

Input/Output trong C++

Thư viện chuẩn C++ cung cấp nhiều khả năng để input/output và sẽ được bàn luận trong các chương sau. Trong chương này, chúng ta thảo luận rất cơ bản và phổ biến nhất về hoạt động I/O cần thiết cho lập trình C++.

I/O trong C++ diễn ra trong các Stream (luồng), mà là dãy các byte. Nếu các byte chảy từ một thiết bị, như bàn phím, disk drive, hoặc một kết nối mạng ..., tới bộ nhớ chính, nó được gọi là hoạt động input. Nếu các byte chảy từ bộ nhớ chính tới các thiết bị, như màn hình hiển thị, máy in, dist drive, hoặc một kết nối mạng ..., nó được gọi là hoạt động output.

Header file cho I/O trong C++

Bảng dưới liệt kê các header file quan trọng cho các chương trình C++:

Header File	Miêu tả
<iostream>	File này định nghĩa các đối tượng cin , cout , cerr và clog , tương ứng với Standard Input Stream (Luồng đầu vào chuẩn), Standard Output Stream (Luồng đầu ra chuẩn), Un-buffered Standard Error Stream (Luồng lỗi chuẩn không được đệm) và Buffered Standard Error Stream (Luồng lỗi chuẩn được đệm)
<iomanip>	File này khai báo các dịch vụ hữu ích để thực hiện hoạt động I/O được định dạng với các bộ thao tác luồng được tham số hóa như setw và setprecision .

<fstream>	File này khai báo các dịch vụ xử lý file được kiểm soát bởi người dùng. Chúng ta sẽ thảo luận chi tiết về nó trong chương File và Stream trong C++
-----------	--

Standard Output Stream (cout) trong C++

Đối tượng tiền định nghĩa cout là một minh họa của lớp ostream. Đối tượng cout được xem như "được kết nối tới" thiết bị đầu ra chuẩn, thường là màn hình. Đối tượng cout được sử dụng kết hợp với toán tử chèn luồng (insertion operator), được viết là <<, như ví dụ dưới đây:

```
#include <iostream>

using namespace std;

int main( )
{
    char str[] = "Xin chao C++";

    cout << "Gia tri cua str la: " << str << endl;
}
```

Khi code trên được biên dịch và thực thi, nó cho kết quả sau:

```
Gia tri cua str la: Xin chao C++
```

Bộ biên dịch C++ cũng quyết định kiểu dữ liệu của biến để được output và chọn toán tử chèn luồng thích hợp để hiển thị giá trị. Toán tử << được nạp chồng (overload) tới dữ liệu đầu ra của kiểu integer, float, double, string, và giá trị con trỏ có sẵn.

Toán tử chèn luồng << có thể được sử dụng nhiều hơn một lần trong một lệnh và endl được sử dụng để thêm một dòng mới tại cuối dòng đó.

Standard Input Stream (cin) trong C++

Đối tượng tiền định nghĩa cin là một minh họa của lớp istream. Đối tượng cin được xem như đính kèm với thiết bị đầu vào chuẩn, mà thường là bàn phím. Đối tượng cin được sử dụng kết hợp với toán tử trích luồng (extraction operator), viết là >>, như trong ví dụ sau:

```
#include <iostream>

using namespace std;

int main( )
{
    char ten[50];

    cout << "Nhập ten của bạn (viết liền): ";

    cin >> ten;

    cout << "Ten bạn là: " << ten << endl;

}
```

Khi code trên được biên dịch và thực thi, nó sẽ nhắc bạn nhập một tên. Bạn nhập một giá trị và sau đó enter để thấy kết quả giống như sau:

```
Nhập ten của bạn (viết liền): NguyenThanhHung
Ten bạn là: NguyenThanhHung
```

Bộ biên dịch C++ cũng quyết định kiểu dữ liệu của giá trị đã nhập và chọn toán tử trích luồng thích hợp để trích giá trị và lưu giữ nó trong các biến đã cung cấp.

Toán tử trích luồng >> có thể được sử dụng nhiều hơn một lần trong một lệnh. Để yêu cầu nhiều hơn một dữ liệu chuẩn, bạn có thể sử dụng:

```
cin >> ten >> tuoi;
```

Nó tương đương với hai lệnh sau:

```
cin >> ten;
```

```
cin >> tuoi;
```

Standard Error Stream (cerr) trong C++

Đối tượng tiền định nghĩa cerr là một minh họa của lớp ostream. Đối tượng cerr được xem như đính kèm với thiết bị lỗi chuẩn, mà cũng là màn hình hiển thị nhưng đối tượng cerr là Un-buffered (không được đệm) và mỗi sự chèn luồng tới cerr làm đầu ra của nó xuất hiện tức thì.

Đối tượng cerr trong C++ cũng được sử dụng kết hợp với toán tử chèn luồng, như trong ví dụ sau:

```
#include <iostream>
```

```
using namespace std;
```

```
int main( )
```

```
{
```

```
    char str[] = "Khong the doc ....";
```

```
cerr << "Thông báo lỗi là: " << str << endl;  
}
```

Khi code trên được biên dịch và thực thi, nó cho kết quả sau:

```
Thông báo lỗi là: Không thể đọc ....
```

Standard Log Stream (clog) trong C++

Đối tượng tiền định nghĩa `clog` là một minh họa của lớp `ostream`. Đối tượng `clog` được xem như đính kèm với thiết bị lỗi chuẩn, cũng là màn hình hiển thị nhưng đối tượng `clog` là được đệm. Nghĩa là, mỗi sự chèn luồng tới `clog` làm đầu ra của nó được giữ trong một bộ đệm (buffer) tới khi bộ đệm bị điền đầy hoặc tới khi bộ đệm bị flush (chuyển từ đệm ra đĩa).

Đối tượng `clog` trong C++ cũng được sử dụng kết hợp với toán tử chèn luồng, như ví dụ sau:

```
#include <iostream>  
  
using namespace std;  
  
int main( )  
{  
    char str[] = "Không thể đọc ....";  
  
    clog << "Thông báo lỗi là: " << str << endl;  
}
```

Khi code trên được biên dịch và thực thi, nó cho kết quả sau:

```
Thông báo lỗi là: Không thể đọc ....
```

Bạn sẽ không thể thấy bất kỳ điểm khác biệt nào giữa cout, cerr và clog với những ví dụ nhỏ này, nhưng trong khi viết và thực thi các chương trình lớn, thì sự khác nhau này trở nên rõ ràng hơn. Vì thế, nó là bài thực hành tốt cho bạn để hiển thị các thông báo lỗi (error message) bởi sử dụng luồng cerr và trong khi hiển thị thông báo log khác, thì clog nên được sử dụng.