



Chương 6

Hàm (funtions)

Nội dung

1

Khái niệm

2

Khai báo và định nghĩa hàm

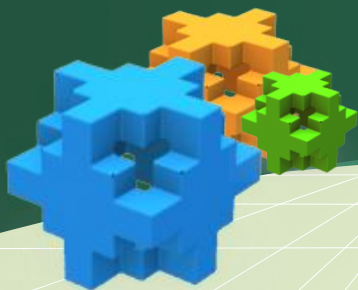
3

Lời gọi và sử dụng hàm

4

Biến và truyền tham số





CHƯƠNG 6: HÀM (FUNCTION)

❖ Mục tiêu

- ☐ Giới thiệu các kiến thức về hàm: khái niệm, cách khai báo và định nghĩa hàm, cách truyền tham số; hàm đệ quy, ...

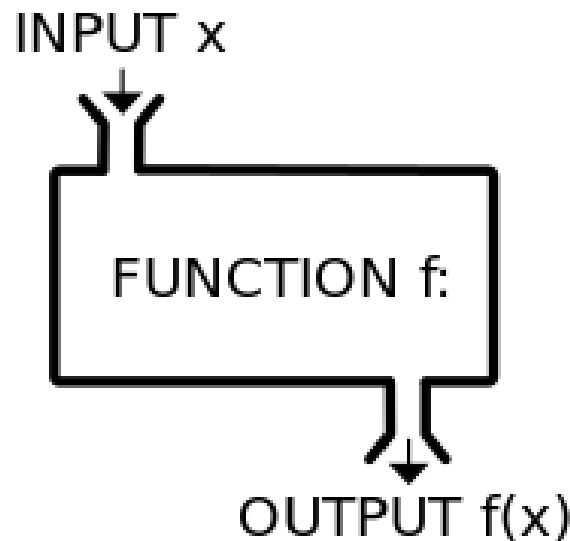
❖ Nội dung

- ☐ Khái niệm
- ☐ Định nghĩa hàm
- ☐ Cách truyền tham số
- ☐ Bài tập

Hàm (funtions)

1.Khái niệm

- Là một module chương trình, giải quyết một công việc hoàn chỉnh, được dùng nhiều lần trong chương trình



Hàm (funtions)

1.Khái niệm

- Hàm là một chương trình con, hàm có thể trả về hay không trả về giá trị; truyền hay không truyền các tham số.
- Một chương trình trong C++ có thể gồm nhiều hàm. Nhưng có một hàm chính với tên gọi là hàm **main()**.
- Khi thực thi chương trình, nó luôn bắt đầu từ hàm **main()**.
- Hàm giúp cho việc phân đoạn chương trình một cách riêng rẽ. Hàm có thể được sử dụng nhiều lần trong một chương trình hoặc sử dụng ở nhiều chương trình khác nhau.

Hàm (funtions)

1.Khái niệm

- Hàm có một số đặc điểm sau:
 - Nằm trong hoặc ngoài chương trình có lời gọi đến hàm. Trong một chương trình có thể chứa nhiều hàm.
 - Một hàm có thể được gọi từ hàm main(), từ một hàm khác hay được gọi bởi chính nó (trường hợp đệ quy).

2. Khai báo và định nghĩa hàm

2. Khai báo hàm:

Hàm được chia làm 2 loại:

- Hàm định nghĩa sẵn
- Hàm do người lập trình tự định nghĩa.

.

2. Khai báo và định nghĩa hàm

2. Khai báo hàm:

- Hàm định nghĩa sẵn: có trong tệp thư viện *.h.
Để sử dụng được các hàm này ta cần phải khai báo tệp tiêu đề theo cú pháp
`#include<Tentaptinthuvien>`

Ví dụ: `#include<iostream>`

`#include<cmath>`

2. Khai báo và định nghĩa hàm

2. Định nghĩa hàm:

- Hàm do người lập trình tự định nghĩa: có thể được khai báo ở đầu chương trình theo cú pháp:

Kieu_du_lieu Ten_ham(Danh_sach_kieu_doi_so);

2. Khai báo và định nghĩa hàm

2. Định nghĩa hàm:

Kieu_du_lieu Ten_ham (Danh_sach_kieu_doi_so);

- Trong đó:
 - **Kieu_du_lieu**: là kiểu trả về của hàm **int, float, char, ...** Trường hợp hàm không trả về giá trị thì khai báo kiểu **void**. Trường hợp không chỉ rõ **kiểu_tra_ve** thì ngầm định là kiểu **int**.
 - **Ten_ham**: được đặt theo quy tắc đặt tên.
 - **Danh_sach_kieu_doi_so**: có thể có hoặc không. Trường hợp không có thì gọi là hàm không có đối số.

2. Khai báo và định nghĩa hàm

2. Định nghĩa hàm:

Ví dụ 1: hàm trả về tổng hai số thực

```
/* ham tinh tong hai so thuc */  
double tong_hai_so( double x, double y) {  
    double z;  
    z = x + y;  
    return z;  
}
```

2. Khai báo và định nghĩa hàm

Định nghĩa hàm:

- Hàm trả về giá trị:
- Hàm không trả về giá trị:

2. Khai báo và định nghĩa hàm

Định nghĩa hàm:

- Hàm trả về giá trị:

Cú pháp:

```
kieu_du_lieu    Ten_ham(Ds_tham_so_hinh_thuc)
{
    Khai báo các biến cục bộ;
    Các câu lệnh;
    return(biểu thức);
}
```

2. Khai báo và định nghĩa hàm

Định nghĩa hàm:

- Hàm trả về giá trị:

Ví dụ:

```
1 int tinhTong2(int a, int b)
2 {
3     int tong = a + b;
4     return tong;
5 }
```

2. Khai báo và định nghĩa hàm

Khai báo và định nghĩa hàm

- **Lệnh return**
 - Trả về giá trị của hàm, có thể trả về là giá trị của 1 biểu thức
 - Có thể có nhiều lệnh return trong 1 hàm
 - Đối với hàm có giá trị trả về, phải CÓ ít nhất 1 lệnh return

2. Khai báo và định nghĩa hàm

Định nghĩa hàm:

- Hàm không trả về giá trị:

Nếu hàm không trả về giá trị (còn gọi là hàm kiểu **void**):

- **kieudulieu**: Sử dụng từ khóa **void**
- Trong thân hàm có thể có khoặc không câu lệnh **return**. Nhưng nếu có thì sau **return** không có biểu thức.

2. Khai báo và định nghĩa hàm

Định nghĩa hàm:

- Hàm không trả về giá trị:

ví dụ 1 :

```
1 void tinhTong1(int a , int b )
2 {
3     printf("%d", a + b);
4 }
```

ví dụ 2 :

```
1 void ClearSrceen()
2 {
3     int i;
4     for (i=1; i<=100; i++)
5         system("cls");
6 }
```


3. Lời gọi và sử dụng hàm

Khái niệm và gọi hàm:

- Khái niệm:
 - ✓ Gọi hàm là việc chuyển quyền điều khiển từ hàm đang gọi đến hàm được gọi.
 - ✓ Hàm có thể được gọi từ hàm khác hoặc từ chính bản thân nó.
- **Gọi hàm:** Để thực hiện việc gọi hàm, sử dụng tên của hàm được gọi và theo sau là các tham số thực tế được đặt trong cặp dấu ngoặc đơn ().

Cú pháp: **Ten_ham(Ds_tham_so_thuc_te);**

Trong đó: **Ds_tham_so_thuc_te** phân tách nhau bởi dấu phẩy (,)

3. Lời gọi và sử dụng hàm

Khái niệm và gọi hàm:

Ví dụ: Tính biểu thức $2x^3 - 5x^2 - 4x + 1$

```
double luythua(float x, int n)
{
    int i;
    double kq = 1;
    for (i=1; i<=n; i++) kq *= x;
    return kq;
}
```

```
void main()
{
    float x;
    double f;
    cout << "x =" ; cin >> x;
    f = 2*luythua(x,3)-5*luythua(x,2)-4*x + 1;
}
```

3. Lời gọi và sử dụng hàm

Khái niệm và gọi hàm:

```
#include<iostream.h>

using namespace std;

void ClearSrceen()
{
    int i;
    for (i=1; i<=100; i++)
        system("cls") ;
    return ;
}

int main()
{
    ClearSrceen() ;
}
```

3. Lời gọi và sử dụng hàm

Vị trí hàm:

- Có thể khai báo sau hoặc trước hàm main().
 - Nếu khai báo sau hàm main() thì phải khai báo prototype của hàm ở đầu chương trình, sau đó phần triển khai nội dung hàm được đặt sau hàm main()
 - Function prototype:
 - Khai báo kiểu hàm, tên hàm và các tham số, kết thúc bằng dấu chấm phẩy
- Ví dụ: `int SUM(int x, int y);` \Rightarrow prototype của hàm SUM

3. Lời gọi và sử dụng hàm


Vị trí hàm:

Ví dụ:

```
int x=5, y=3, z; z = addition ( x , y );
```

```
int addition (int a, int b)
```

```
z = addition ( x , y );
```



4. Biến và truyền tham số

- **Phân loại biến:**

- + Biến thường

- + Biến con trỏ (với dấu * trước tên biến)

- + Biến tham chiếu: Thực chất là một bí danh được gán cho một biến nào đó. Lúc này, chỗ nào xuất hiện biến thì cũng tương đương dùng bí danh và ngược lại.

- **Cú pháp khai báo biến tham chiếu:**

Kieudulieu &ten_bien_tham_chieu=ten_bien_duoc_tham_chieu;

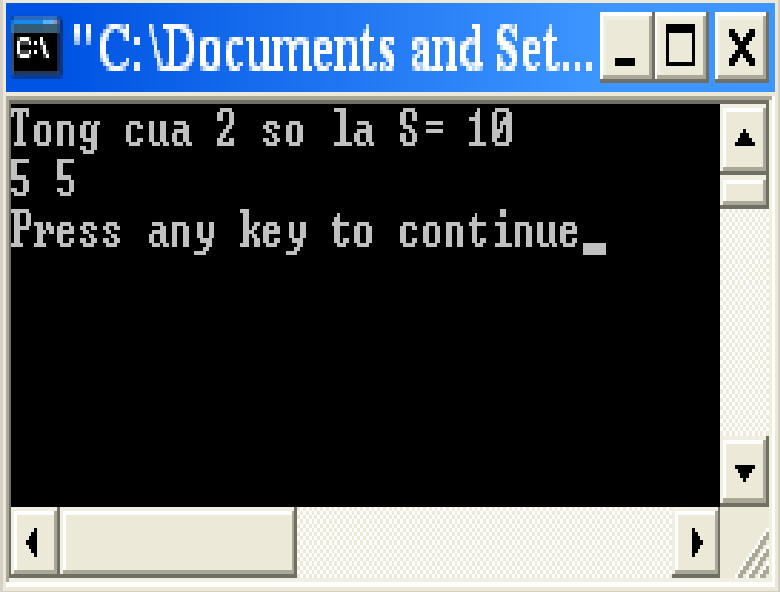
4. Biến và truyền tham số

- **Truyền tham số:** Có 3 cách truyền
 - + Truyền theo **tham trị**
 - + Truyền theo **tham biến**
 - + Truyền theo **con trỏ**

4. Biến và truyền tham số

Ví dụ:

```
int Tong(int, int);  
int main()  
{ int a = 5, b = 5;  
  cout<<"Tong cua 2 so la S= ";  
  cout<<Tong(a,b)<<endl;  
  cout<<a<<" "<<b<<endl;  
  return 0;  
}  
int Tong(int x, int y)  
{ return x+y; }
```



The screenshot shows a Windows command prompt window with the title bar "C:\Documents and Set...". The window contains the following text:
Tong cua 2 so la S= 10
5 5
Press any key to continue_

Bài tập

1. Tìm số lớn nhất trong 3 số a,b,c nhập từ bàn phím
2. Tính giai thừa $n! = 1 * 2 * 3 * \dots * (n-1) * n$.
3. Viết hàm tính tổng $S = 1 + 2 + \dots + n$.
4. Tìm số ký số của n. ví dụ $n = 12345 \rightarrow$ số ký số: 5
5. Tìm số đảo ngược của n
6. Với n cho trước. Kiểm tra n có phải số hoàn hảo không?
7. Với n cho trước. Kiểm tra n có phải số nguyên tố ?

Bài tập

Viết chương trình tính $S = a! + b! + c!$ với a, b, c là 3 số nguyên dương nhập từ bàn phím.

