

Chương 01.4: Vào/ra dữ liệu với C++

I. Lệnh vào/ra dữ liệu

II. Định dạng dữ liệu đưa ra

III. Một chương trình C++ đơn giản

I. Lệnh vào ra dữ liệu

1. Khai báo thư viện vào/ra dữ liệu

2. Lệnh đưa dữ liệu ra màn hình

3. Lệnh lấy dữ liệu vào từ bàn phím

1. Khai báo thư viện chương trình vào/ra dữ liệu

✧ Để có thể sử dụng các lệnh vào/ra dữ liệu với bàn phím, màn hình của C++ khi lập trình trên DOS/Linux ta phải khai báo sử dụng thư viện hàm:

```
#include<iostream>
```

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

✧ Để có thể sử dụng các lệnh vào/ra dữ liệu với tệp văn bản của C++ khi lập trình trên DOS/Linux ta phải khai báo sử dụng thêm thư viện hàm:

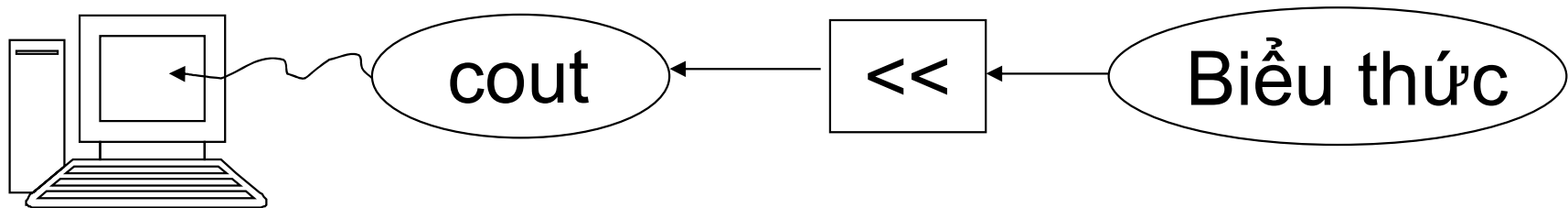
```
#include<fstream>
```

2. Lệnh đưa dữ liệu ra màn hình/tệp

✧ Để đưa dữ liệu ra màn hình ta dùng lệnh sau:

```
cout<<Biểu thức;
```

trong đó cout (đọc là C Out) là một đối tượng của C++ gắn với màn hình máy tính, << là toán tử xuất (“đưa tới”). Toán tử << sẽ đưa giá trị bên phải nó tới màn hình.



2. Đưa dữ liệu ra màn hình (tiếp)

- ✧ Có thể dùng một lệnh để đưa nhiều giá trị ra màn hình. Lệnh này được viết như sau:

```
cout<<Biểu thức 1<<.....<<Biểu thức N;
```

Khi đó giá trị của các biểu thức sẽ được đưa ra liên tiếp nhau.

- ✧ Khi đưa dữ liệu ra màn hình, muốn đặt con trỏ màn hình xuống đầu dòng tiếp theo ta phải đưa ra ký tự xuống dòng '\n' hoặc tác tử endl

```
cout<<Biểu thức<<'\n';
```

```
cout<<Biểu thức<<endl;
```

- ✧ *Ví dụ:* `cout<<a<<c+b<<'\n'; cout<<100;`

Đưa dữ liệu ra tệp văn bản

✧ Khai báo tệp đưa dữ liệu ra gắn với một tên tệp:

```
ofstream fout("Tên tệp");
```

Ví dụ: `ofstream fout("tamgiac.txt");`

✧ Ghi dữ liệu ra tệp fout giống như đưa dữ liệu ra màn hình cout:

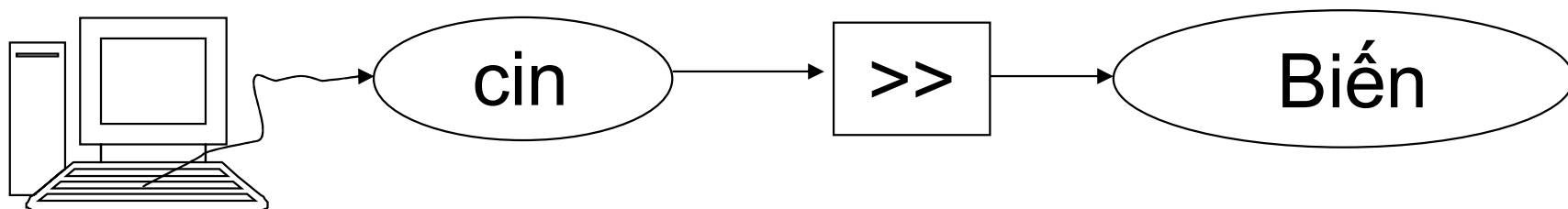
Ví dụ: `fout<<100<<" " <<a+b;`

3. Lệnh lấy dữ liệu vào từ bàn phím/tệp

✧ Để lấy dữ liệu từ bàn phím vào biến ta dùng lệnh sau: `cin>>Một biến;`

trong đó `cin` (đọc là C In) là một đối tượng của C++ gắn với bàn phím, `>>` là toán tử nhập (“lấy từ”). Toán tử `>>` lấy dữ liệu từ bàn phím đặt vào biến bên phải nó.

✧ Khi thực hiện lệnh `cin` chương trình chờ người sử dụng gõ vào giá trị cho biến và ấn Enter. Giá trị gõ vào nên đúng với kiểu của biến.



3. Lệnh lấy dữ liệu vào từ bàn phím/tệp

✧ Có thể dùng một lệnh để lấy dữ liệu từ bàn phím cho nhiều biến.

`cin>>Biến1>>Biến2>>.....>>BiếnN;`

Với lệnh này, khi nhập giá trị cho các biến thì giữa các giá trị phải cách nhau ít nhất một khoảng trắng (Enter hoặc Space hoặc Tab).

Ví dụ: `cin>>a>>b>>c;`

Kết hợp cin và cout để nhập dữ liệu từ bàn phím

```
cout<<“Lời nhắc: ”; cin>>Biến;
```

Nhập dữ liệu từ tệp văn bản

✧ Khai báo tệp lấy dữ liệu vào gắn với một tên tệp:

```
ifstream fin("Tên tệp");
```

Ví dụ: `ifstream fin("tamgiac.txt");`

✧ Lấy dữ liệu từ tệp `fin` giống như lấy dữ liệu từ bàn phím `cin`:

Ví dụ: `fin>>a>>b>>c;`

II. Định dạng dữ liệu đưa ra

1. Xác định số chỗ cho dữ liệu đưa ra
2. Thiết lập canh trái, phải cho dữ liệu
3. Xác định số chữ số sau dấu chấm thập phân

1. Xác định số chỗ trên màn hình cho giá trị đưa ra

- ✧ Khi đưa dữ liệu ra màn hình ở chế độ Text ta có thể ấn định số chỗ màn hình dành cho dữ liệu. Mỗi chỗ trên màn hình chứa được một ký tự. Màn hình Text thường có 25 dòng, mỗi dòng 80 chỗ. Để ấn định số chỗ ta dùng hàm thành viên `width(w)` của đối tượng `cout`. Viết lệnh như sau: `cout.width(số chỗ);`
- ✧ Lệnh `cout.width(số chỗ);` chỉ có tác dụng đối với 1 giá trị đưa ra màn hình ngay sau đó.
Ví dụ: `cout.width(8); cout<<a+b;`
- ✧ Cứ mỗi giá trị đưa ra cần một lệnh ấn định số chỗ cho nó.

2. Thiết lập căn trái, phải cho dữ liệu

- ✧ Trong số chỗ màn hình dành cho giá trị đưa ra, giá trị có thể nằm về phía bên trái (canh trái) hoặc bên phải (canh phải). Mặc định là canh phải.
- ✧ Để canh trái ta dùng lệnh: `cout.setf(ios::left);`
Lệnh này đặt trước lệnh đưa ra giá trị muốn canh trái.
Ví dụ: `cout.setf(ios::left); cout<<1500;`
- ✧ Tương tự như vậy, để canh phải ta dùng lệnh: `cout.setf(ios::right);`
- ✧ Lệnh thiết lập canh trái/phải ảnh hưởng tới tất cả các lệnh đưa dữ liệu ra màn hình nằm sau nó.

3. Xác định số chữ số sau dấu chấm thập phân

✧ Để xác định số chữ số hiển thị sau dấu chấm thập phân khi đưa ra màn hình một số thực ta dùng lệnh:

```
cout.precision(số lượng chữ số);
```

Ví dụ: `cout.precision(2); cout<<12.345678;`

sau 2 lệnh này trên màn hình hiện 12.35

✧ Lệnh này sẽ làm tròn số nếu số thực cần đưa ra có số chữ số phần thập phân nhiều hơn số chữ số thiết lập.

3. Xác định số chữ số sau dấu chấm thập phân (tiếp)

- ✧ Lệnh `cout.precision` sẽ ảnh hưởng tới tất cả các lệnh `cout` nằm sau nó.
- ✧ Nếu ta dùng lệnh `cout.precision(0)`; thì các số được đưa ra theo mặc định (6 chữ số phần thập phân).

III. Một chương trình C++ đơn giản

Ví dụ 4.1:

Viết chương trình tính diện tích và chu vi hình chữ nhật có 2 cạnh a , b .

Ví dụ 4.2:

Viết chương trình tính diện tích và chu vi hình tròn có bán kính r . Viết 2 phiên bản, phiên bản 1 vào/ra với màn hình/bàn phím; phiên bản 2 vào/ra với tệp văn bản.

Viết trên DOS/Windows

```
//Khai bao su dung thu vien chuong trinh
#include<iostream.h>

#define PI 3.14          //Khai bao hang

void main()
{
    float r,dt,cv;
    cout<<"Nhap vao ban kinh r: ";
    cin>>r;
    dt=PI*r*r;
    cv=2*PI*r;
    cout<<"Dien tich hinh tron la: "<<dt<<endl;
    cout<<"Chu vi hinh tron la: "<<cv;
}
```

Viết trên Linux

```
//Khai bao su dung thu vien chuong trinh
#include<iostream>
using namespace std;
#define PI 3.14          //Khai bao hang
int main()
{
    float r,dt,cv;
    cout<<"Nhap vao ban kinh r: ";
    cin>>r;
    dt=PI*r*r;
    cv=2*PI*r;
    cout<<"Dien tich hinh tron la: "<<dt<<endl;
    cout<<"Chu vi hinh tron la: "<<cv<<endl;
    return 0;
}
```

BÀI TẬP

1) Viết chương trình tính giá trị của biểu thức:

$$Y = 2^x(\log_5(x^2 + 1))$$