



一起++ 第3弹

流输入/输出

作者：@孤言 审校：@Alex Cui

在第2弹中，我们学习了变量的有关知识。本弹我们学习 C++ 的一种输入输出方式，称为流输入/输出。你知道吗？输入的英文为 input，输出的英文为 output，所以我们通常把输入输出简写为 I/O。

学完流输入/输出以后将它和变量一起综合运用，与计算机的交流会更进一步哦。

知识点 1 流输出

1. 流输出的代码格式

流输出是 C++ 的一种输出方式。利用对象 `cout` 和运算符 `<<` 我们可以实现数据的输出（对象的概念将在以后介绍），其常用格式为：

```
cout << 表达式;
```

也可以用如下格式来连接 2 个或更多需要输出的内容。（省略号处表示可加更多表达式）

```
cout << 表达式 1 << 表达式 2 <<...;
```

示例 3-1 流输出

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      //字符变量的声明
8      char first = 'Y',
9          mid = 'a',
10         last = 'n';
11     //整型变量的声明
12     int hour = 8;
13     //输出
14     cout << "Hello, " << first << mid << last << '!' << endl;
15     cout << "It's " << hour << " o'clock." << endl;
16     return 0;
17 }
```

输出：

```
Hello, Yan!
It's 8 o'clock.
```

2. 流输出的几点解释

如示例 3-1 所示，我们成功输出了两行内容。如果你有兴趣，也可以尝试着输出其它的内容，但这里我们还有几个需要解释的地方。

C++中，我们把需要输出的字符串看成是一个“字符流”，输出时，我们其实是把一系列字符串插入到这个“流”中。用在流输出的这个运算符`<<`，通常称为**流插入运算符**，可将右边的信息插入到输出流中。箭头的方向形象地指示了流的方向。因此 Ln14 的字符串**"Hello, "**就被插入进去，输出了。

此外，cout 是一个“聪明”的对象，当变量 first 插入时，它不会输出字符串"first"，而是将其所表示的字符输出。同样 Ln 的 15 也按变量 hour 的值输出。（其实会被转化为字符串，不作探讨）

同时我们注意到，如果不作特殊的控制，采用 cout 输出的几个元素之间不会换行，也不会自动空格。Ln14-15 的 3 个字符串内都加上了空格以使程序输出空格。

3.控制符

相信大家已经注意到帮助换行的得力助手 `endl` 了。诸如 endl 等对 cout 有特殊含义的符号通常称为**控制符**。

cout,endl 以及后面要学的 cin 等，都是在头文件 iostream（stream 是流的意思 io 是输入输出的意思）中被定义的，并且包含在命名空间 std 中。所以我们使用它们时必须在程序开头包含头文件：

```
#include <iostream>
```

本教程中暂不讲解命名空间的其它用法，目前只好将 std 命名空间的使用在开头写好。

```
using namespace std;
```

cout 格式化的输出

利用 cout 的一些格式控制符，我们可以实现保留几位小数输出、指定宽度输出等多种格式化的输出方式，学有余力的同学可以去尝试。

cin 和 cout 是 C++中用于输入输出的对象。在 C 语言中，有格式化输入输出函数 printf()和 scanf()，这在 C++中也可以使用，将在以后介绍。

 **注意** 使用格式控制符注意包含头文件 `iomanip`。

知识点 2 流输入

1.流输入的代码格式

同理，利用 `cin` 和**流提取运算符**`>>`我们可以实现数据的输入。常用格式也类似：

```
cin >> 变量;
```

或：

```
cin >> 变量 1 >> 变量 2 >> ...;
```

事实上，cin 也可以用于我们后面会说的数组元素、字符串等的输入。

 **注意** 流输入和流输出的符号分别为`>>`和`<<`，不要写错方向。

</>

示例 3-2A+B Problem (A+B 问题)

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      double a, b;
8      /*声明两个 double 型变量,
9      这样就能愉快地做小数运算了。*/
10     cin >> a >> b;          //输入 a,b
11     cout << a + b << endl; //输出计算结果并换行
12
13     return 0;
14 }
```

Ln7 我们声明了两个变量：a 和 b，为 double 类型，在这里用于存放小数。L10 中用 cin 读入数据，并且把这些数据赋给相应的变量 a,b。整个程序可以用图 1 所示的 Scratch 脚本来近似表示。

2.输入数据的方式

一般地，我们用空格或空行等方式分开各个输入量。比如对于示例 3-2 我们可以这样输入：

```
5.31 4.69

也可以多个空格（虽然很少这样做）：

5.31    4.69

还可以换行或采用其它空白字符：

5.31
4.69
```

输入完毕后，再次敲击回车键，程序都能正常输出结果。

Ln11 中 a+b 是一个表达式，程序会在计算出结果后把结果代替 a+b 输出。比如对于上面的输入，程序算出 a+b 的值为 10，然后用 10 代替原来的 a+b，输出。如果你愿意，也可以把 a+b 赋给 double 型变量 c，然后将 c 输出，这是基本等价的。

至此，你已经有了变量和输入输出这两个编程中的利器，可以先愉快地用它们玩耍一番了。编程的过程也是一个试错的过程，读完本篇教程，你可以自己试着写点别的有趣的东西，同时巩固知识。

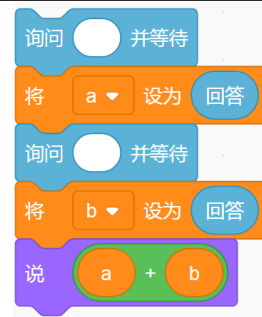


图 1

cin 和 cout 的知识远不只上面所提的这些。其复杂性有时也会造成初学者对它们产生一些误解。



微笑中透露着疲倦
*图片来自网络

误解一：它们是函数。

你家函数长这样？cin 和 cout 分别是 istream 和 ostream 类的对象，而不是函数。

误解二：它们是关键字。

不知为何，常有人觉得 cin（或者是 cout）是保留字或是关键字。如果你真的这么认为，在评论区扣个 0，本来就知道的扣个 1。通常来说，保留字指的是 C++ 保留的一些单词，供自己和 C++ 库使用。而关键字是保留字的一个子集（一部分），是组成编程语言词汇表的标识符。常规的编辑器中会用不同的字体格式区分普通的单词和保留字。如图 2，普通的变量名 apple, number 在该编辑器中显示为白色，cout 和 cin 也是如此（你的编译器也可能是不同的显示格式），而关键字 break, return 和替代标记 and, xor 为另一格式。这样，我们可以从直观上知道 cin, cout 不是保留字。它们没有被内置到语言当中，而只是属于语言标准指定的类，类的定义是写在头文件中的，而不是内置于编译器中。

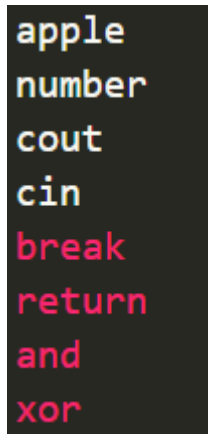


图 2

运算符重载

如果你学习过 C++ 的位运算，你可能会知道 << 和 >> 分别在位运算中是左移位运算符和右移位运算符，但在 cin 和 cout 里却被赋予了不同的含义，具体是哪一个用法会根据前后的情况自动判断。大概来说，这种情况叫运算符重载。



代码练习 1



同学，同学，别跑啊！看这里！新版一起++系列增设了人(臭)见(名)人(昭)爱(著)的代码练习环节。你的第一个代码练习来了！为了方便学习，答案会放在文末。建议先自己做一遍再对答案，开始吧！

Q1 编写一个程序，输出如下内容。

```
*
***
*****
```

Q2 在主函数外部声明整型变量 a 和 b，分别赋值为 12 和 24，将 a 的值除以 2，将 b 的值加 1，然后在函数内部声明整型变量 c，将它赋值为 a, b 的积，输出如下内容（输出中要用到变量 c）：

c = 150

Q3 输入 x 的值，输出式子 $\frac{2}{5}x + 6$ 和式子 $\frac{3}{4}x$ 的值, 用空格隔开。

#输入示例

2.5

#输出示例

7 1.875

Q4 附加题：输入一个小写英文字母，输出它的大写英文字母。（可以参考第 2 弹的蓝框知识）

#输入示例

a

#输出示例

A



答案



Q1 能正确输出的答案都可以。

```
#include <iostream>
#include <cstdlib> //你可以不写，以后同
```

```
using namespace std;

int main()
{
    cout << "  *  " << endl
         << " *** " << endl
         << "*****" << endl;
    /*这里 cout 分 3 行写，是为了美观，你也可以
    写在一行，C++通常以分号为语句结束的标志。*/
    system("pause"); //你可以不写，以后同
    return 0;
}
```

Q2 按要求做，并且答案正确即可。

```
#include <iostream>
#include <cstdlib>

using namespace std;

int a = 12, b = 24; //主函数外部声明变量（它们其实是全局变量）

int main()
{
    a /= 2;
    b++;
    int c; //内部声明变量（是局部变量）
    c = a * b;
    cout << "c = " << c << endl;
    //第一处为字符串，第二处为变量
    system("pause");
    return 0;
}
```

Q3 做这道题时你是否遇到了这样的问题：把表达式写成 $2/5*x+6$ 和 $3/4*x$ ，然后输出了错误结果？这是因为两个整型数据相除时，结果仍是整型（虽然原理上不是，但可以理解为自动向下取整了）。你可以试试输出表达式 $2/5$ 和 $5/5$ 以及 $6/5$ 的值，你发现了吗？

```
#include <iostream>
#include <cstdlib>

using namespace std;

double x; //输入小数，声明为 double 类型

int main()
{
    cin >> x; //输入 x
    cout << 0.4 * x + 6 << " " //注意这里要加空格
         << 0.75 * x << endl;
    system("pause");
    return 0;
}
```

Q4 之前说过，计算机存 char 型字符（这里以 a 为例）时，存的是其字符编码，输入 a 以后，变量 ch 中存放的实际上是其 ASCII 码 97。通过看表我们知道，大写字母的 ASCII 码比对应小写字母小 32，于是将 ch 减 32，得到 ASCII 码 65，cout 对象发现它是 char 型就以 ASCII 码 65 对应的字符 A 输出。这是小写转大写，那么大写转小写你能再写写看吗？

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main()
{
    char ch; //声明一个 char 型变量存字符
    cin >> ch; //输入字符
    ch -= 32;
    cout << ch << endl; //输出字符

    system("pause");
    return 0;
}
```

©2019-2020 孤言，版权所有。未经作者许可，不得以任何形式和方式使用本文的任何内容（包括但不限于文字、图片等）。

第二版日期：2019.9.8

第三版日期：2020.5.9

第四版日期：2020.8.17（2020.9.5 为 A 营重新排版）

字数：3019

