

KỸ THUẬT LẬP TRÌNH C/C++ C++

Thi-Lan Le

Thi-Lan.Le@mica.edu.vn; lan.lethi1@hust.edu.vn

[Webpage: http://www.mica.edu.vn/perso/Le-Thi-Lan](http://www.mica.edu.vn/perso/Le-Thi-Lan)

Các nội dung chính

1. Các đặc điểm mới của C++ so với C
2. Các khái niệm cơ bản của lập trình hướng đối tượng

1. Các đặc điểm mới của C++ so với C

- ◆ C++ bổ sung khả năng lập trình hướng đối tượng (HĐT) với các khái niệm và các thành phần mới như: lớp, đối tượng, sự che dấu thông tin, sự kế thừa, sự đa hình,...Điều đó tạo cho C++ khả năng tổ chức chương trình theo cả phương pháp lập trình hướng chức năng và hướng đối tượng-một ngôn ngữ lập trình lai.
- ◆ Cho phép định nghĩa **chồng các hàm**, tức là các hàm có thể trùng tên với nhau trong cùng phạm vi định nghĩa và sử dụng.
- ◆ Cho phép tạo các mẫu (**template**): mẫu hàm và mẫu lớp.
- ◆ Cho phép **các hàm có các tham số nhận giá trị mặc định**.
- ◆ Bổ sung các lớp nhập/xuất mới nhằm đơn giản hoá các thao tác nhập/ xuất, tăng tính mở của các thao tác này khi phải nhập/xuất dữ liệu với các kiểu dữ liệu hay các đối tượng mới.

1. Các đặc điểm mới của C++ so với C

- ◆ Bổ sung từ khóa **bool** và các giá trị (**true**, **false**) cho kiểu logic
- ◆ Bổ sung loại hàm con **inline** nhằm tăng tốc độ thực hiện các hàm con.
- ◆ Bổ sung các hàm cấp phát và giải phóng vùng nhớ động mới là **new** và **delete**.
- ◆ Bổ sung đối tượng, **tham số kiểu tham chiếu** giúp cho việc sử dụng các tham số của các hàm con được dễ dàng và hiệu quả hơn.
- ◆ Bổ sung loại chú thích **mới-chú thích trên một dòng: //**
- ◆ Cho phép đan xen giữa khai báo các đối tượng dữ liệu và các lệnh xử lý.

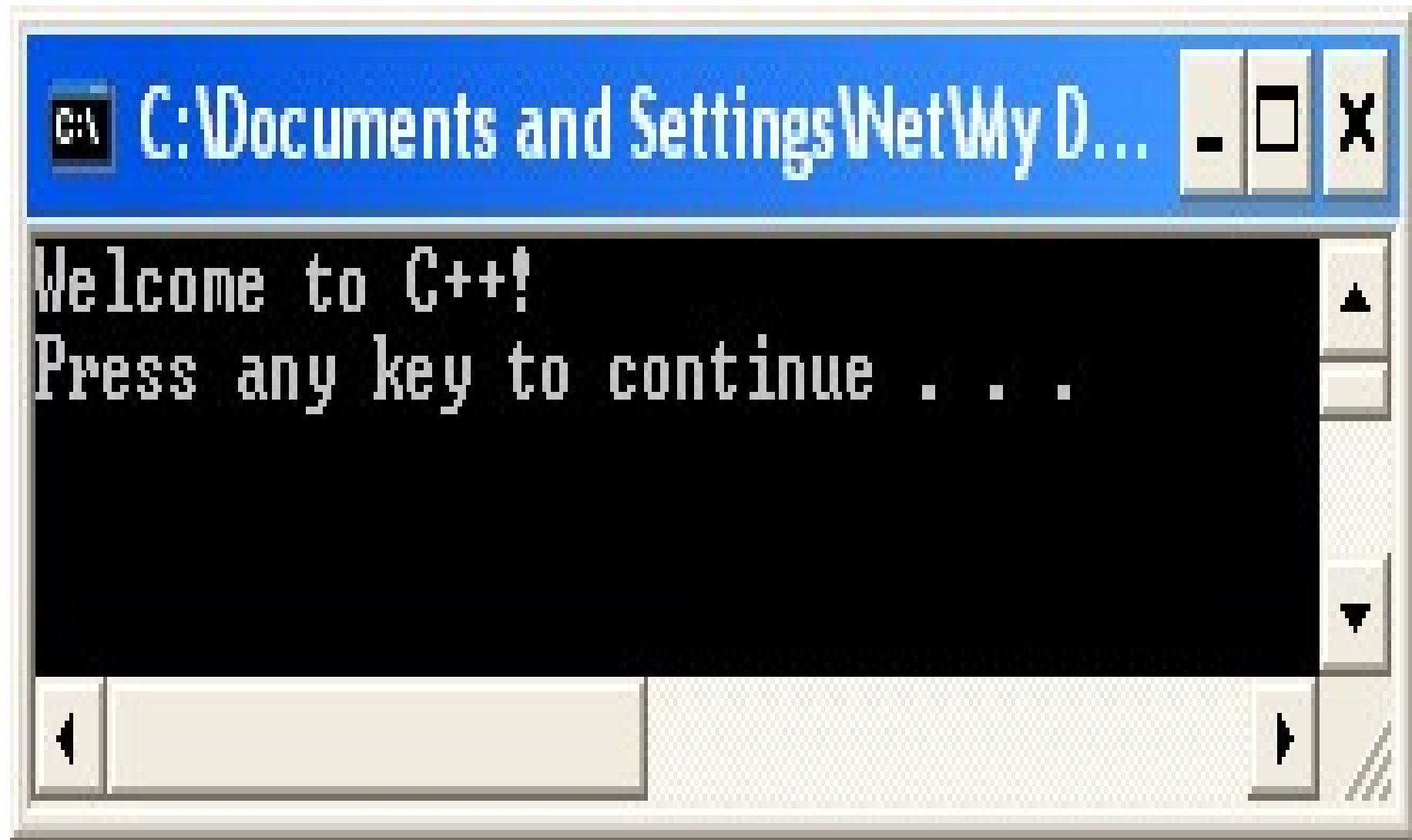
Chương trình C++ đầu tiên

◆ Program 8.1: In ra màn hình dòng “Welcome to C++!”

```
#include <cstdlib> //Cho EXIT_SUCCESS;
#include <iostream> //tệp thư viện nhập/xuất chính trong C++
using namespace std; //Khai báo không gian tên mặc định

int main()
{
    cout<<"Welcome to C++!"<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Kết quả chạy Program 8.1



Khả năng nhập/xuất mới

- ◆ Khai báo thư viện sử dụng: **#include** <iostream>
- ◆ Nhập dữ liệu: sử dụng đối tượng **cin** của lớp istream và phép toán >>
- ◆ Xuất dữ liệu: Sử dụng đối tượng **cout** của lớp ostream và phép toán <<
- ◆ Ví dụ:

```
int i;  
cout << "Please enter an integer value: ";  
cin >> i;  
cout << "The value you entered is " << i;
```

Khả năng nhập/xuất mới

◆ Nhập liệu chuỗi ký tự/xâu

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string mystr;
    cout << "What's your name? ";
    getline (cin, mystr);
    cout << "Hello " << mystr << ".\n";
    cout << "What is your favorite team? ";
    getline (cin, mystr);
    cout << "I like " << mystr << " too!\n";
    return 0;
}
```


Tham chiếu

- ◆ **Khái niệm:** tham chiếu là một tên gọi mới của một vùng nhớ được cấp phát cho một đối tượng.

```
int n=10;  
int &m = n; //m là biến tham chiếu đến n  
m = 20;    //tương đương n=20
```

Tham chiếu

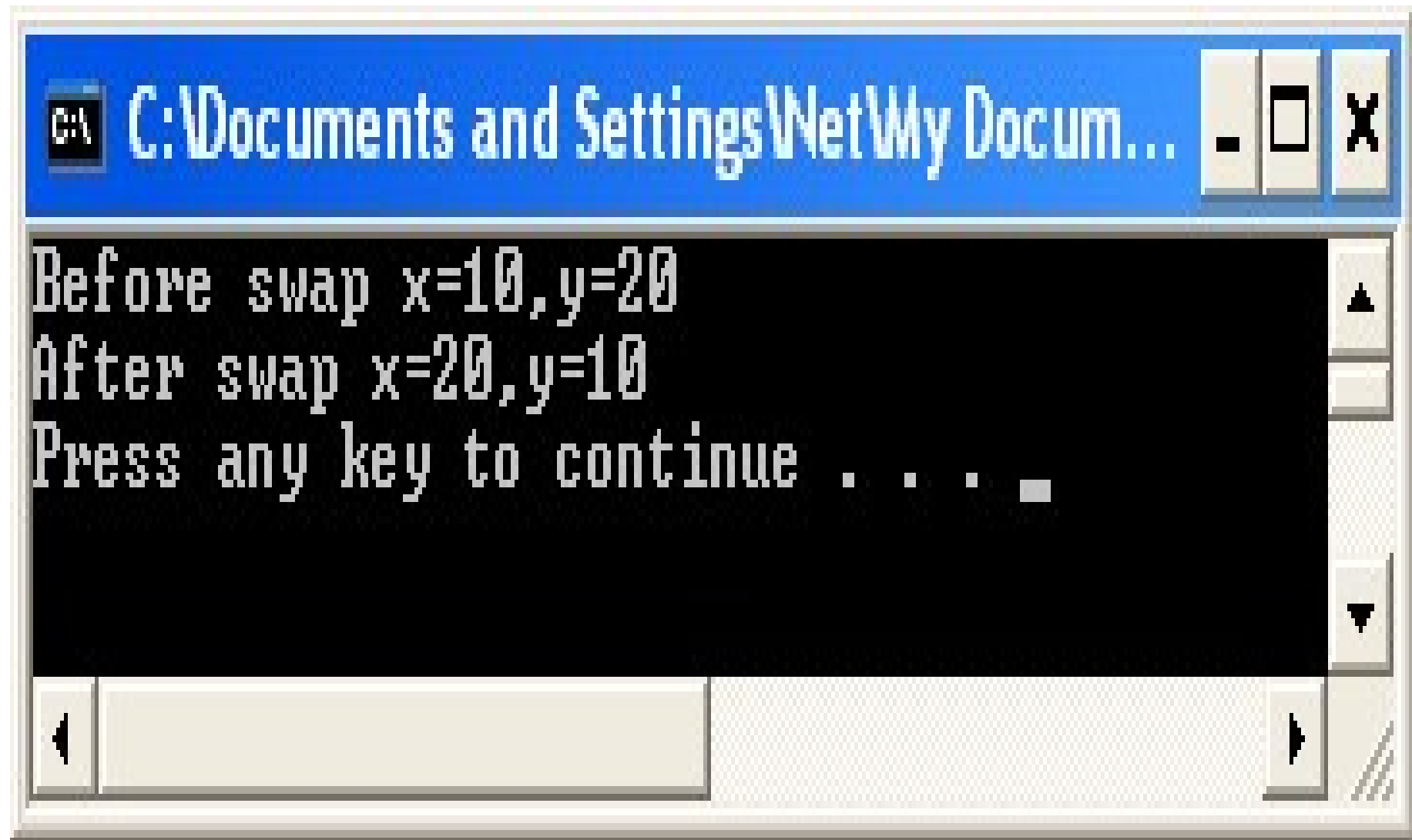
- ◆ **Vai trò:** thường được sử dụng trong tham số của hàm để thực hiện truyền tham biến (trong C chỉ cho phép truyền tham trị).

```
void exchange(int &a, int &b)
{
    int c=a;
    a=b;
    b=c;
}

int main(){
    int x=10, y=20;
    cout<<"Before swap x="<<x<<",y="<<y<<endl;

    exchange(x,y);
    cout<<"After swap x="<<x<<",y="<<y<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Kết quả chạy chương trình



A screenshot of a Windows XP command prompt window. The title bar is blue and contains the text "C:\Documents and Settings\Net\My Docum..." followed by minimize, maximize, and close buttons. The command prompt area has a black background with white text. It displays three lines of output: "Before swap x=10,y=20", "After swap x=20,y=10", and "Press any key to continue . . .". The text is in a monospaced font. The window has a standard Windows XP border and a taskbar at the bottom.

```
Before swap x=10,y=20
After swap x=20,y=10
Press any key to continue . . .
```

Hàm với tham số nhận giá trị mặc định

- ◆ Là mở rộng trong C++ cho phép khi gọi một hàm con, ta có thể bỏ qua một số tham số của nó, khi đó các tham số này sẽ nhận các giá trị mặc định mà đã được quy định trước đó khi khai báo hàm con này.

Chương trình minh họa

//Khai báo hàm với giá trị ngầm định

```
void HamND(int a = 10, int b = 20);
```

```
int main()
```

```
{
```

```
    cout<<"Goi ham khong co tham so:"<<endl;
```

```
    HamND(); //Gọi hàm với giá trị ngầm định
```

```
    cout<<"Goi ham co 1 tham so:"<<endl;
```

```
    HamND(30);
```

```
    cout<<"Goi ham co 2 tham so:"<<endl;
```

```
    HamND(30,40);
```

```
    system("PAUSE");
```

```
    return EXIT_SUCCESS;
```

```
}
```

//Định nghĩa hàm với giá trị ngầm định

```
void HamND(int a, int b)
```

```
{
```

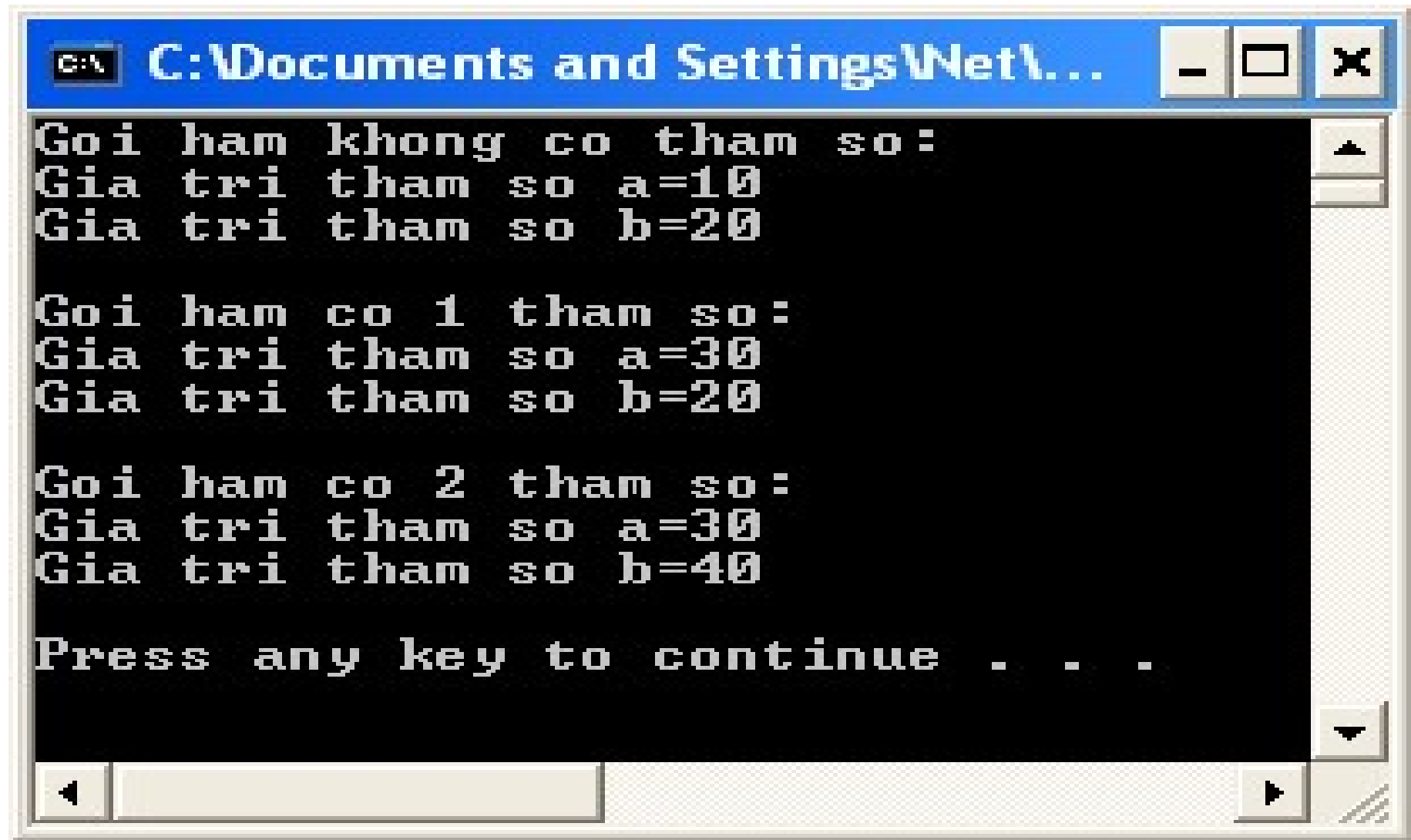
```
    cout<<"Gia tri tham so a="<<a<<endl;
```

```
    cout<<"Gia tri tham so b="<<b<<endl;
```

```
    cout<<endl;
```

```
}
```

Kết quả chạy chương trình



```
C:\Documents and Settings\Net\...  
Goi ham khong co tham so:  
Gia tri tham so a=10  
Gia tri tham so b=20  
  
Goi ham co 1 tham so:  
Gia tri tham so a=30  
Gia tri tham so b=20  
  
Goi ham co 2 tham so:  
Gia tri tham so a=30  
Gia tri tham so b=40  
  
Press any key to continue . . .
```

Định nghĩa chồng hàm

- ◆ Là khả năng cho phép định nghĩa lại một hàm nhiều lần với cùng một tên hàm, nhưng với các tham số khác nhau (có thể khác nhau về số lượng tham số và/hoặc kiểu dữ liệu của tham số)

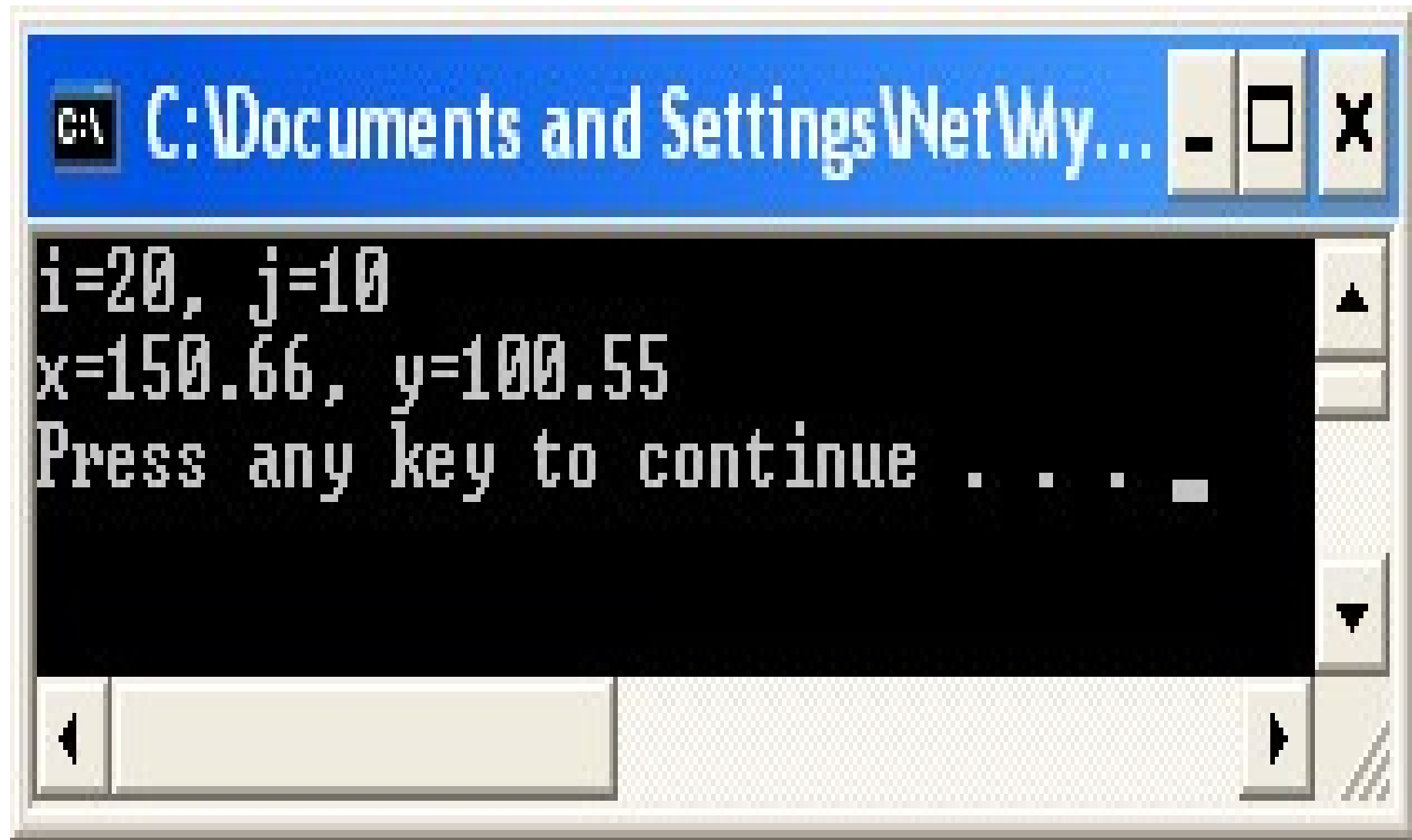
Chương trình ví dụ

//Định nghĩa chồng hàm swap

```
void swap(int &a, int &b){  
    int c=a;  
    a=b;  
    b=c;  
}  
void swap(float &a, float &b){  
    float c=a;  
    a=b;  
    b=c;  
}
```

```
int main()  
{  
    int i=10, j=20;  
    float x=100.55, y=150.66;  
    swap(i,j); //Gọi hàm swap(int, int)  
    cout<<"i="<<i<<"", j="<<j<<endl;  
    swap(x,y); //Gọi hàm swap(float, float)  
    cout<<"x="<<x<<"", y="<<y<<endl;  
  
    system("PAUSE");  
    return EXIT_SUCCESS;  
}
```


Kết quả chạy chương trình



A screenshot of a Windows XP command prompt window. The title bar is blue and contains the text "C:\Documents and Settings\Net\My..." followed by standard window control buttons (minimize, maximize, close). The command prompt icon is visible on the left. The main area is black with white text. The output shows two lines of variable assignments: "i=20, j=10" and "x=150.66, y=100.55". Below these, it says "Press any key to continue . . .". At the bottom, there is a scroll bar and a small icon on the left.

```
C:\Documents and Settings\Net\My...  
i=20, j=10  
x=150.66, y=100.55  
Press any key to continue . . .
```

Tìm hiểu về namespace

- ◆ **namespace** (không gian tên): là công cụ cho phép quản lý sự xung đột về tên của các thành phần của chương trình như tên biến, tên lớp, tên hàm, v.v
- ◆ Trong một không gian tên, các biến, hàm ... phải có tên duy nhất → không được sử dụng tên trùng nhau. Muốn có các hàm, biến có cùng tên → định nghĩa trong các namespace khác nhau.

Tìm hiểu về namespace

```
namespace A
{
    int x = 5;
    void printX()
    {
        // function statements goes here
        cout<<x<<endl;
    }
}
```

```
namespace B
{
    int x=10;
    void printX()
    {
        // function statementsgoes here
        cout<<x<<endl;
    }
}
```

```
int main()
{
    A::printX() ;
    B::printX();

    return 0;
}
```

Tìm hiểu về namespace

- ◆ using namespace std
- ◆ **#include <iostream>**
 - Std viết tắt là Standard là không gian tên chuẩn
 - Vì cout và cin được định nghĩa trong không gian tên std và nằm trong iostream
 - endl = '\n': kí tự xuống dòng

Cấp phát động bộ nhớ

- ◆ Trong C++ cung cấp 2 thao tác giúp cho việc quản lý bộ nhớ động trong quá trình chạy chương trình:
 - **new**: cấp phát bộ nhớ động
 - **delete**: giải phóng vùng nhớ động đã cấp phát trước đó
- ◆ Với khả năng quản lý bộ nhớ động, ta có thể tạo ra các đối tượng mà kích thước lưu trữ có tính động không biết trước, mà thường thay đổi liên tục trong quá trình chạy CT.

Cú pháp cho **new** và **delete**

◆ Cú pháp lệnh **new**:

void * new T [N];

- Trong đó: T: kiểu dữ liệu cho mỗi phần tử
 N: số lượng phần tử
- Ý nghĩa: cấp phát N ngăn nhớ liên tục để chứa N phần tử có kiểu dữ liệu T, rồi trả về địa chỉ của ngăn đầu tiên nếu cấp phát thành công. Trái lại trả về NULL

Cú pháp cho **new** và **delete**

◆ Cú pháp lệnh **delete**:

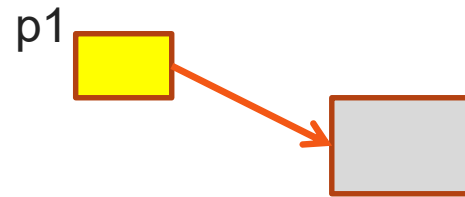
Cú pháp 1: **delete** p;

Cú pháp 2: **delete** [] p;

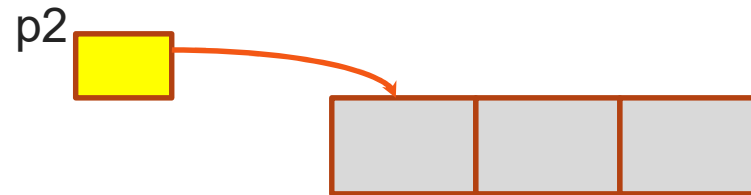
- *Trong đó:* p là con trỏ đang trỏ vào vùng nhớ đã được cấp phát động trước đó.
- *Ý nghĩa:* cú pháp 1 dùng để giải phóng một ngăn nhớ đang được trỏ bởi p. Kích thước ngăn nhớ này chính là kích thước của kiểu dữ liệu mà p sẽ trỏ đến. Còn cú pháp 2 dùng để giải phóng một mảng các ngăn nhớ đang được trỏ bởi p.

Ví dụ về các lệnh new và delete

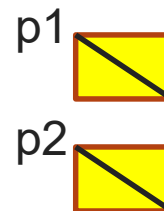
```
int * p1;  
p1 = new int;
```



```
int * p2;  
p2 = new int [3];
```



```
delete p1;  
delete [] p2;
```



Làm việc với con trỏ

◆ Các hàm new và delete

- ❑ `p = (int*) malloc(n * sizeof(int));`
- ❑ `p = new int [n];`
- ❑ `free (p);`
- ❑ `delete [] p;`

◆ Con trỏ 2 chiều:

◆ Xin cấp phát con trỏ 2 chiều:

```
int** a= new int*[m];  
for(int i=0;i<m; i++)  
a[i]= new int[n];
```

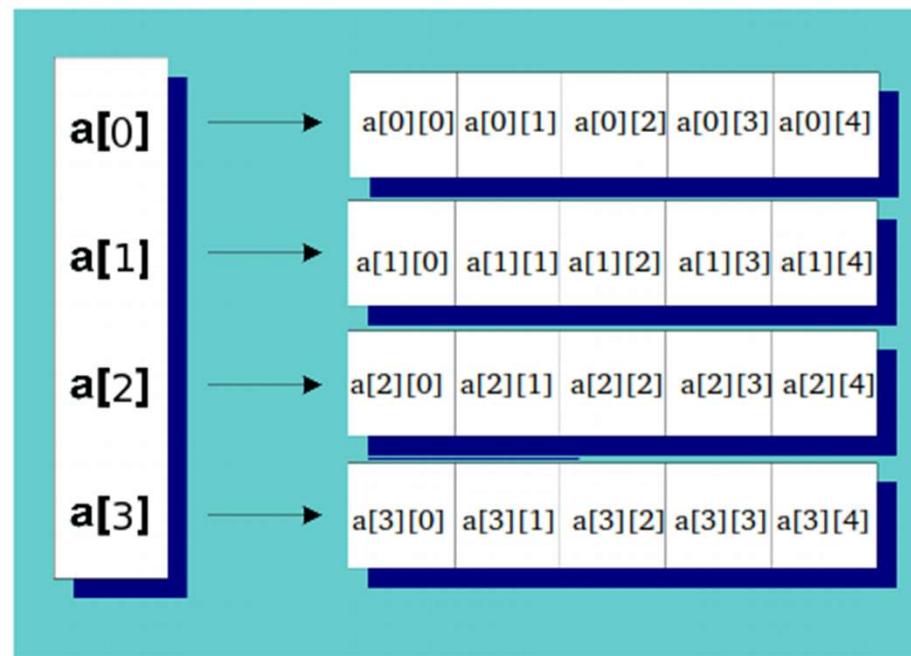
Làm việc với con trỏ

◆ Giải phóng con trỏ 2 chiều:

```
for(int i=0;i<m;i++)
```

```
delete [] a[i];
```

```
delete[] a;
```



2. Các khái niệm cơ bản của lập trình hướng đối tượng (object-oriented programming)

- ◆ Đối tượng và lớp (object and class)
- ◆ Một số đặc điểm và tính chất của lập trình hướng đối tượng

Đối tượng và lớp

◆ Đối tượng:

- Là thành phần cơ bản nhất một chương trình theo kiểu hướng đối tượng, biểu diễn cho một đối tượng của bài toán
- Là sự kết hợp gắn kết của các dữ liệu của đối tượng (thuộc tính) và các thao tác xử lý cần thiết trên các đối tượng dữ liệu đó (phương thức)
- Ví dụ:

Hình tròn A
Bán kính $r = 2$
TínhChuVi() TínhDiệnTích()

Hình tròn B
Bán kính $r = 5$
TínhChuVi() TínhDiệnTích()

Hình chữ nhật C
Chiều rộng $a = 2$ Chiều dài $b = 3$
TínhChuVi() TínhDiệnTích()

Đối tượng và lớp

◆ Lớp

- Là sự khái quát hóa các đối tượng cùng loại
- Có ý nghĩa đối với đối tượng tương tự như ý nghĩa của kiểu dữ liệu đối với các đối tượng dữ liệu, là cho phép dễ dàng tạo ra nhiều đối tượng cùng một kiểu (chung các kiểu dữ liệu và các phương thức)

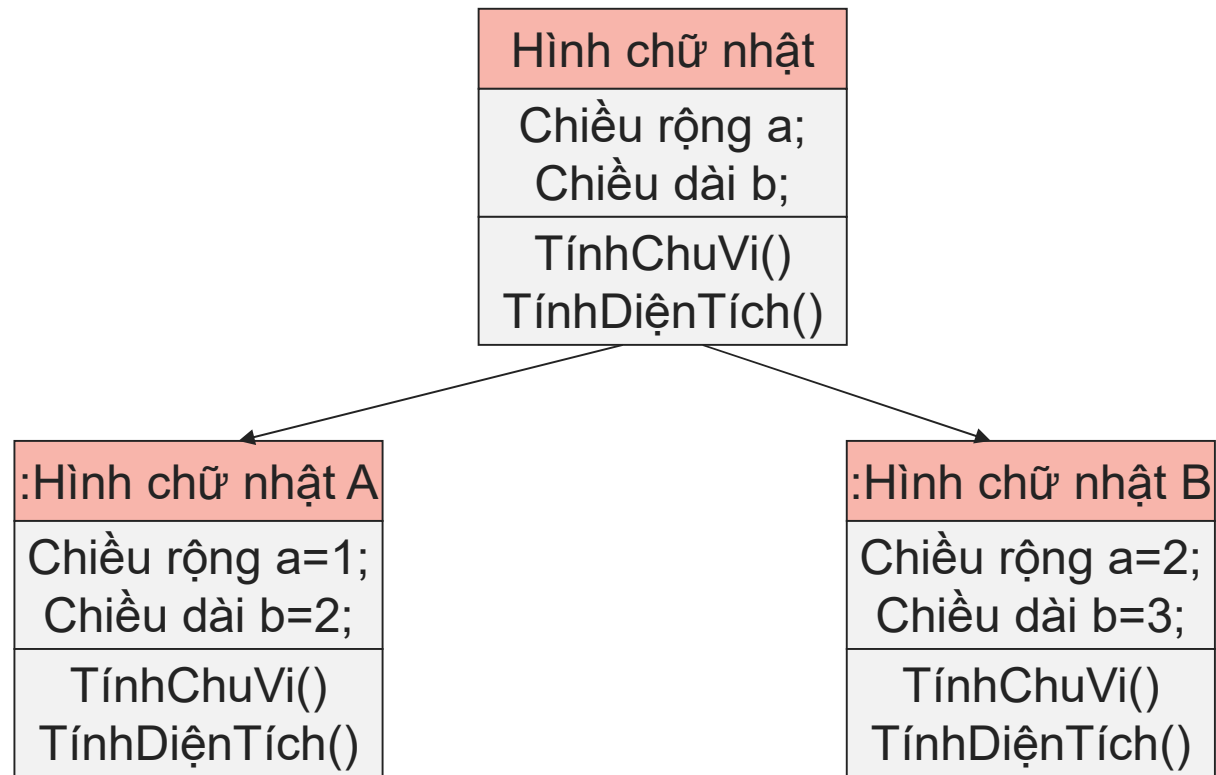
Hình tròn
Bán kính r
TínhChuVi() TínhDiệnTích()

Hình chữ nhật
Chiều rộng a; Chiều dài b;
TínhChuVi() TínhDiệnTích()

PT bậc 2
a; b; c;
TínhDelta() TínhNghiem()

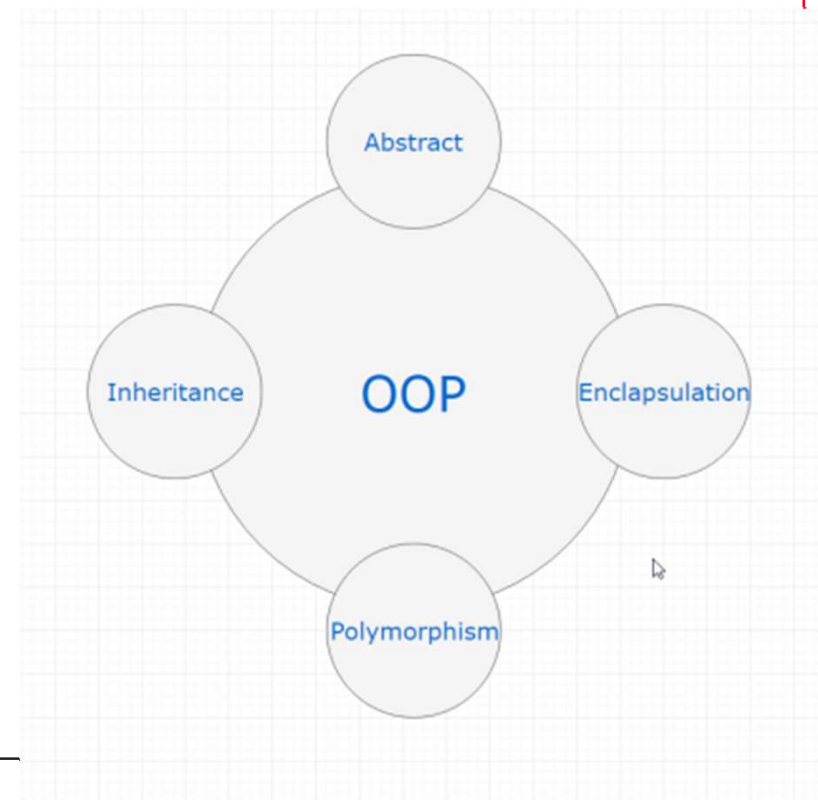
Quan hệ giữa đối tượng và lớp

- ◆ Đối tượng còn được gọi là *thể hiện* (instance) của một lớp



Đặc điểm của lập trình hướng đối tượng

- ◆ Tính trừu tượng (Abstraction)
- ◆ Sự che dấu các thành phần của lớp (còn gọi là sự đóng gói, encapsulation)
- ◆ Sự kế thừa (inheritance)
- ◆ Sự đa hình (polymorphism)



Bài tập

- ◆ Câu 1: Viết một đoạn chương trình cho phép người dùng nhập vào một mảng 2 chiều với số hàng và số cột do người dùng đưa ra sau đó viết hàm tìm số phần tử âm của ma trận được đưa vào
- ◆ Câu 2: Viết một đoạn chương trình cho phép người dùng. Lưu trữ ma trận ở dạng con trỏ hai chiều, sử dụng các hàm của C++:
 - Nhập vào một mảng 2 chiều với số hàng và số cột do người dùng đưa ra
 - Đếm số phần tử âm của ma trận được đưa vào.