NOIP C++ 编程语言

# 第1章 C/C++快速介绍

## 说，你好！ --- Hello world!

学习一门新程序设计语言的惟一途径就是使用它编写程序。对于所有语言的初学者来说，编写的第一个程序都是相同的，即：

**请打印出下列内容：**

hello, world!

尽管这个练习很简单，但对于初学语言的人来说，它仍然可能成为一大障碍，因为要实现这个目的，我们首先必须编写程序文本，然后成功地运行编译，并加载、运行，最后输出到某个地方。掌握了这些操作细节以后，其它事情就比较容易了。

在 C++语言中，我们可以用下列程序打印出“hello, world!”：



**注意：上面图中代码的高亮显示是编辑器加上帮助我们更好地阅读代码的，你的编辑器不一定使用同样的高亮，或者没有高亮显示，没有关系，不要被它困扰。**

#include <iostream>

这一行指示compiler（编译器）include（包含）iostream里的标准I/O（input/output 输入输出）设施的声明。如果没有这些声明，下面这一句就找不到相应的声明了。（可以试着删掉#include <iostream>这一行，看看编译有什么错误）

std::cout << "Hello world!\n";

<<操作符（“输出到”）将它的第二个参数的内容输出到第一个参数。在这个例子里，它将字符串常量“Hello world!\n” 输出到标准输出流std::cout，即标准字符输出设备.

字符串常量中的\n表示换行符，在打印中遇到它时，输出打印将换行，从下一行的左端行首开始。如果去掉字符串中的\n(这是个值得一做的练习)，即使输出打印完成后也不会换行。

<<操作符的输出不会添加自动换行，这意味着可以将例子中的字符串分多次输出以分阶段得到一个长的输出行，例如上面给出的第一个程序也可以改写成下列形式:



<< 操作符的多次输出可以串联，如下所示：



此外，各个例子中字符串中的\n也可以使用输出单独的std::endl(end of line)来替换，比如：



请注意，\n 只代表一个字符。类似于\n 的转义字符序列为表示无法输入的字符或不可见字符提供了一种通用的可扩充的机制。除此之外，C++ 语言提供的转义字符序列还包括:\t 表示制表符;\b 表示回退符;\"表示双引号;\\表示反斜杠符本身。

**练习 1-1** 在你自己的系统中运行“hello,world”程序。再有意去掉程序中的部分 内容，看看会得到什么出错信息。

**练习 1-2** 做个实验，当“hello,world”程序中的字符串中包含\c(其中 c 是上面的转义字符序列中未曾列出的某一个字符)时，观察一下会出现什么情况。

## 算术表达式

计算机的“本职”工作是计算，因此接下来先从算术运算入手，看看如何用计算机进行复杂的计算。

### 程序1-1　计算并输出1 + 2的值

*本程序主要介绍：*

*整数算术表达式：+ - \* / %*

1 #include <iostream>

2

3 int main()

4 {

5 std::cout << 1 + 2 << std::endl;

6 return 0;

7 }

下面做4个实验。

实验1：修改程序1-1，输出3-4的结果。

实验2：修改程序1-1，输出5×6的结果。

实验3：修改程序1-1，输出8÷4的结果。

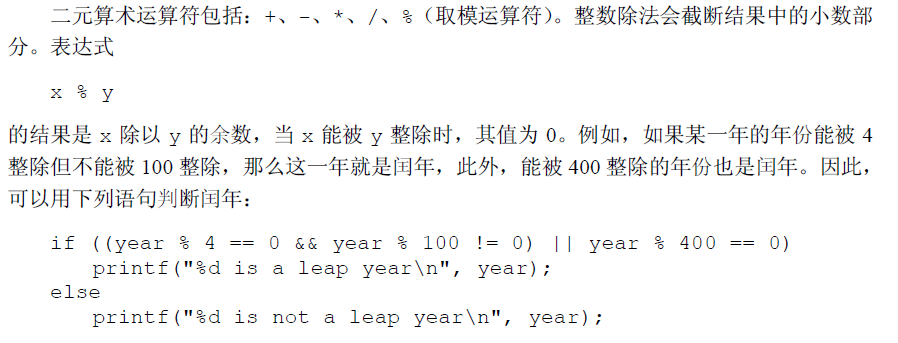
实验4：修改程序1-1，输出8÷5的结果。

**注意：**

整数算术与小数（浮点数）算术

数学里有整数算术和小数算术，计算机如何知道你要整数算术还是小数？

### 算术运算符



### 变量与输入

*本程序主要介绍：*

*变量：即存储*

*标准输入设备*

*完整的输入、存储、处理和输出*

*操作系统？？？待定*

1.1节的程序虽好，但有一个遗憾：计算的数据是事先确定的。为了计算1+2和2+3，下面不得不编写两个程序。可不可以让程序读取键盘输入，并根据输入内容计算结果呢？答案是肯定的。程序如下：

#include <iostream>

int main()

{

int a, b;

std::cin >> a;

std::cin >> b;

std::cout << a + b << std::endl;

return 0;

}

#### 输入

std::cin >> a;

std::cin >> b;

std::cin是C++语言标准库定义的一个标准字符输入设备，它对应着计算机四个任务中的输入（input）。

>>操作符（“输入到”）从它的第一个参数输入内容到第二个参数。在这个例子里，它从标准输入流std::cin先后输入两个整数到变量a和b中.

#### 变量 – 有名字和类型的一块存储

所有的计算机由四部分组成：输入，存储，处理，输出。

在C++语言中，所有变量都必须先声明（申请）后使用。声明用于说明变量的属性，它由一个类型名和一个变量表组成：

* 类型规定了变量可能的取值范围以及可以对其可以执行哪些操作；
* 变量在内存中占据一块存储空间，空间大小由其类型决定；
* 变量的存储空间里放置的是二进制**比特位（bit）序列**，它具体表示的值需要根据变量的类型解释；
* 一个变量就是一个有名字的对象。

例如，本例中定义了两个变量：a和b：

类型int表示其后所列变量为整数（integer），与之相对应的，float表示所列变量为浮点数（即，可以带有小数部分的数）。

除 int与float类型之外，C语高还提供了其它一些基本数据类型，例如：

char 字符

short 短整型

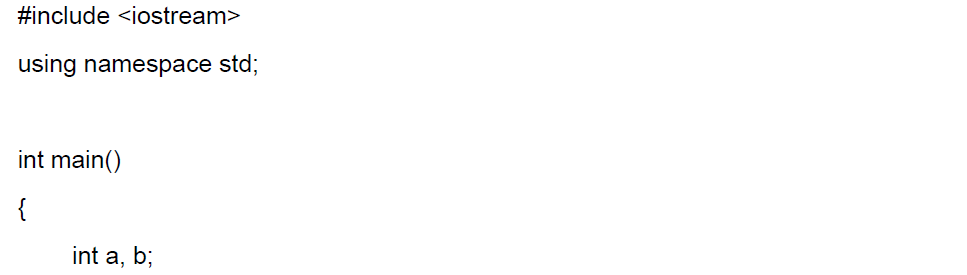
long 长整型

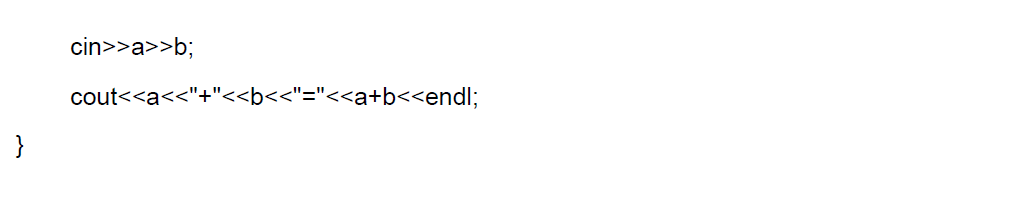
double 双精度浮点型

计算机内部使用二进制数字存放信息，信息的基本单位是比特（bit，binary digit），8个比特组成一个字节（byte）。字节是计算机存储的基本单位。类型的大小通常使用它占用的存储空间的字节数或者位数来表示，同一个类型的变量占据的存储空间的大小是相同的。

各种整型和浮点类型的大小取决于具体的机器，对于int类型，通常为4个字节（32位），其取值范围在-2147483648～2147483647 之间。float 类型通常也是4个字节（32位），它至少有6位有效数字，取值范围一般在10-38～1038之间。

### 练习：阅读程序写结果





## 命名空间-namespace， 哪个小明？

我们来看下一个程序：

输入一个长方体的长、宽和高，求它的棱长总和、表面积和体积（不用管单位）。

例如，

输入： 2,3,5

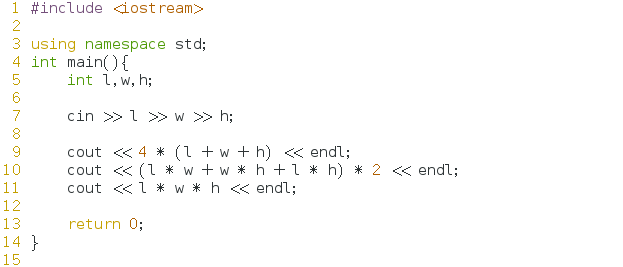
输出：

40

62

30

程序代码如下：



首先，using namespace std 这一行，表示使用名字为std的命名空间，即标准命名空间，它指示编译器在查找名字的时候到标准命名空间中去查找，这样我们在后面使用cout和endl这两个名字的时候就不需要写成**std::**cout和**std::**endl了。

## 分支/条件语句

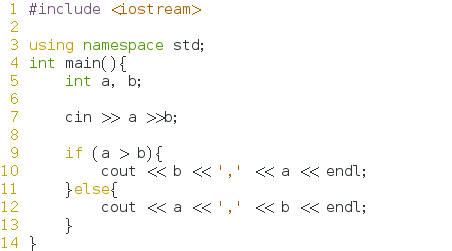
例题：输入两个整数a，b将它们按照从小到大的顺序输出，使用逗号分隔。

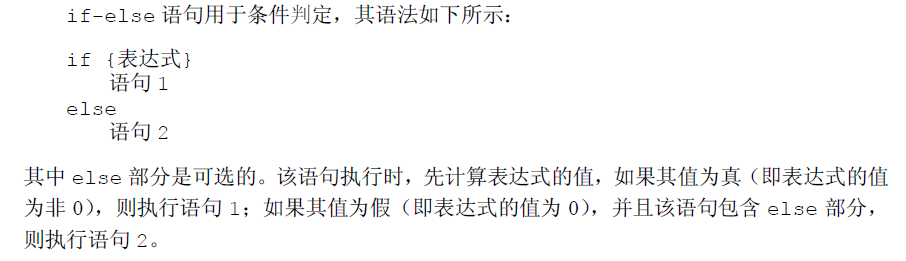
例如输入：

7 6

输出：

6，7

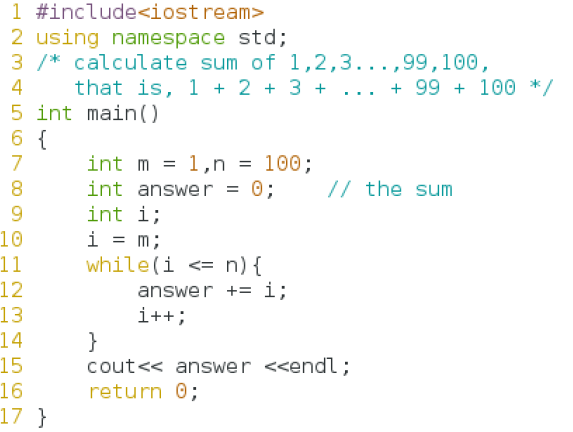




## While 循环

我们来看下一个程序，计算1 + 2 + 3 + … + 99 + 100的和：

因为计算机做算术运算特别快，特别是加法，我们在这个程序中利用计算机的这一个特长来给一个直观的做法---挨个加。此程序中仍然只包括一个名为main 的函数定义。它比前面打印“hello, world”的程序长一些，但并不复杂。这个程序中引入了一些新的概念，包括注释、声明、变量、算术表达式以及循环。该程序如下所示：



第3行和第4行：



称为注释，此处，它简单地解释，该程序是做什么用的。包含在/\*与\*/之间的字符序列将被编译器忽略。注释可以自由地运用在程序中，使得程序更易于理解。程序中允许出现空格、制表符或换行符之处，都可以使用这种注释。注释也能以 // 开始，直到行末为止。例如第8行，绿色部分也是注释，解释下前面的变量的用途：answer的值就是要求的sum。



while 循环语句的执行方式是这样的：首先测试圆括号中的条件；如果条件为真(i <= n)，则执行循环体（括在花括号中的2 条语句）；然后再重新测试圆括号中的条件，如果为真，则再次执行循环体；当圆括号中的条件测试结果为假(i > n)时，循环结束，并继续执行跟在while循环语句之后的下一条语句。即输出我们的答案。

while 语句的循环体可以是用花括号括起来的一条或多条语句（如上面的求和程序），也可以是不用花括号包括的单条语句，例如：

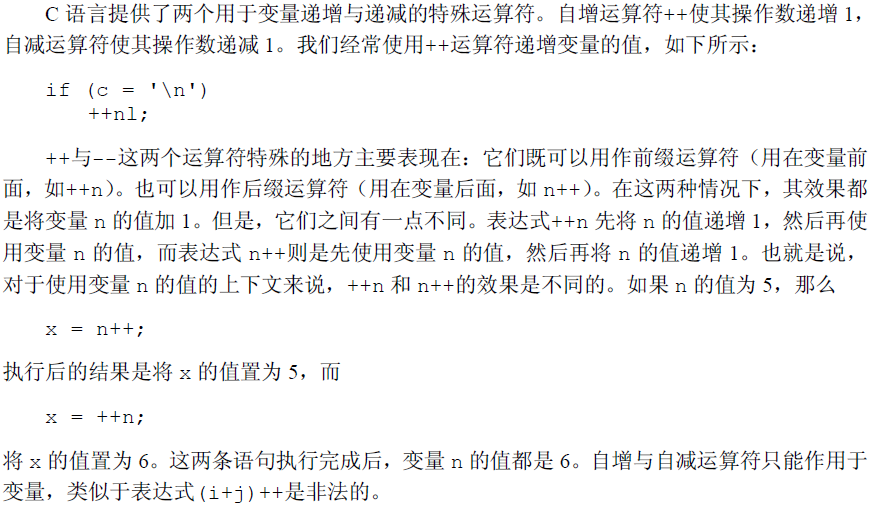
while (i < j)

i = 2 \* i;

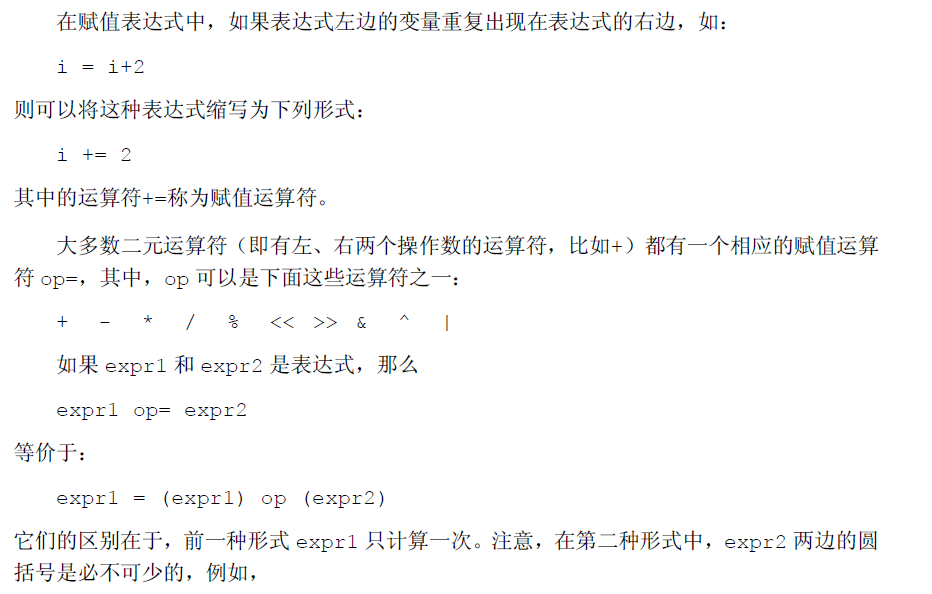
在这两种情况下，我们总是把由while控制的语句缩进一个制表位，这样就可以很容易地看出循环语句中包含哪些语句。这种缩进方式突出了程序的逻辑结构。尽管C 编译器并不关心程序的外观形式，但正确的缩进以及保留适当空格的程序设计风格对程序的易读性非常重要。

我们建议每行只书写一条语句，并在运算符两边各加上一个空格字符，这样可以使得运算的结合关系更清楚明了。相比而言，花括号的位置就不那么重要了。我们从比较流行的一些风格中选择了一种，读者可以选择适合自己的一种风格，并养成一直使用这种风格的好习惯。

## 自增运算符与自减运算符



## 赋值运算符



## 小结：

输入与输出：cin、cout，是标准字符输入输出设备，支持流式操作

存储：变量。变量有类型，有名字。

处理：

程序控制结构：顺序、分支和循环

自增、自减运算符：

练习：三位数反转

输入一个三位数，分离出它的百位、十位和个位，反转后输出

样例输入：

127

样例输出：

721

2019/2/8

编程培训

练习：输入8个数字，输出其中最大的数

介绍数组的含义

如何访问数组的元素

数组的循环遍历

练习：输入两个数字，将大的放入变量a，小的放入变量b

分支语句（条件语句）

交换两个变量的值

练习：输入三个数字，从大到小依次放入变量a，b，c

1. 分而治之（分六种情况）
2. 冒泡（分两次遍历）
3. 函数的定义与使用

练习：输入8个数字，将其从大到小排序输出

冒泡

双层循环

练习：整数的加减乘除

+ - \* /

取余数 %

练习：输入两个数，求它们的和差积商余

变量，存储，类型

Int

练习：简单的计算器，两个操作数和一个操作符，例如 15 + 36

Char

分支语句： if…else if…else if …

2019/2/9

# 简单计算器0

输入简单的二元算术表达式，能够根据输入的操作符（+ - \* /）执行相应的算术运算，例如输入 3 + 4， 输出7； 输入3 \* 5， 输出15.

## 字符与ASCII

文本的ASCII编码

字符类型char,是character，保存一个字符的ASCII码。

定义一个字符变量：

char c；

## 逻辑表达式

命题：a > b. 可能是真（true）或是假（false）

布尔型变量

bool

## else if 语句

