

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>

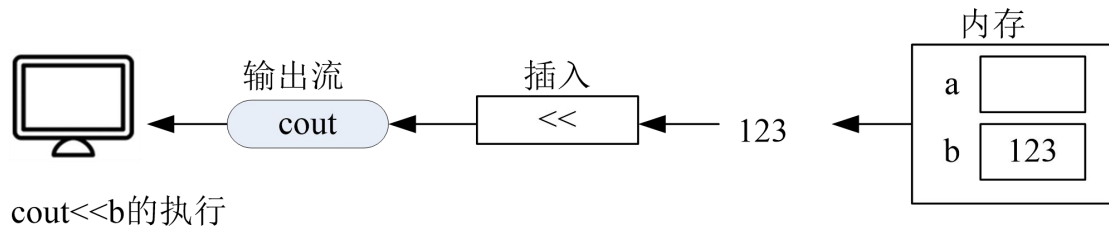




本节主题：

标准输出流

cout流对象



- ❏ 标准输出流，是流向标准输出设备(显示器)的数据。
- ❏ ostream类定义了3个输出流对象：cout, cerr, clog
- ❏ cout是console output的缩写，意为在控制台(终端显示器)的输出。
 - ❏ cout不是C++预定义的关键字，它是ostream流类的对象，在iostream中定义。
 - ❏ “cout<<”输出基本类型的数据时，可以不必考虑数据是什么类型，系统会判断数据的类型，并根据其类型选择调用与之匹配的运算符重载函数。
 - ❏ cout流在内存中对应开辟了一个缓冲区，用来存放流中的数据，当向cout流插入一个endl时，立即输出流中所有数据，然后插入一个换行符，并刷新流(清空缓冲区)。
 - ❏ cout.flush();输出流中所有数据，刷新缓冲区。

cerr流对象

- ❏ cerr流对象是标准错误流，被指定为与显示器关联。
- ❏ cerr的作用是向标准错误设备(standard error device)输出有关出错信息。
- ❏ cerr与标准输出流cout的作用和用法差不多，但
 - ❏ cout流通常是传送到显示器输出，但也可以被重定向输出到磁盘文件；
 - ❏ cerr流中的信息只能在显示器输出。
- ❏ 用途
 - ❏ 当调试程序时，往往不希望程序运行时的出错信息被送到其他文件，而要求在显示器上及时输出，这时应该用cerr
 - ❏ cerr流中的信息是用户根据需要指定的

cerr流对象使用：解方程 $ax^2+bx+c=0$

```
int main( )
{
    float a,b,c,delta;
    cout<<"please input a,b,c:";
    cin>>a>>b>>c;
    if (a==0)
        cerr<<"a is equal to zero,error!"<<endl;
    else if ((delta=b*b-4*a*c)<0)
        cerr<<"delta < 0"<<endl;
    else
    {
        cout<<"x1="<<(-b+sqrt(delta))/(2*a)<<endl;
        cout<<"x2="<<(-b-sqrt(delta))/(2*a)<<endl;
    }
    return 0;
}
```



clog流对象

- ❏ clog流对象也是标准错误流，它是console log的缩写。
- ❏ clog的作用和cerr相同，都是在终端显示器上显示出错信息。
- ❏ 区别
 - ❏ cerr是不经过缓冲区，直接向显示器上输出有关信息
 - ❏ clog中的信息存放在缓冲区中，缓冲区满后或遇endl时向显示器输出。

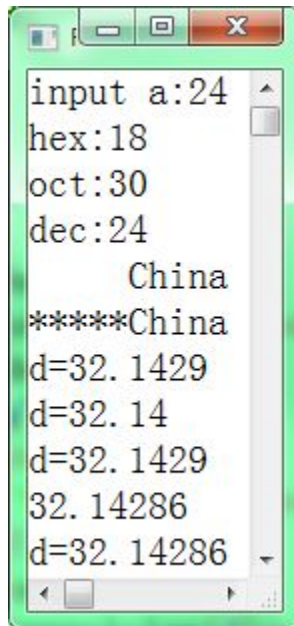


使用控制符控制输出格式

控制符	作用
dec	设置整数的基数为 10
hex	设置整数的基数为 16
oct	设置整数的基数为 8
setbase(n)	设置整数的基数为 n(n 只能是 8,10,16 三者之一)
setfill(c)	设置填充字符 c,c 可以是字符常量或字符变量
setprecision(n)	设置实数的精度为 n 位。在以一般十进制小数形式输出时 n 代表有效数字。在以 fixed(固定小数位数)形式和 scientific(指数)形式输出时 n 为小数位数
setw(n)	设置字段宽度为 n 位
setiosflags(ios::fixed)	设置浮点数以固定的小数位数显示
setiosflags(ios::scientific)	设置浮点数以科学记数法(即指数形式)显示
setiosflags(ios::left)	输出数据左对齐
setiosflags(ios::right)	输出数据右对齐
setiosflags(ios::skipws)	忽略前导的空格
setiosflags(ios::uppercase)	在以科学记数法输出 E 和以十六进制输出字母 X 时以大写表示
setiosflags(ios::showpos)	输出正数时给出“+”号
resetioflags()	终止已设置的输出格式状态,在括号中应指定内容

格式控制示例

```
#include <iostream>
#include <iomanip> //不要忘记包含此头文件
using namespace std;
int main()
{
    int a;
    cout<<"input a:";
    cin>>a;
    cout<<"hex:"<<hex<<a<<endl; //以十六进制形式输出整数a
    cout<<"oct:"<<setbase(8)<<a<<endl; //以八进制形式输出整数a
    cout<<"dec:"<<dec<<a<<endl; //以十进制形式输出整数
    char *pt="China";
    cout<<setw(10)<<pt<<endl; //pt指向字符串"China"
    cout<<setfill('*')<<setw(10)<<pt<<endl; //指定域宽10, 输出字符串, 空
    double d=225.0/7.0; //计算d值
    cout<<"d="<<d<<endl; //输出d值
    cout<<"d="<<setprecision(4)<<d<<endl; //改为4位小数
    cout<<"d="<<setiosflags(ios::fixed)<<d<<endl; //改为固定位数形式输出
    cout<<setprecision(5)<<d<<endl;
    cout<<"d="<<setiosflags(ios::fixed)<<setprecision(5)<<d<<endl; //改为
    return 0;
}
```



用流对象的成员函数控制输出格式

通过调用流对象cout中用于控制输出格式的成员函数来控制输出格式。

流成员函数	与之作用相同的控制符	作 用
precision(n)	setprecision(n)	设置实数的精度为 n 位
width(n)	setw(n)	设置字段宽度为 n 位
fill(c)	setfill(c)	设置填充字符 c
setf()	setiosflags()	设置输出格式状态, 括号中应给出格式状态, 内容与控制符 setiosflags 括号中的内容相同, 如表 13.5 所示
unsetf()	resetioflags()	终止已设置的输出格式状态, 在括号中应指定内容

例

```
cout.width(10);
```

```
cout.setf(ios::oct);
```

```
cout.unsetf(ios::showpoint);
```

流成员函数setf中的格式标志

如：`cout.setf(ios::oct); cout.setf(ios::left);`

格式标志	作 用
<code>ios::left</code>	输出数据在本域宽范围内向左对齐
<code>ios::right</code>	输出数据在本域宽范围内向右对齐
<code>ios::internal</code>	数值的符号位在域宽内左对齐,数值右对齐,中间由填充字符填充
<code>ios::dec</code>	设置整数的基数为 10
<code>ios::oct</code>	设置整数的基数为 8
<code>ios::hex</code>	设置整数的基数为 16
<code>ios::showbase</code>	强制输出整数的基数(八进制数以 0 打头,十六进制数以 0x 打头)
<code>ios::showpoint</code>	强制输出浮点数的小点和尾数 0
<code>ios::uppercase</code>	在以科学记数法格式 E 和以十六进制输出字母时以大写表示
<code>ios::showpos</code>	对正数显示“+”号
<code>ios::scientific</code>	浮点数以科学记数法格式输出
<code>ios::fixed</code>	浮点数以定点格式(小数形式)输出
<code>ios::unitbuf</code>	每次输出之后刷新所有的流
<code>ios::stdio</code>	每次输出之后清除 <code>stdout, stderr</code>

输出流对象的成员函数——先认识 put

- ostream类中专用于输出单个字符的成员函数

```
ostream& put(char c);
```

- 如

```
cout.put('a');
```

- 可以在一个语句中连续调用put函数

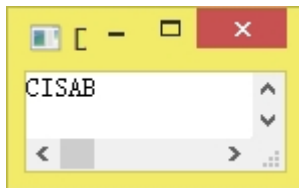
```
cout.put(71).put(79).put(79).put(68);
```

- 例 反序输出字符串"BASIC"

```
char *a="BASIC";
```

```
for(int i=4;i>=0;i--)
```

```
    cout.put(*(a+i));
```



THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>

