \end{cases}$$

- Khai báo hàm: `double sqrt3(double x);`



# Định nghĩa hàm sqrt3()

```
1. double sqrt3(double x)
2. {
3.     double y = 0;    // temporary variable
4.
5.     if (x > 0)
6.         y = pow(x, 1/(double)3);
7.     else
8.         if (x < 0)
9.             y = -pow(-x, 1/(double)3);
10.
11.     return y;        // returns result
12.} // end of function
```

## Ví dụ minh họa

- Viết hàm tìm giá trị lớn nhất của hai số nguyên  $a$  và  $b$ .

cuu duong than cong . com

cuu duong than cong . com

# Ví dụ minh họa

```
1. int LonNhat(int a, int b)
2. {
3.     int max = a;
4.     if (b > a)
5.         max = b;
6.     return max;
7. }
```