

Hàm & Kỹ thuật tổ chức chương trình

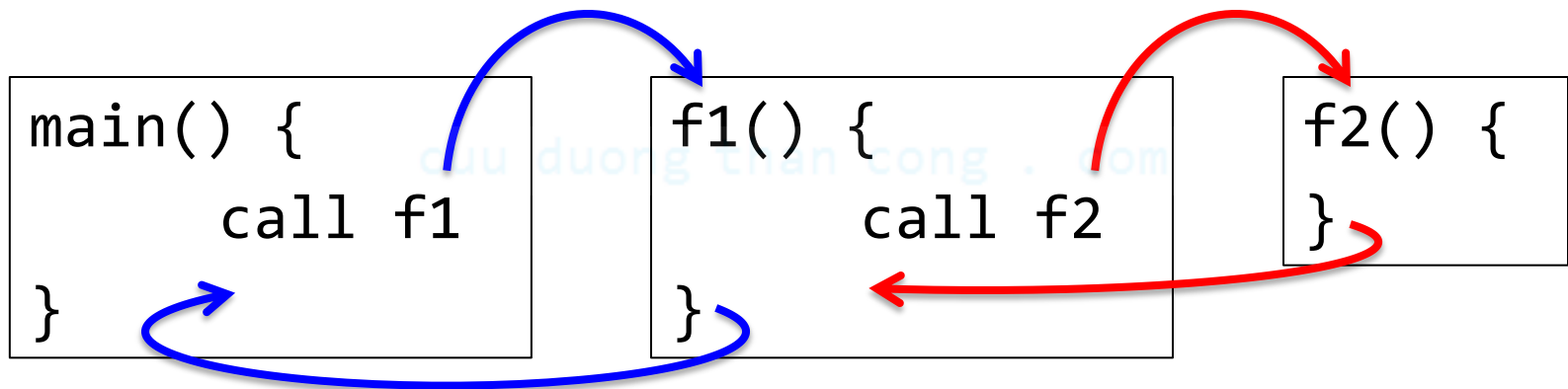
Phần b: Truyền tham số cho hàm

Nhập môn lập trình

Trình bày: Nguyễn Sơn Hoàng Quốc
Email: nshquoc@fit.hcmus.edu.vn

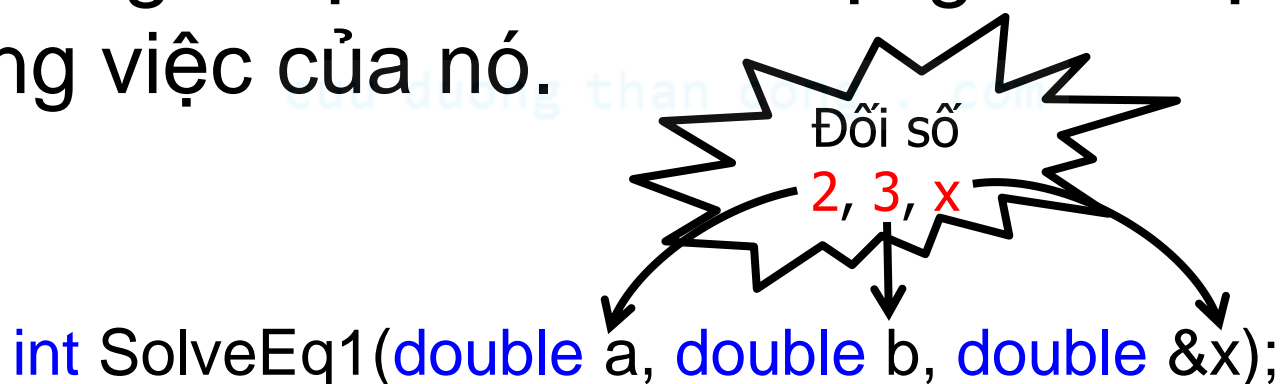
Sự thực thi của hàm

- Các câu lệnh bên trong hàm chỉ được **thực thi** khi hàm **được gọi** từ một phần khác của chương trình.
- Khi gọi hàm, chương trình có thể truyền đến hàm thông tin dưới dạng **một hay nhiều đối số**.



Khái niệm đối số

- **Đối số** (argument) hay tham số thực (actual parameter) là **dữ liệu** của chương trình **truyền đến hàm** có kiểu dữ liệu ứng với tham số hình thức được khai báo trong nguyên mẫu hàm. Dữ liệu này thường được hàm sử dụng để thực hiện công việc của nó.



Truyền đối số cho hàm

- Có hai cách truyền đối số
 - Truyền **bằng giá trị** (pass by value)
 - Đối số **không đổi** do hàm tạo bản sao của đối số khi nhận.
 - Thông thường là **dữ liệu có sẵn**.
 - Tham số hình thức tương ứng được gọi là **tham trị**.
 - Truyền **bằng tham chiếu** (pass by reference): C++
 - Đối số **có thể thay đổi** khi gọi hàm.
 - Thông thường là **dữ liệu cần tính toán, xác định**.
 - Tham số hình thức tương ứng được gọi là **tham chiếu** hay **tham biến**.

Ví dụ về tham trị

```
int Inc(int x);  
void main() {  
    int a = 9, b;  
    b = Inc(a); // a is passed by value  
    printf("a = %d, b = %d\n", a, b);  
}  
int Inc(int x) {  
    x++;  
    return x;  
}
```

int x = 9;

Alias (Bí danh)

```
1. int dtbCuaNam = 5;  
2. int &dtbCuaTi = dtbCuaNam;  
3. dtbCuaTi = 6;  
4. printf("%d", dtbCuaNam)
```

“Nam” có bí danh là “Ti”

dtbCuaNam 6 dtbCuaTi

Có thể xem **dtbCuaNam** và **dtbCuaTi** là **như nhau**

Ví dụ về tham biến

```
int Inc(int &x); // C++
```

```
void main() {
```

```
    int a = 9, b;
```

```
    b = Inc(a);
```

```
    printf("a = %d,
```

```
          b = %d\n", a, b);
```

```
}
```

```
int Inc(int &x) {
```

```
    x++;
```

```
    return x;
```

```
}
```

```
int Inc(int *x); /* C */
```

```
void main() {
```

```
    int a = 9, b;
```

```
    b = Inc(&a);
```

```
    printf("a = %d,
```

```
          b = %d\n", a, b);
```

```
}
```

```
int Inc(int *x) {
```

```
    (*x)++;
```

```
    return (*x);
```

```
}
```

Địa chỉ
của a

int *x = &a;

Tham biến hằng

```
void f1(double x);  
void f2(double &x);  
void f3(const double &x);  
void main() {  
    double a = 15.06;  
    f1(a); // passed by value  
    f2(a); // passed by reference  
    f3(a); // passed by const reference  
}  
// defines f1(), f2(), f3() here...
```

double x = 15.06;

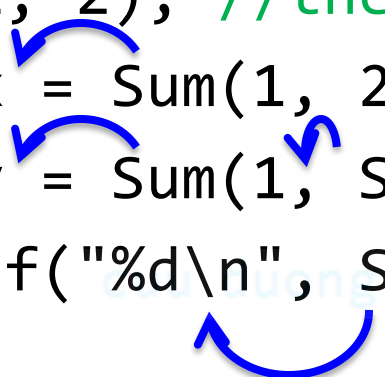
Tồn bộ nhớ
khi x lớn

Lời gọi hàm

- Có hai cách để gọi hàm
 - Mọi hàm đều có thể được gọi bằng cách sử dụng **tên hàm** kèm danh sách **các đối số** trong một câu lệnh đơn. Nếu hàm có giá trị trả về, giá trị này sẽ bị bỏ qua.
 - Đối với các hàm có **giá trị trả về**, do các hàm này được quy thành một giá trị (do hàm trả về) nên chúng là các **biểu thức C** hợp lệ và có thể được sử dụng ở bất kỳ nơi đâu mà một biểu thức C có thể được sử dụng.

Ví dụ lời gọi hàm

```
1. void DoSomething();
2. int Sum(int x, int y);
3. void main() {
4.     DoSomething();
5.     Sum(1, 2); //the return value is discarded
6.     int x = Sum(1, 2);
7.     int y = Sum(1, Sum(2, 3));
8.     printf("%d\n", Sum(1, 2));
9. }
10.// defines DoSomething() and Sum() here...
```



Lưu ý về lời gọi hàm

- Nếu cố sử dụng các hàm có kiểu trả về là `void` như một biểu thức thì trình biên dịch sẽ phát sinh một thông báo lỗi.

```
1. void DoSomething();  
2. void main() {  
3.     DoSomething();  
4.     int x = DoSomething(); // error  
5.     printf("%d\n", DoSomething()); // error  
6. }  
7. // defines DoSomething() here...
```

Lưu ý về lời gọi hàm

- Hãy truyền đối số vào hàm để làm cho hàm tổng quát để có thể tái sử dụng.

```
1. int Sum() { // non generic
2.     int x, y;
3.     // inputs x, y here...
4.     return x + y;
5. }
6. int Sum(int x, int y){//generic & reusable
7.     return x + y;
8. }
```

Lưu ý về lời gọi hàm

- Nên tận dụng ưu điểm của khả năng đặt hàm vào trong biểu thức nhưng tránh làm cho câu lệnh dài dòng, khó hiểu.

```
1. int Sum(int x, int y);
```

```
2. void main() {
```

```
3.     int a = 1, b = 2, c = 3, x;
```

```
4.     printf("%d", Sum(a, Sum(b, c))); // !!!
```

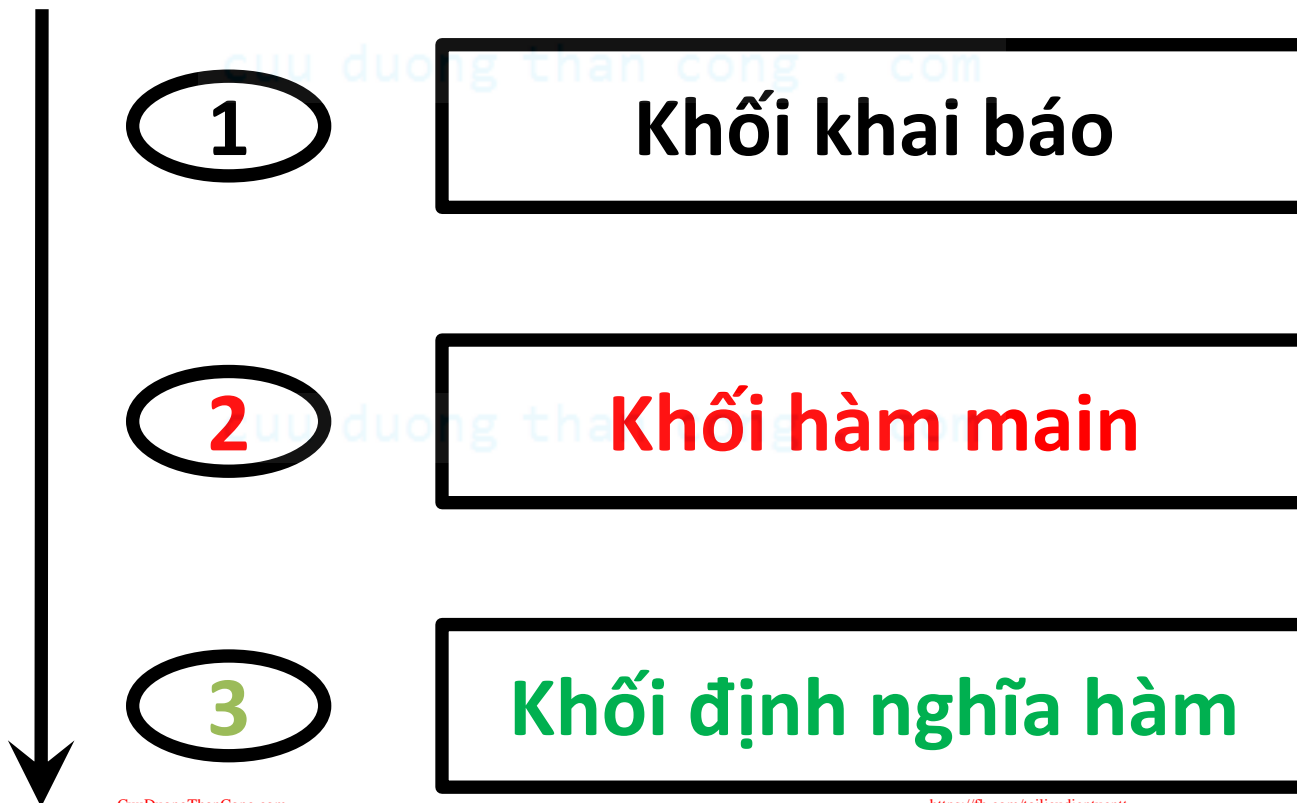
```
5.     x = Sum(b, c);
```

```
6.     printf("%d", Sum(a, x));           // better
```

```
7. }
```

Kiến trúc một chương trình đơn giản

- Kiến trúc của một chương trình cơ bản gồm 3 khối lệnh chính: **khối khai báo**, **khối hàm main** và **khối định nghĩa hàm**



Kiến trúc một chương trình đơn giản

- **Khối khai báo:** chứa các **khai báo** sử dụng thư viện, khai báo hằng, khai báo biến toàn cục, khai báo hàm, khai báo kiểu dữ liệu...
- **Khối hàm main:** chứa duy nhất **hàm main** và thân hàm của nó. Trong thân hàm main chứa các lời gọi hàm cần thiết cho chương trình.
- **Khối định nghĩa hàm:** chứa các **định nghĩa hàm** đã được khai báo trong khối khai báo.

Ví dụ minh họa 1 (Tìm số lớn nhất)

- Viết chương trình nhập vào hai số nguyên a, b . Tìm số lớn nhất và xuất kết quả

cuu duong than cong . com

cuu duong than cong . com

Ví dụ minh họa 1 (Số lớn nhất) (Khởi khai báo)

```
1. #include <stdio.h>
```

```
2. void Nhap(int &x);
```

```
3. void Xuat(int x);
```

```
4. int LonNhat(int x, int y);
```

cuu duong than cong . com

Ví dụ minh họa 1 (Số lớn nhất) (Khởi hàm main)

```
1. void main()  
2. {  
3.     int a, b, max;  
4.     printf("Nhap a = ");  
5.     Nhap(a);  
6.     printf("Nhap b = ");  
7.     Nhap(b);  
8.     max = LonNhat(a, b);  
9.     printf("So lon nhat la = ");  
10.    Xuat(max);  
11. }
```

Ví dụ minh họa 1 (Khởi định nghĩa hàm)

```
1. void Nhap(int &x)
```

```
2. {
```

```
3.     scanf("%d", &x);
```

```
4. }
```

```
5. void Xuat(int x)
```

```
6. {
```

```
7.     printf("%d", x);
```

```
8. }
```

Ví dụ minh họa 1 (Khởi định nghĩa hàm)

```
1. int LonNhat(int x, int y)
2. {
3.     int max = x;
4.
5.     if (y > x)
6.         max = y;
7.
8.     return max;
9. }
```

Ví dụ minh họa 2 (Tính tổng)

- Viết chương trình nhập vào hai số nguyên dương. Tính tổng giữa chúng và xuất kết quả

cuu duong than cong . com

cuu duong than cong . com

Ví dụ minh họa 2 (Tính tổng) (Khối khai báo)

1. `#include <stdio.h>`

2. `void Nhap(int &x);`

3. `void Xuat(int x);`

4. `int Tong(int x, int y);`

[cuu duong than cong . com](http://cuuduongthancong.com)

Ví dụ minh họa 2 (Tính tổng) (Khối hàm main)

```
1. void main()  
2. {  
3.     int a, b, tong;  
4.     printf("Nhap a = ");  
5.     Nhap(a);  
6.     printf("Nhap b = ");  
7.     Nhap(b);  
8.     tong = Tong(a, b);  
9.     printf("Tong = ");  
10.    Xuat(tong);  
11. }
```

Ví dụ minh họa 2 (Khởi định nghĩa hàm)

```
1. void Nhap(int &x)
2. {
3.     scanf("%d", &x);
4. }
```

```
5. void Xuat(int x)
6. {
7.     printf("%d", x);
8. }
```

```
9. int Tong(int x, int y)
10. {
11.     return x + y;
12. }
```