

BÀI 6

CHƯƠNG TRÌNH CON

(HÀM – THỦ TỤC)



Giảng viên: **Nguyễn Văn Đồng – Khoa Công nghệ thông tin – ĐH Thủy Lợi**

Email: nvdong@tlu.edu.vn

NỘI DUNG

- Khái niệm hàm
- Khai báo hàm
- Sử dụng hàm
- Phạm vi của biến



BÀI TOÁN

```
#include <iostream>
#include<cmath>
using namespace std;

int main()
{
    double x=1;
    double fx;
    fx= pow(x,3) + 2*pow(x,2) -3;
    cout<<"fx = "<<fx<<endl;
    return 0;
}
```

BÀI TOÁN

```
#include <iostream>
#include<cmath>
using namespace std;

int main()
{
    double x=1, y=2, z=3;
    double fx, fy, fz;
    fx= pow(x,3) + 2*pow(x,2) -3;
    fy= pow(y,3) + 2*pow(y,2) -3;
    fz= pow(z,3) + 2*pow(z,2) -3;
    cout<<"fx = "<<fx<<endl;
    cout<<"fy = "<<fy<<endl;
    cout<<"fz = "<<fz<<endl;
    return 0;
}
```

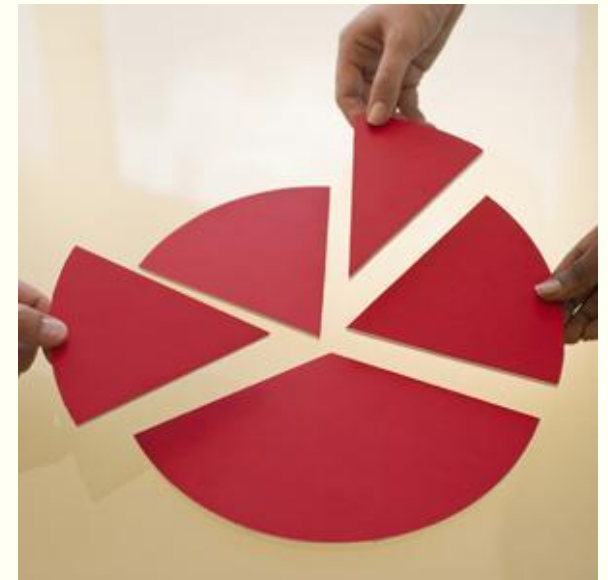
```
#include <iostream>
#include<cmath>
using namespace std;

double f(double x)
{
    double fs=pow(x,3) + 2*pow(x,2) -3;
    return fs;
}

int main()
{
    double x=1, y=2, z=3;
    double fx, fy, fz;
    fx= f(x);
    fy= f(y);
    fz= f(z);
    cout<<"fx = "<<fx<<endl;
    cout<<"fy = "<<fy<<endl;
    cout<<"fz = "<<fz<<endl;
    return 0;
}
```

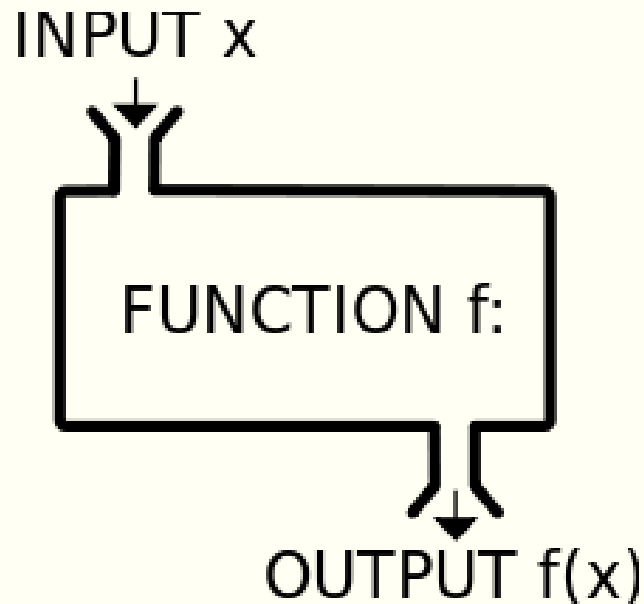
KHÁI NIỆM HÀM

- Khi bài toán quá lớn, khó phát triển -> chia thành các bài toán nhỏ
- Các bài toán nhỏ gọi là hàm
- Hàm **main()** là hàm khởi nguồn, thực hiện đầu tiên



KHÁI NIỆM HÀM

- Là tập các câu lệnh được tách ra từ chương trình chính
- Có thể có giá trị đầu vào và trả kết quả đầu ra
- Có thể được gọi nhiều lần trong chương trình



CẤU TRÚC CHƯƠNG TRÌNH KHI VIẾT HÀM

```
//Phần khai báo thư viện  
#include <iostream>  
using namespace std;
```

```
//Phần định nghĩa hàm
```

```
//Hàm chính  
int main () {
```

```
    //Lời gọi hàm
```

```
    return 0;
```

```
}
```

**Viết định nghĩa
hàm tại đây**

Gọi hàm

CẤU TRÚC CHƯƠNG TRÌNH KHI VIẾT HÀM

```
//Phần khai báo thư viện
#include <iostream>
using namespace std;


//Phần khai báo nguyên mẫu hàm

//Hàm chính
int main () {

    //Lời gọi hàm

    return 0;
}

//Phần định nghĩa hàm
```



Viết nguyên mẫu hàm tại đây



Gọi hàm



Viết định nghĩa hàm tại đây

CẤU TRÚC CHƯƠNG TRÌNH KHI VIẾT HÀM

▪ Ví dụ:

```
#include <iostream>
using namespace std;
double binhPhuong(double x);

int main () {
    double x;
    cout<<"Nhap gia tri x=";
    cin>>x;
    cout<<"x binh phuong ="<<binhPhuong(x);
    return 0;
}

double binhPhuong(double x)
{
    double s = x*x;
    return s;
}
```

```
#include <iostream>
using namespace std;

double binhPhuong(double x)
{
    double s = x*x;
    return s;
}

int main () {
    double x;
    cout<<"Nhap gia tri x=";
    cin>>x;
    cout<<"x binh phuong ="<<binhPhuong(x);
    return 0;
}
```

ĐỊNH NGHĨA HÀM

(Xem mục 2.6 trong giáo trình)

▪ Cú pháp:

```
kiutrave tenham (kieudulieu thamso1, kieudulieu thamso2,..)
{
    //các câu lệnh xử lý

    return giatri; //câu lệnh trả về giá trị
}
```



Tên hàm và số tham số phải trùng với nguyên mẫu hàm

Không có dấu ; khi định nghĩa hàm

KHAI BÁO NGUYÊN MẪU HÀM

(Xem mục 2.6 trong giáo trình)

▪ Chỉ là mô tả mẫu hàm

- Tên hàm
- Các tham số

▪ Cú pháp:

```
kietrave tenham (kieudulieu thamso1, kieudulieu thamso2,..);
```

```
kietrave tenham (kieudulieu, kieudulieu ....);
```

- **kietrave**: kiểu dữ liệu trả về của hàm
- **tenham**: đặt theo quy tắc định danh
- **kieudulieu**: là kiểu dữ liệu của các tham số đầu vào
- **thamso1, thamso2**: tên các tham số đầu vào, sử dụng trong hàm, đặt theo quy tắc định danh

ĐỊNH NGHĨA HÀM

- Ví dụ:

```
#include <iostream>
using namespace std;
double binhPhuong(double);
int main () {
    double x, s;
    cout<<"Nhap gia tri x="; cin>>x;
    s= binhPhuong(x);
    cout<<"x*x ="<<s;
    return 0;
}

double binhPhuong(double x)
{
    double s = x*x;
    return s;
}
```

LỜI GỌI HÀM

- Là gọi hàm để sử dụng

- Cú pháp:

```
tenham (giatri1, giatri2...)
```

- Các trường hợp gọi hàm:

- Gọi trong lệnh gán giá trị cho biến
- Gọi trong biểu thức toán học
- Gọi trong câu lệnh ghi ra màn hình

- Ví dụ:

```
double s = tong(a, 10, 2.5); //Goi ham tinh tong 3 so  
double tb = tong(a, b, c)/3; //Goi ham tinh tong 3 so  
cout<<tong(a,b,c);
```


ĐỊNH NGHĨA HÀM

▪ Ví dụ:

gọi hàm

```
#include <iostream>
using namespace std;
double binhPhuong(double);
int main () {
    double x, s;
    cout<<"Nhap gia tri x="; cin>>x;
    s= binhPhuong(x);
    cout<<"x*x ="<<s;
    return 0;
}

double binhPhuong(double x)
{
    double s = x*x;
    return s;
}
```



LỆNH return

- Trả về giá trị cho hàm
- Có tác dụng kết thúc hàm
- Có thể trả về giá trị của cả biểu thức
- Có thể xuất hiện lệnh `return` nhiều lần trong hàm

- Ví dụ:

```
double tuyetDoi(double u, double v)
{
    double s = u + v ;
    if(s>0)
        return s;
    else
        return -s;
}
```



TÌM LỖI SAI TRONG CHƯƠNG TRÌNH SAU?

```
1. #include <iostream>
2. using namespace std;
3. double bieuThuc(double, double)
4.
5. int main () {
6.     double x, y;
7.     cout<<"Nhap gia tri x, y:";
8.     cin>>x>>y;
9.     cout<<"Bieu thuc voi x= "x<<" va "<<20<<" la:" <<bieuThuc(x, 20)<<endl;
10.    cout<<"Bieu thuc voi x= "<<x<<" va "<<y<<" la:" <<bieuThuc(z, y)<<endl;
11.    cout<<"Bieu thuc voi x= "<<x<<" va "<<x<<" la:" <<bienThuc(x)<<endl;
12.    return 0;
13. }
14.
15. double biThuc(double x, int y)
16. {
17.     return x*x - 5*x + y;
18. }
```





TÌM LỖI SAI TRONG CHƯƠNG TRÌNH SAU?

```
1. #include <iostream>
2. using namespace std;
3. int main ()
4. {
5.     int t; double vtoc;
6.     cout<<"Nhap gia tri van toc va thoi gian:";
7.     cin>>x>>y;
8.     cout<<"Quang duong= "<<quangduong(t, vtoc)<<endl;
9.     return 0;
10. }
11.
12. double quangduong(double v, int t);
13. {
14.     double s = v*t;
15.     return s;
16. }
```





BÀI TẬP

- **Bài 1:**Viết hàm tính diện tích hình tròn với tham số đầu vào là bán kính. Viết chương trình sử dụng hàm tính diện tích trên và hiển thị kết quả với bán kính $r=2$, $r=6$.

- **Bài 2:** Viết hàm tính độ dài đoạn thẳng khi biết tọa độ 2 điểm. Lập trình đọc vào tọa độ 3 điểm A, B, C thuộc tam giác. Tính các đoạn thẳng AB, AC, BC và đưa kết quả ra màn hình.



BÀI TOÁN

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string hoten, lop;
    cout<<"Nhap vao ho ten:";getline(cin, hoten);
    cout<<"Nhap vao lop:";cin>>lop;
    cout<<"Loi chao tieng viet";
    cout<<"Xin chao "<<hoten<<" ,Lop "<<lop<<" !"<<endl<<endl;
    cout<<"Loi chao tieng anh";
    cout<<"Hello "<<hoten<<" ,Class "<<lop<<" !";
    return 0;
}
```

BÀI TOÁN

```
#include <iostream>
#include <string>
using namespace std;
void hienthi(string hoten, string lop)
{
    cout<<"Loi chao tieng viet";
    cout<<"Xin chao "<<hoten<<" ,Lop "<<lop<<" !"<<endl<<endl;
    cout<<"Loi chao tieng anh";
    cout<<"Hello "<<hoten<<" ,Class "<<lop<<" !";
}

int main()
{
    string hoten, lop;
    cout<<"Nhap vao ho ten:";getline(cin, hoten);
    cout<<"Nhap vao lop:";cin>>lop;
    hienthi(hoten, lop);
    return 0;
}
```

THỦ TỤC (hàm không trả về giá trị)

(Xem mục 2.7 trong giáo trình)

- Dùng từ khóa **void** để thay cho kiểu trả về của hàm
- Trong thân hàm không có lệnh **return giá trị**
- Không được gọi trong câu lệnh ghi ra màn hình **cout**, hay gán giá trị cho biến

▪ Ví dụ:

```
#include <iostream>
using namespace std;
void trungBinh(double x, double y)
{
    double s = x + y ;
    cout<<s/2;
}
int main () {
    double x, y;
    cout<<"Nhap gia tri x, y:";
    cin>>x>>y;
    trungBinh(x,y); //Loi gọi ham trungBinh
    return 0;
}
```

TRƯỜNG HỢP HÀM KHÔNG CÓ THAM SỐ

- Không cần khai báo tham số trong định nghĩa hàm và khai báo nguyên mẫu hàm
- Ví dụ:

```
void hienThi()  
{  
    cout<<"Chao mung ban den voi khoa hoc!";  
}
```



TÌM LỖI SAI TRONG CHƯƠNG TRÌNH SAU?

```
1. #include <iostream>
2. using namespace std;
3. void hienthi(double x)
4. {
5.     cout<<"Gia tri vua nhap ="<<x<<endl;
6. }
7. int main ()
8. {
9.     double x, y, s;
10.    cout<<"Nhap x= ";cin>>x;
11.    cout<<"Nhap y= ";cin>>y;
12.    hienthi(x,y);
13.    cout<<hienthi(y);
14.    cout<<hienthi(x+y);
15.    return 0;
16. }
```





TÌM LỖI SAI TRONG CHƯƠNG TRÌNH SAU?

```
1. #include <iostream>
2. using namespace std;
3. double dientroTD(double, double );
4. int main ()
5. {
6.     double r1, r2;
7.     cout<<"Nhap r1= ";cin>>r1;
8.     cout<<"Nhap r2= ";cin>>r2;
9.     cout<<"Dien tro tuong duong=";
10.    dientroTD(r1,r2);
11.    return 0;
12. }
13. double dientroTD(double r1, double r2)
14. {
15.     double s;
16.     s = 1/(1/r1+1/r2);
}
```



TRUYỀN THAM TRỊ

(Xem mục 2.8 trong giáo trình)

- Tham số hình thức: là tham số khai báo trong phần khai báo hàm
- Tham số thực: là tham số được truyền vào trong lời gọi hàm
 - Truyền bằng tham trị
 - Truyền bằng tham chiếu
- **Truyền bằng tham trị:**
 - ✓ Tạo một bản sao giá trị của tham số gán cho tham số hình thức
 - ✓ Không làm thay đổi giá trị của biến truyền vào

TRUYỀN THAM TRỊ

(Xem mục 2.8 trong giáo trình)

▪ Ví dụ:

```
#include <iostream>
using namespace std;
void hoanVi(int x, int y)
{
    int temp =x;
    x=y; y=temp;
}
int main () {
    int x, y;
    cout<<"Nhap gia tri x, y:";
    cin>>x>>y;
    cout<<"Gia tri x, y truoc khi hoan doi "<<x<<" "<<y<<endl;
    hoanVi(x,y); //Loi goi ham hoan vi
    cout<<"Gia tri x, y sau khi hoan doi "<<x<<" "<<y<<endl;
    return 0;
}
```

```
Nhap gia tri x, y: 3 4
Gia tri x, y truoc khi hoan doi 3 4
Gia tri x, y sau khi hoan doi 3 4
```

TRUYỀN THAM CHIẾU

- Truyền bằng tham chiếu:

- ✓ Lời gọi hàm sẽ truyền trực tiếp tham số
- ✓ Làm thay đổi giá trị của biến truyền vào

- Nguyên mẫu hàm truyền tham chiếu:

```
kiutrave tenham (kieudulieu &, kieudulieu &,....);
```

- Định nghĩa hàm truyền tham chiếu:

```
kiutrave tenham (kieudulieu &thamso1, kieudulieu &thamso2,..)  
{  
    //các câu lệnh xử lý  
}
```

TRUYỀN THAM TRỊ

▪ Ví dụ:

```
#include <iostream>
using namespace std;
void hoanVi(int &x, int &y)
{
    int temp =x;
    x=y; y=temp;
}
int main () {
    int x, y;
    cout<<"Nhap gia tri x, y:";
    cin>>x>>y;
    cout<<"Gia tri x, y truoc khi hoan doi "<<x<<" "<<y<<endl;
    hoanVi(x,y); //Loi goi ham hoan vi
    cout<<"Gia tri x, y sau khi hoan doi "<<x<<" "<<y<<endl;
    return 0;
}
```

```
Nhap gia tri x, y: 3 4
Gia tri x, y truoc khi hoan doi 3 4
Gia tri x, y sau khi hoan doi 4 3
```

KHAI BÁO VỚI GIÁ TRỊ MẶC ĐỊNH

- Truyền giá trị mặc định khi định nghĩa hàm
- Giá trị mặc định được sử dụng khi không truyền tham số trong lời gọi hàm
- Định nghĩa hàm:

```
kieutrave tenham (kieudulieu thamso1 = giatri, kieudulieu thamso2 = giatri,..)
{
    //các câu lệnh xử lý
}
```

KHAI BÁO VỚI GIÁ TRỊ MẶC ĐỊNH

- Ví dụ:

```
#include <iostream>
using namespace std;
int phepChia(int a, int b=2)
{
    int r;
    r=a/b;
    return (r);
}
int main ()
{
    cout << phepChia(12)<<endl; // gọi hàm với giá trị mặc định
    cout << phepChia(20,4);
    return 0;
}
```

PHẠM VI CỦA BIẾN

▪ Biến cục bộ:

- Là những biến được khai báo và sử dụng bên trong một hàm, khối lệnh
- Được tạo ra khi thực hiện khối lệnh và bị hủy khi khối lệnh kết thúc

▪ Ví dụ:

```
int phepChia(int a, int b)
{
    int r;
    r=a/b;
    return r;
}

int phepNhan(int a, int b)
{
    int r;
    r=a*b;
    return r;
}
```

PHẠM VI CỦA BIẾN

- **Biến toàn cục:**

- Là những biến được sử dụng trên toàn bộ chương trình
- Khai báo bên ngoài các hàm

PHẠM VI CỦA BIẾN

▪ Ví dụ:

```
#include <iostream>
using namespace std;
double a, b, tb;
void nhap()
{
    cout<<"Nhap gia tri a va b:";
    cin>>a>>b;
}
void trungBinh()
{
    tb= (a+b)/2;
}
void ketQua()
{
    cout<<"Trung binh cong la:"<<tb;
}
int main()
{
    nhap();
    trungBinh();
    ketQua();
}
```

TỔNG HỢP

- **Khi viết hàm, xác định bài toán:**

- Cần bao nhiêu tham số đầu vào? Kiểu của tham số là gì?
- Hàm có trả về giá trị hay không?
- Gọi hàm ở đâu?
- Truyền giá trị nào cho hàm?



BÀI TẬP

Bài 1:Viết hàm tính diện tích tam giác ABC khi biết độ dài 3 cạnh. Viết chương trình nhập độ dài 3 cạnh tam giác, gọi hàm tính diện tích bên trên và in kết quả ra màn hình. (Dùng công thức Hêrông)



Bài 2: Viết hàm tính độ dài đoạn thẳng khi biết tọa độ 2 điểm sử dụng nguyên mẫu hàm sau:

void tinhdodai(float xa,float ya,float xb, float yb, float &AB);

Viết chương trình nhập tọa độ 3 điểm A, B, C, sử dụng hàm bên trên để tính các đoạn thẳng AB, AC, BC.



CHUẨN BỊ CHO BÀI SAU!

- **Làm việc với sâu:** *Mục 5.5 trong giáo trình*
- **Chương trình con chung và các kiểu chung:** *Mục 5.6 trong giáo trình*