3. 标准输出流

- 标准输出流——流向标准输出设备(显示器)的数据

一般使用 cout 流对象进行输出操作。

例如，用流插入运算符“<<”输出数据.

- cout << “a=”<< a << endl;

cout是ostream类的对象，除了“<<”符号外，常用函数如下

函数	功能
put	无格式插入一个字节
write	无格式插入一字节序列
flush	刷新输出流
seekp	移动输出流指针
tellp	返回输出流中指定位置的指针值

(A)用流对象的成员函数控制输出格式

(1)设置状态标志流成员函数setf

调用格式：cout.unsetf(ios::状态标志);

(2)清除状态标志流成员函数unsetf

调用格式：cout.unsetf(ios::状态标志);

(3)设置域宽流成员函数width

调用格式：cout.width(n);

只对下一次流输出有效，输出完成后该函数的作用就消失。

(4)设置实数的精度流成员函数precision

调用格式：cout.precision(n);

参数n在十进制小数形式输出时代表有效数字。在以fixed形式和scientific形式输出时代表小数位数

(5)填充字符流成员函数fill

调用格式：cout.fill(ch);

在cout.setf(ios::状态标志)和cout.unsetf(ios::状态标志)中常用的状态标志如下：

ios类的常用状态标志

状态标志	功能
left	输出数据在本域宽范围内左对齐
right	输出数据在本域宽范围内右对齐
dec	设置整数的基数为10
oct	设置整数的基数为8
hex	设置整数的基数为16
showpoint	浮点数输出时，强制显示小数点
uppercase	在以科学表示法格式E 和以十六进制输出字母时用大写表示
scientific	用科学表示法格式显示浮点数
fixed	用定点格式（固定小数位数）显示浮点数

在引用这些值之前要加上ios::，如果有多项标志，中间则用"|"分隔。

【例4】用流对象的成员函数控制输出

运行结果如下：

```
123.46
****123.46
999.123
99.123
1.235e+002
```

```
#include<iostream>
using namespace std;
int main()
{   cout.setf(ios::left|ios::showpoint);  //设左对齐
    cout.precision(5);          //设置除小数点外有
    cout<<123.456789<<endl;
    cout.width(10);             //设置显示区域宽10
    cout.fill('*');             //在显示区域空白处用*填充
    cout.unsetf(ios::left);     //清除状态左对齐
    cout.setf(ios::right);      //设置右对齐
    cout<<123.456789<<endl;
    cout.setf(ios::left|ios::fixed);  //设左对齐，以固定小数位数显示
    cout.precision(3);          //设置实数显示3位小数
    cout<<999.123456<<endl;         cout<<99.123456<<endl;
    cout.unsetf(ios::left|ios::fixed);   //清除状态左对齐和定点格式
    cout.setf(ios::left|ios::scientific); //设置左对齐，以科学计数法显示
    cout.precision(3);                    //设置保留3位小数
    cout<<123.45678<<endl;
    return 0;   }
```

(B)用C++流格式控制符控制输出格式

多数C++流格式控制符与前面的成员函数有等价对应的关系，两者都可实现同样的功能

C++流格式控制符一般与符号“<<”联用

(1)dec：设置后面的整数按10进制方式显示；
(2)hex：设置后面的整数按16进制方式显示；
(3)oct：设置后面的整数按8进制方式显示；
(4)endl：输出一个换行符并刷新输出流；
(5)setfill(c)：设置填充符(默认为空格)；
(6)setprecision(n)：设置实数精度n，原理和成员函数precision一样；
(7)setw(n)：设置域宽n；
(8)setiosflags(flags)：设置状态标志，多个用'|'分隔
(9)resetiosflags(flags)：清除状态标志，多个用'|'分隔

setiosflags和resetiosflags的常用状态标志

状态标志	功能
ios::left	按域宽左对齐输出
ios::right	按域宽右对齐输出
ios::fixed	固定小数位数输出
ios::showpos	强制设置显示正号
ios::uppercase	科学记数法或16进制输出数据时字母大写
ios::lowercase	科学记数法或16进制输出数据时字母小写

【例5】使用C++控制符控制输出格式

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
        int a=128;
        cout<<"dec:"<<dec<<a<<endl;  //以
        cout<<"hex:"<<hex<<a<<endl;  //以16进制形式输出
        cout<<"oct:"<<oct<<a<<endl;   //以8进制形式输出
        char pt[]="xi'an";
        cout<<setw(10)<<pt<<endl;  //域宽为10，输出字符串
        //指定域宽10，输出字符串，空白处以"*"填充
        cout<<setfill('*')<<setw(10)<<pt<<endl;
        double B=27.123456789;
        //按指数形式输出，8位小数
        cout<<setiosflags(ios::scientific)<<setprecision(8);
        cout<<"B="<<B<<endl; //输出B值
```

```
dec:128
hex:80
oct:200
      xi'an
*****xi'an
B=2.71234568e+001
B=2.7123e+001
B=27.123457
```

```
    cout<<"B="<<setprecision(4)<<B<<endl; //4位小数
    cout<<resetiosflags(ios::scientific);        //清除格式设定
    //改为小数形式输出，小数点后6位
    cout<<"B="<<setiosflags(ios::fixed)<<setprecision(6)<<B;
    cout<<endl;
    return 0;
}
```

```
dec:128
hex:80
oct:200
      xi'an
*****xi'an
B=2.71234568e+001          (输出8位小数)
B=2.7123e+001              (输出4位小数)
B=27.123457                (输出6位小数)
```