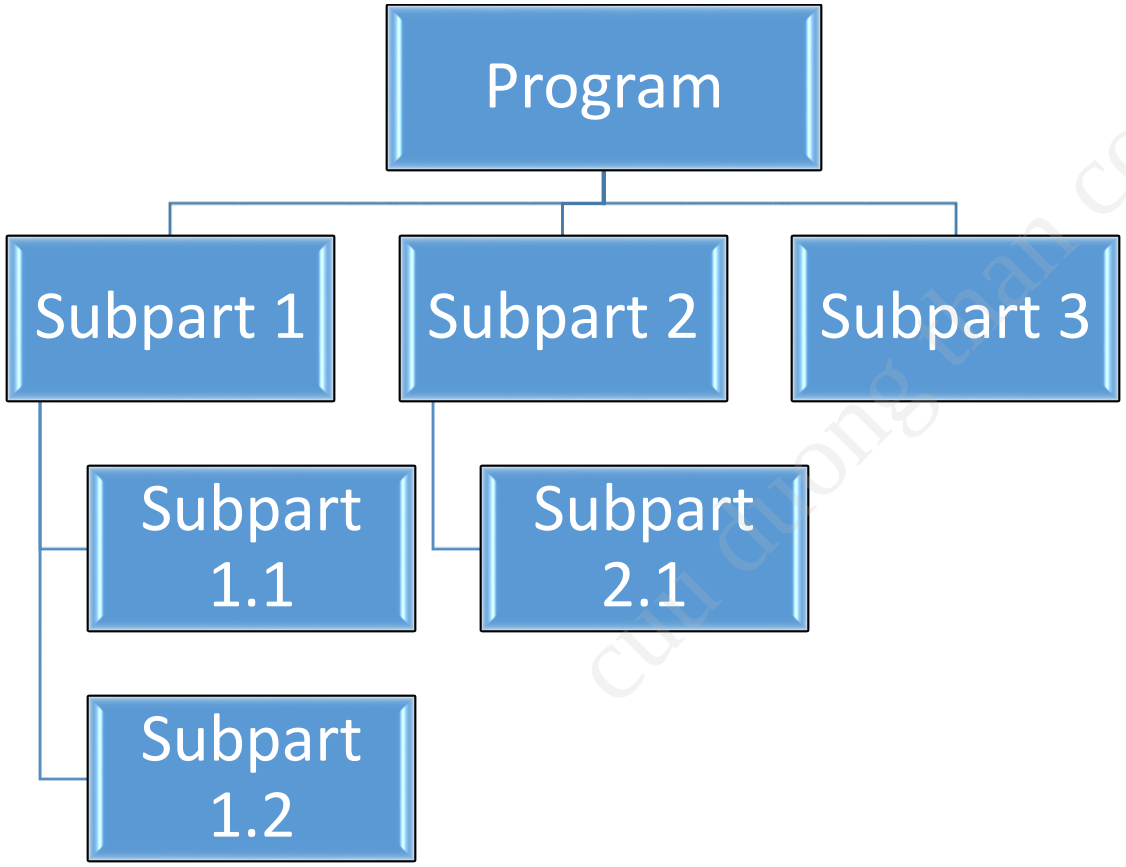
HÀM

Giới thiệu về hàm



- Thành phần con của một chương trình, mỗi thành phần thực hiện 1 chức năng nhất định.
- Tên gọi của thành phần con:
  - Thủ tục, chương trình con, phương thức
  - Với C++: hàm
- Xử lý: I-P-O
  - Input – Process – Output

Hàm sẵn có

- Các thư viện bao gồm nhiều hàm hữu ích
- Hai loại:
  - Hàm có giá trị trả về
  - Hàm không có giá trị trả về (void)
- Phải "#include" các thư viện phù hợp.
  - Ví dụ
    - <conio.h>
    - <iostream> (for cout, cin)

Sử dụng hàm sẵn có

- Có rất nhiều hàm toán học
  - Thư viện: <cmath>
  - Hầu hết đều có giá trị trả về
- Ví dụ: y = pow(x,2);
  - Thành phần:
    pow = tên hàm
    y = biến nhận giá trị trả về
    x, 2: tham số, giá trị ban đầu của hàm
  - I-P-O:
    - I = x, 2
    - P = tính lũy thừa
    - O = x^2

Lời gọi hàm

- Ví dụ: y = pow(x,2);
  - Biểu thức pow(x,2) là lời gọi hàm.
  - Tham số của hàm: giá trị cố định, biến, biểu thức
    - Ví dụ: y = pow(x+2,2);
  - Lời gọi hàm có thể là thành phần của một biểu thức
    - z = pow(x, 2)/10;

Các hàm sẵn có khác

- #include <cstdlib>
  - Thư viện chứa các hàm như:
    - abs()
    - labs()
  - fabs() trong thư viện <cmath>!
    - Có thể nhầm lẫn

NAME	DESCRIPTION	TYPE OF ARGUMENTS	TYPE OF VALUE RETURNED	EXAMPLE	VALUE	LIBRARY HEADER
sqrt	Square root	double	double	sqrt(4.0)	2.0	cmath
pow	Powers	double	double	pow(2.0,3.0)	8.0	cmath
abs	Absolute value for int	int	int	abs(-7) abs(7)	7 7	cstdlib
labs	Absolute value for long	long	long	labs(-70000) labs(70000)	70000 70000	cstdlib
fabs	Absolute value for double	double	double	fabs(-7.5) fabs(7.5)	7.5 7.5	cmath
ceil	Ceiling (round up)	double	double	ceil(3.2) ceil(3.9)	4.0 4.0	cmath
floor	Floor (round down)	double	double	floor(3.2) floor(3.9)	3.0 3.0	cmath
exit	End program	int	void	exit(1);	None	cstdlib
rand	Random number	None	int	rand( )	Varies	cstdlib
srand	Set seed for rand	unsigned int	void	srand(42);	None	cstdlib

Hàm tự định nghĩa

- Lập trinh viên tự xây dựng hàm
- Nguyên tắc xây dựng:
  - Chia để trị
  - Dễ đọc
  - Dễ dùng
- Định nghĩa hàm có thể
  - Cùng file với hàm main()
  - Khác file với main

Sử dụng hàm

1. Khai báo hàm
   - Thông tin dành cho trình biên dịch
   - Biên dịch phù hợp
2. Định nghĩa hàm
   - Thực thi thực sự: xây dựng hàm chi tiết
3. Gọi hàm
   - Chuyển quyền điều khiển cho hàm

Khai báo hàm

- Nguyên mẫu hàm (function prototoype)
- Khai báo thông tin cho trình biên dịch
  - Cú pháp:
    <Kiểu_dữ_liệu> Tên_hàm (Kiểu_dữ_liệu_1 tham_số_1, …);
  - Ví dụ:
    double tinhDTB (double diem1, double diem2);
- Vị trí:
  - Trước hàm main()
  - Các hàm ngang hàng nhau, không có trường hợp hàm này lồng vào hàm kia

Định nghĩa hàm

- Ví dụ:

```
double tinhDTB (double diem1, double diem2)
{
        return (diem1 + diem2)/2 ;
}
```

- Tham số hình thức trong định nghĩa hàm: Nơi lưu trữ dữ liệu
- Lệnh return: Chuyển dữ liệu trả về cho "người gọi hàm"

Lời gọi hàm

- Tương tự gọi hàm đã định nghĩa sẵn
  double dtb = tinhDTB(d1, d2)
- tinhDTB trả về giá trị kiểu double
  - Giá trị trả về được gán vào biến dtb
- Tham số thực: d1, d2
  - Tham số: giá trị không đổi, biến, biểu thức
  - Trong lời gọi hàm, các tham số thường được gọi là tham số thực sự
    - Bởi vì chúng chứa dữ liệu thực sự được truyền

Tham số

- Tham số hình thức
  - Khai báo hàm
  - Định nghĩa hàm
- Tham số thực sự
  - Trong lời gọi hàm

Hàm trả về giá trị logic Boolean

- Hàm có thể trả về bất kỳ kiểu dữ liệu nào
  - Khai báo hàm
    bool dat(int diem);
  - Định nghĩa hàm
    bool dat (int diem)
    {
    return (diem>=4)&&(rate<=10);
    }
  - Trả về "true" hoặc "false"
  - Lời gọi hàm từ hàm khác:

```
if (dat(diem))
        cout << “Dat";
else
        cout << “Rot”;
```

Hàm có kiểu trả về void

- Tương tự với hàm trả về một giá trị bất kỳ
- Khai báo kiểu trả về là void
- Ví dụ:
  - Khai báo hàm:

    void showResults(double fDegrees, double cDegrees);

    - Kiểu trả về: "void"
    - Không trả về giá trị nào hết

Định nghĩa hàm có kiểu trả về void

- Định nghĩa hàm:

```
void showResults(double fDegrees, double cDegrees)
{
        cout.setf(ios::fixed);
        cout.setf(ios::showpoint);
        cout.precision(1);
        cout   << fDegrees
                << " degrees fahrenheit equals \n"
                << cDegrees << " degrees celsius.\n";
}
```

- Không có lệnh return hoặc sử dụng hàm return;

Gọi hàm có kiểu trả về void

- Giống như hàm các hàm void đã định nghĩa sẵn
- Từ một hàm khác, như main():
  - showResults(degreesF, degreesC);
  - showResults(32.5, 0.3);
- Không có câu lệnh gán vì không trả về giá trị
- Tham số thực sự (degreesF, degreesC)
  - Truyền cho hàm
  - Hàm được gọi chỉ việc thực thi với dữ liệu truyền vào

Lệnh return

- Trả quyền điều khiển về hàm đang gọi
  - Với các hàm có kiểu trả về khác void, phải có ít nhất 1 câu lệnh return
- Lệnh return có thể có hoặc không có trong hàm có kiểu trả về là void
  - return;

main()

- main() là hàm đặc biệt:
  - Chỉ có một và chỉ một hàm main tồn tại trong chương trình.
- Ai gọi hàm main?
  - Hệ điều hành
  - Trả về kiểu int

Nguyên mẫu (prototype):

- Nguyên tắc: hàm phải được định nghĩa trước khi gọi.
- Tuy nhiên, trong lập trình thì hàm main() có thể được đặt trước các hàm khác. Nếu hàm main() có gọi các hàm khác thì trình biên dịch sẽ báo lỗi hàm này chưa định nghĩa.
- Do vậy, cần khai báo nguyên mẫu hàm trước hàm main().
- Cú pháp khai báo nguyên mẫu của hàm giống như định nghĩa hàm, chỉ khác là kết thúc bằng dấu ; và không có thân hàm.

Ví dụ:

```
double larger(double x, double y);
double compareThree(double x, double y, double z);
```

Tham số

- Hai cách truyền tham số
  - Tham trị
    - Giá trị của tham số được truyền
  - Tham biến
    - Địa chỉ của tham số thực được truyền

Tham trị

- Sao chép giá trị của tham số thực
- Xem như biến cục bộ bên trong hàm
- Nếu có thay đổi, chỉ thay đổi giá trị cục bộ; hàm không có quyền thay đổi giá trị của tham số thực.

Tham trị

```
1  #include <iostream>
2  using namespace std;
3  double tinhDTB(double diem1, double diem2);
4  int main()
5  {
6      double diem1=2, diem2=3;
7      double dtb = tinhDTB(diem1, diem2);
8      cout<<"Diem1 = "<<diem1<<endl;
9      cout<<"Diem2 = "<<diem2<<endl;
10     cout<<"Diemtb = "<<dtb;
11 }
12 double tinhDTB(double diem1, double diem2)
13 {
14     double dtb = (diem1+diem2)/2;
15     diem1 = diem1*2;
16     return dtb;
17 }
```

Tham chiếu

- Tham chiếu tới tham số thực sự
  - Trỏ tới vị trí vùng nhớ của tham số thực sự
  - Thay đổi giá trị tại vùng nhớ này => thay đổi giá trị của tham số thực sự
- Sử dụng: tên_kiểu& tên_biến

Tham chiếu

```
1  #include <iostream>
2  using namespace std;
3  double tinhDTB(double &diem1, double diem2);
4  int main()
5  {
6      double diem1=2, diem2=3;
7      double dtb = tinhDTB(diem1, diem2);
8      cout<<"Diem1 = "<<diem1<<endl;
9      cout<<"Diem2 = "<<diem2<<endl;
10     cout<<"Diemtb = "<<dtb;
11 }
12 double tinhDTB(double &diem1, double diem2)
13 {
14     double dtb = (diem1+diem2)/2;
15     diem1 = diem1*2;
16     return dtb;
17 }
```

Phạm vi của biến (scope) và biến static

- *Biến được khai báo trong hàm nào chỉ có tác dụng trong hàm đó (local identifier)*
- *Biến được khai báo ngoài hàm sẽ có tác dụng đối với tất cả các hàm (global identifier)*
- *Biến khai báo trong hàm được tự động xóa đi khi kết thúc hàm, do đó được gọi là biến tự động (automatic).*
- Muốn một biến khai báo trong hàm nhưng vẫn giữ lại giá trị khi kết thúc hàm thì dùng thêm từ khóa static

Hàm với đối số mặc định:

- Nếu gọi hàm mà thiếu đối số, giá trị mặc định sẽ được dùng thay.

Ví dụ:

```
void funcExp(int x, int y, double t, char z = 'A', int u = 67,
             char v = 'G', double w = 78.34);
```

- Với khai báo hàm như trên, hàm có thể được gọi với ít nhất là 3 đối số. Các tham số còn lại dùng giá trị mặc định.

Xuất nhập file

- File là công cụ cơ bản để lưu dữ liệu lên đĩa khi chương trình thực hiện xong.
- File gồm 2 loại: file văn bản và file nhị phân. File nhị phân chứa các ký tự ASCII và có thể được đọc bởi bất kỳ trình soạn thảo nào. File nhị phân chứa mã nhị phân theo cấu trúc. Chỉ có chương trình biết cấu trúc của file thì mới đọc được.
- Để ghi dữ liệu vào file hoặc đọc dữ liệu từ file cần thực hiện các công việc sau:

-Khai báo #include <fstream>

-Khai báo các biến stream

-Mở file và liên kết file vừa mở với biến stream.

-Đọc, ghi file dùng các toán tử >> và <<

-Đóng file.

```cpp
#include <fstream>

//Add additional header files you use

using namespace std;

int main()
{
        //Declare file stream variables such as the following
    ifstream inData;
    ofstream outData;
    .
    .
    .

        //Open the files
    inData.open("prog.dat");  //open the input file
    outData.open("prog.out"); //open the output file

        //Code for data manipulation

        //Close files
    inData.close();
    outData.close();

    return 0;
}
```