Introduction to C++ (Season 2)

Unit 7: File Input and Output

第7单元:出入县同趣, 所向各有宜一文件输入输出

Section 04 : Formating Output

第04节: 格式化输出



Format output to the console (格式化输出到控制台)

- #include <iomanip>
- ❖ setw(n) 设置域宽
 - cout<<setw(3)<<'a'<<'b'<<endl;</pre>

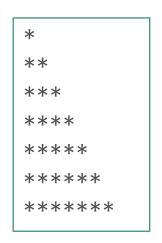
_ _ ab

setfill(c)

- 设置填充字符,即"<<"符号后面的数据长度小于域宽时,使用什么字符进行填充。
- cout<<setfill('*')<<setw(5)<<'a'<<endl;</pre>

****a

```
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    cout<<setfill('*')</pre>
        <<setw(2)<<'\n'
        <<setw(3)<<'\n'
        <<setw(4)<<'\n'
        <<setw(5)<<'\n'
        <<setw(6)<<'\n'
        <<setw(7)<<'\n'
        <<setw(8)<<'\n';
    return 0;
```



The scope of setw()

- ❖ setw()控制符只对其后输出的第一个数据有效
- ❖其他控制符则对其后的所有输入输出产生影响。



- ❖setw()的默认为setw(0),按实际输出。
- ❖如果输出的数值占用的宽度超过setw(int n)设置的宽度,则按实际宽度输出。

```
float f=0.12345;
cout<<setw(3)<<f<<endl;</pre>
```



setprecision

- setprecision(int n)
 - ■可以控制显示浮点数的有效短
 - n代表数字总位数

```
#include <iostream>
#include <iomanip>
using namespace std;
int main( ) {
    float f=17/7.0;
    cout<<f<<endl;</pre>
    cout<<setprecision(0)<<f<<endl;</pre>
    cout<<setprecision(1)<<f<<endl;</pre>
    cout<<setprecision(2)<<f<<endl;</pre>
    cout<<setprecision(3)<<f<<endl;</pre>
    cout<<setprecision(6)<<f<<endl;</pre>
    cout<<setprecision(8)<<f<<endl;</pre>
    return 0;
```

```
setprecision(0)的效果取决于编译器。
不同编译器的实现是不同的。
本例中:
VC++2013输出:2.42857
Dev C++ 5.6.0 (MinGW GCC 4.8.1)
输出: 2
```

```
2.42857
2.42857
2.4
2.4
2.43
2.42857
2.42857
```

Formatting Output in File Operation(在文件操作中格式化输出)

❖ The stream manipulator also works to format output to a file(流控制符同样可以用于文件输出)

Operator	Description
setw(width)	specifies the width of a print field
setprecision(n)	sets the precision of a floating-point number
fixed	displays floating-point numbers in fixed point notation
	将一个浮点数以定点数的形式输出
showpoint	causes a floating-point number to be displayed with
	a decimal point with trailing zeros even if it has
	no fractional part
	将浮点数以带小数点、带结尾 0 的形式输出,即便它没有小数部分
left	justifies the output to the left 输出内容左对齐
right	justifies the output to the right 输出内容右对齐

getline

❖ When using (≥>), Data are delimited by whitespace. (>>运算符要求数据用空

```
格分隔 ifstream input("state.txt");
char city[40];
input >> city;
New York#New Mexico#Texas#Indiana
```

♦ How to read in "New York"?

```
getline(char array[], int size, char delimitChar)

while (!input.eof()) {// Continue if not end of file
   input.getline(city, 40, '#');
   cout << city << endl;
}</pre>
```

get and put

- Two other useful functions are get and put.
- *get: read a character
- * put write a character.

