

零距离电脑培训学校

编程系列

Visual C++程序设计教程

周鸣扬 于秋生 等编著



机械工业出版社

本书系统、全面地讲述了 Visual C++ 6.0 中文版的使用方法。全书共 11 课, 内容包括: C++ 语言基础、Visual C++ 编程基础、Windows 编程基础、对话框程序设计、控件编程、ActiveX 控件程序设计、多媒体程序设计、“文档/视图”程序设计、网络程序设计、数据库应用程序开发、多线程程序设计等。

本书由浅入深、循序渐进, 通过大量的小例子来说明编程思路, 便于读者掌握所学知识。

本书可作为计算机程序设计培训班的教材, 也可以作为大专院校 Visual C++ 6.0 课程的教材

图书在版编目 (CIP) 数据

Visual C++ 程序设计教程/周鸣扬等编著. —北京: 机械工业出版社, 2004.9
(零距离电脑培训学校编程系列)

ISBN 7-111-15125-9

I. V... II. 周... III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 084803 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 陈振虹

责任印制: 石 冉

三河市宏达印刷有限公司印刷 • 新华书店北京发行所发行

2004 年 9 月第 1 版 • 第 1 次印刷

787mm×1092mm 1/16 • 16.75 印张 • 402 千字

0001—5000 册

定价: 25.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

出版说明

近几年来,电脑在我国迅速普及,人们的日常生活、娱乐和工作越来越离不开电脑。能够熟练使用电脑也是许多行业对从业者的基本要求。

目前,我国有大量电脑初级用户,他们迫切要求掌握电脑操作的基本方法;还有许多已有一定电脑基础知识的中级用户,希望学会使用与自身工作密切相关的软件。但是在紧张的工作之后,多数人不可能花费太多的时间来系统地学习电脑知识。基于以上背景,我社邀请国内著名计算机职业教育学校的资深老师,为电脑初、中级用户编写了这套“零距离电脑培训学校”丛书。

本套丛书紧紧围绕“短期培训”这个中心,尽量将基础知识与基本技能贯穿于基本操作和应用能力教学之中,书中列举了大量实例,鼓励读者在练中学。丛书强调“不求全、不求精、只求会”,对每一种软件或技术不要求学全,只要学会其中最重要的、与学习者的工作或专业联系最密切的内容就可以。本套丛书通俗易懂、图文并茂,便于读者迅速掌握所学知识。

为了便于读者自学以及培训班授课,我们为部分图书配了电子教案或习题答案,读者可以在我社网站(<http://www.cmpbook.com>)免费下载。

本套丛书覆盖了电脑应用的大部分领域。今后我们会不断补充新的图书,以满足广大读者的需求。

机械工业出版社

前 言

随着 Windows XP、Windows Server 2003 操作系统的发布以及 Visual Studio .NET 2003 在中国的同步上市，Microsoft 在整个软件市场上的盟主地位日趋巩固，由 Microsoft 推出的 Visual 系列开发工具越来越受到程序员的青睐。但是由于 Visual Studio.NET 2003 过于庞大及复杂，使得初学者对学习 Visual C++ 产生一种恐惧感，并且对于如此庞大的系统，许多用户的计算机可能是有点“力不从心”。

而 Visual C++ 6.0 是一个非常稳定成熟的版本，并且该版本对系统的需求也比较低，开发的应用程序几乎可以满足各种类型项目的需要，并且目前大部分用户仍在使用该版本。因此我们决定本书以 Visual C++ 6.0 为基础进行编写。本书详细地向读者介绍了 Visual C++ 的基本编程及高级应用。从简单的控件使用，到高级的网络程序设计，都一一做了详细说明。

本书的特点：

1. 实例与原理并重，内容全面，在语言和结构方面易于理解。
2. 配合 Windows XP 程序来讲述 Visual C++，使全书的内容比较新颖。
3. 本书基本上涵盖了 Visual C++ 在各个方面的编程应用，读者学完本书后可以独立完成不同类型项目的开发。
4. 本书是作者长期从事 Visual C++ 程序设计及教学工作的经验总结。

本书编写过程中，张江涛、刘伟、尹建民、张海霞、吴建文、智雨青、刘旭、赵磊、徐日强、范翠丽、俞雷、薛年喜、郑艳华、王建平、李士良、仇谷峰、李龙等也参加了部分内容的编写及素材整理工作。

本书源代码和习题答案可在机械工业出版社网站 <http://www.cmpbook.com> 免费下载。

编 者

目 录

出版说明

前言

第1课 C++语言基础	1
1.1 课前导读	1
1.2 课堂教学	1
1.2.1 C++语言的发展史	1
1.2.2 C++语言的数据类型	2
1.2.3 C++语言的常量与变量	3
1.2.4 C++语言的运算符	7
1.2.5 C++语言的三种程序结构	11
1.2.6 C++语言的结构、联合、枚举	17
1.2.7 C++语言与面向对象程序设计	20
1.2.8 C++语言的函数	22
1.2.9 C++语言的指针	26
1.3 上机操作	27
1.4 常见问题解答	29
1.5 课后作业	29
第2课 Visual C++编程基础	31
2.1 课前导读	31
2.2 课堂教学	31
2.2.1 Visual Studio 集成开发环境的基本使用	31
2.2.2 选择一种视图	32
2.2.3 使用类视图	33
2.2.4 使用资源视图	34
2.2.5 使用文件视图	37
2.2.6 输出视图	37
2.2.7 编辑代码	38
2.2.8 使用快捷菜单	39
2.2.9 学习菜单系统	40
2.2.10 使用应用程序向导	51
2.3 上机操作	55
2.4 常见问题解答	56
2.5 课后作业	56
第3课 Windows 编程基础	58

3.1 课前导读	58
3.2 课堂教学	58
3.2.1 Windows 的窗口	58
3.2.2 Windows 中的消息	59
3.2.3 理解消息循环	61
3.2.4 消息映射	63
3.2.5 在 MFC 中捕获的消息	64
3.2.6 Windows 消息处理对话框	66
3.2.7 识别消息	67
3.2.8 理解命令及命令更新	67
3.2.9 常见的窗口消息及响应过程	70
3.2.10 使用自定义消息	72
3.3 上机操作	73
3.4 Windows 编程常见问题	75
3.5 课后作业	76
第 4 课 对话框程序设计	77
4.1 课前导读	77
4.2 课堂教学	78
4.2.1 理解对话框	78
4.2.2 使用资源编辑器编辑对话框	79
4.2.3 将对话框资源转换成对话框类	83
4.2.4 在对话框中引用控件	86
4.2.5 对话框的数据交换和数据检查	90
4.2.6 模态对话框与非模态对话框	93
4.2.7 公用对话框	95
4.3 上机操作	102
4.3.1 使用文件对话框	102
4.3.2 使用色彩选择对话框	103
4.4 常见问题解答	104
4.5 课后作业	104
第 5 课 Visual C++ 控件编程	106
5.1 课前导读	106
5.2 课堂教学	106
5.2.1 通用控件与标准控件	106
5.2.2 按钮控件的使用	110
5.2.3 静态文本控件的使用	111
5.2.4 文本编辑框控件	111
5.2.5 列表控件的基本使用	112
5.2.6 树形控件的基本使用	119

5.2.7 工具条控件的基本使用	124
5.2.8 状态条的基本使用	125
5.2.9 对话框工具条的使用	126
5.2.10 自绘控件的编程应用	127
5.3 上机操作	128
5.4 常见问题解答	130
5.5 课后作业	131
第 6 课 ActiveX 控件程序设计	132
6.1 课前导读	132
6.2 课堂教学	132
6.2.1 控件的注册与注销	132
6.2.2 在程序设计中使用 ActiveX 控件	134
6.2.3 使用 ActiveX 控件的属性和函数	135
6.2.4 开发 ActiveX 控件概述	137
6.2.5 用 MFC 开发一个控件	139
6.2.6 用 ATL 开发一个控件	141
6.2.7 调试 ActiveX 控件	144
6.2.8 重绘控件	145
6.3 上机操作	146
6.4 常见问题解答	151
6.5 课后作业	152
第 7 课 Visual C++ 多媒体程序设计	153
7.1 课前导读	153
7.2 课堂教学	153
7.2.1 GDI 与 GDI+	153
7.2.2 从 GDI 编程到 GDI+ 编程	155
7.2.3 GDI+ 程序的开发与项目分发	157
7.2.4 构造 Graphics 对象	160
7.2.5 绘制直线、矩形、曲线和多边形	162
7.2.6 使用 GDI+ 填充区域	169
7.2.7 在 GDI+ 中使用色彩及图片	170
7.2.8 在 GDI+ 中进行文本输出	170
7.2.9 播放动画	171
7.2.10 播放声音	173
7.2.11 使用媒体控制接口播放声音	176
7.3 上机操作	179
7.4 常见问题解答	181
7.5 课后作业	182
第 8 课 Visual C++ “文档/视图”程序设计	183

8.1 课前导读	183
8.2 课堂教学	183
8.2.1 文档/视图概念	183
8.2.2 使用文档/视图结构	185
8.2.3 理解文档类	188
8.2.4 理解视图类	189
8.2.5 在文档与视图之间传递数据	191
8.2.6 使用切分窗口	194
8.3 上机操作	194
8.4 常见问题解答	199
8.5 课后作业	199
第9课 Visual C++网络程序设计	200
9.1 课前导读	200
9.2 课堂教学	200
9.2.1 Windows Sockets 编程基础	200
9.2.2 理解套接字	201
9.2.3 使用 Windows Sockets API	202
9.2.4 Windows Sockets API 程序设计方法	206
9.2.5 Windows Sockets API 编程实例	207
9.2.6 客户端程序设计	207
9.2.7 服务器端的程序设计	210
9.2.8 MFC 对 Sockets 的两种封装类	213
9.3 上机操作	215
9.4 常见问题解答	218
9.5 课后作业	218
第10课 Visual C++数据库应用程序开发	219
10.1 课前导读	219
10.2 课堂教学	219
10.2.1 使用 ODBC 开发数据应用程序	219
10.2.2 注册 ODBC 数据源	220
10.2.3 MFC 的 ODBC 类简介	223
10.2.4 域数据成员与数据交换	225
10.2.5 SQL 查询	227
10.2.6 移动记录	227
10.2.7 CRecordView 类	228
10.2.8 使用 DAO 开发数据库应用程序	228
10.3 上机操作	233
10.4 常见问题解答	236
10.5 课后作业	237

第 11 课 多线程程序设计	238
11.1 课前导读	238
11.2 课堂教学	238
11.2.1 程序、进程、线程三个基本概念	238
11.2.2 多线程编程的实现方式	240
11.2.3 创建工作线程	241
11.2.4 创建用户界面线程	245
11.2.5 线程间通信	249
11.3 上机操作	250
11.4 常见问题解答	256
11.5 课后作业	256

第 1 课 C++语言基础

学习目的:

- 数据类型。
- 常量与变量。
- 运算符。
- 三种不同的程序结构。
- 结构、联合、枚举。
- 类与面向对象程序设计。
- 函数。
- 指针。

学习重点:

- 基本的 C++语法规则。
- C++的程序结构。
- C++的类与面向对象程序设计。

1.1 课前导读

“我该选择什么样的语言开始我的程序设计？”，这是任何一个准备学习程序设计的初学者所面临的第一个问题。

对于程序设计语言的选择，业界有句精典的说法：聪明的程序员使用 Delphi，真正的程序员使用 Visual C++，这句话道出了选择 Visual C++作为程序设计语言的两个特点：Visual C++语言很专业，但学习起来不容易。

从 C 语言诞生到现在已经有 30 多个年头了，其间所经历的变化并不能够改变它是目前程序员最为钟爱的程序设计语言这一事实。微软的 Visual Studio 2003 的推出，让 C++这一个不老的传说有了新的活力。

1.2 课堂教学

1.2.1 C++语言的发展史

C 语言是在 20 世纪 70 年代初问世的。1978 年美国电话电报公司（AT&T）贝尔实验室



正式发表了C语言。同时由B.W.Kernighan和D.M.Ritchie合著了著名的《The C Programming Language》一书，通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的标准C语言，后来由美国国家标准学会在此基础上制定了一个C语言标准，并于1983年发表，通常称之为ANSI C。

早期的C语言主要是用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐为人们认识，到了20世纪80年代，C语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

C语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，还可以直接访问内存的物理地址，进行位（bit）一级的操作。由于C语言实现了对硬件的编程操作，因此C语言集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适合于应用软件的开发。此外，C语言还具有效率高，可移植性强等特点，因此被广泛地移植到了各种类型的计算机上，从而形成了多种版本的C语言。

在C语言的基础上，1983年又由贝尔实验室的Bjarne Stroustrup推出了C++语言。C++语言进一步扩充和完善了C语言，成为一种面向对象的程序设计语言。C++语言目前比较流行的版本是Borland C++ Builder、Symantec C++ 6.1和Microsoft Visual C++系列。C++语言提出了一些更为深入的概念，它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法，因而也增加了整个语言的复杂性，掌握起来有一定难度。

1.2.2 C++语言的数据类型

C++语言有五种基本数据类型：字符、整型、单精度实型、双精度实型和空类型。整数一般与CPU字长相等，一个字符通常为一个字节，浮点值的确切格式则根据实现而定。对于多数计算机，表1-1给出了五种数据的长度和范围。

表 1-1 基本类型的字长和取值范围

类 型	长度/bit	取 值 范 围
char (字符型)	8	0~255
int (整型)	16	-32 768~32 767
float (单精度型)	32	约精确到6位数
double (双精度型)	64	约精确到12位数
void (空值型)	0	无值

表1-1中的长度和范围的取值是假定CPU的字长为16bit。

C++语言还提供了几种聚合类型，包括数组、指针、结构、联合、位域和枚举。这些复杂类型在以后的章节中讨论。

除void类型外，基本类型的前面可以有各种修饰符。修饰符用来改变基本类型的意义，以便更准确地适应各种情况的需求。修饰符如下：

- signed (有符号)。
- unsigned (无符号)。

- long (长型符)。
- short (短型符)。

修饰符 signed、short、long 和 unsigned 适用于字符和整数两种基本类型，而 long 还可用于 double (注意，由于 long float 与 double 意思相同，所以 ANSI 标准删除了多余的 longfloat)。

表 1-2 给出所有根据 ANSI 标准而组合的类型、字宽和范围。切记，在计算机字长大于 16 位的系统中，short int 与 signed char 可能不等。

表 1-2 ANSI 标准中的数据类型

类 型	长度/bit	范 围
char (字符型)	8	ASCII 字符
unsigned char (无符号字符型)	8	0~255
signed char (有符号字符型)	8	-128~127
int (整型)	16	32 768~32 767
unsigned int (无符号整型)	16	0~65 535
signed int (有符号整型)	16	同 int
short int (短整型)	8	-128~127
unsigned short int (无符号短整型)	8	0~255
Signed short int (有符号短整型)	8	同 short int
long int (长整型)	32	2 147 483 648~2 147 483 649
Signed long int (有符号长整型)	32	2 147 483 648~2 147 483 649
unsigned long int (无符号长整型)	32	0~4 294 967 296
float (单精度型)	32	约精确到 6 位数
double (双精度型)	64	约精确到 12 位数

因为整数的默认定义是有符号数，所以 signed 这一用法是多余的，但仍允许使用。某些编译器允许将 unsigned 用于浮点型，如 unsigned double。但这一用法降低了程序的可移植性，故建议一般不要采用。

为了方便，C 编译程序允许使用整型的简写形式：

- short int 简写为 short。
- long int 简写为 long。
- unsigned short int 简写为 unsigned short。
- unsigned int 简写为 unsigned。
- unsigned long int 简写为 unsigned long。

即，int 可缺省。

1.2.3 C++语言的常量与变量

C++语言中，常量是不接受程序修改的固定值，常量可为任意数据类型。C++语言还支持另一种预定义数据类型的常量，这就是串。所有串常量括在双撇号之间，例如"This is a test"。切记，不要把字符和串相混淆，单个字符常量是由单撇号括起来的，如'a'。



C++语言中，其值可以改变的量称为变量。一个变量应该有一个名字（标识符），在内存中占据一定的存储单元，在该存储单元中存放变量的值。请注意区分变量名和变量值这两个不同的概念。

所有的 C 变量必须在使用之前定义。定义变量的一般形式是：

```
type variable_list;
```

这里的 `type` 必须是有效的 C 语言数据类型，`variable_list`（变量表）可以由一个或多个由逗号分隔的多个标识符名构成。下面给出一些定义的范例。

```
int i,j,l;
short int si;
unsigned int ui;
double balance,profit,loss;
```

注意 C++语言中变量名与其类型无关。下面给出在 C++语言中经常使用到的不同类型的常量与变量。

1. 整型常量

整型常量及整常数。它可以是十进制、八进制、十六进制数字表示的整数值。

十进制常数的形式是：

`digits`

这里 `digits` 可以是 0~9 的一个或多个十进制数位，第一位不能是 0。

八进制常数的形式是：

`0digits`

在此，`digits` 可以是一个或多个八进制数（0~7 之间），起始 0 是必须的引导符。

十六进制常数的形式是：

`0x hdigits`
`0X hdigits`

这里 `hdigits` 可以是一个或多个十六进制数（从 0~9 的数字，及从 a~f 的字母）。引导符 0 是必须有的，字母 X 可用大写或小写。

注意，空白字符不可出现在整数数字之间。表 1-3 列出了整常数的形式。

表 1-3 整常数的例子

十 进 制	八 进 制	十 六 进 制
10	12	0Xa 或 0XA
132	204	0X84
32179	76663	0X7db3 或 0X7DB3

整常数在不加特别说明时总是正值。如果需要的是负值，则负号“-”必须放置于常数表达式的前面。



每个常数依其值要给出一种类型。当整常数应用于表达式时,或出现有负号时,常数类型自动执行相应的转换,十进制常数可等价于带符号的整型或长整型,这取决于所需的常数的尺寸。八进制和十六进制常数可对应整型、无符号整型、长整型或无符号长整型,具体类型也取决于常数的大小。如果常数可用整型表示,则使用整型。如果常数值大于一个整型所能表示的最大值,但又小于整型位数所能表示的最大数,则使用无符号整型。同理,如果一个常数比无符号整型所表示的值还大,则它为长整型。如果需要,当然也可用无符号长整型。

在一个常数后面加一个字母 l 或 L,则认为是长整型。如 10L、79L、012L、0115L、0XAL、0x4fL 等。

2. 字符常量

字符常量是指用一对单引号括起来的一个字符。如 'a'、'9'、'!'。字符常量中的单引号只起定界作用,并不表示字符本身。单引号中的字符不能是单引号 (') 和反斜杠 (\),它们特有的表示法,在转义字符中介绍。

在 C++语言中,字符是按其所对应的 ASCII 码值来存储的,一个字符占一个字节。注意字符 '9' 和数字 9 的区别,前者是字符常量,后者是整型常量,它们的含义和在计算机中的存储方式都截然不同。

由于 C++语言中字符常量是按整数 (short 型) 存储的,所以字符常量可以像整数一样在程序中参与相关的运算。例如:

```
'a'-32;    /*执行结果 97-32=65*/
'A'+32;    /*执行结果 65+32=97*/
'9'-9;     /*执行结果 57-9=48*/
```

字符串常量在内存中存储时,系统自动在字符串的末尾加一个“串结束标志”,即 ASCII 码值为 0 的字符 NULL,常用 \0 表示。因此在程序中,长度为 n 个字符的字符串常量,在内存中占有 n+1 个字节的存储空间。

例如,字符串 China 有 5 个字符,作为字符串常量 "China" 存储于内存中时,共占 6 个字节,系统自动在后面加上 NULL 字符,其存储形式为:

C	h	i	n	a	\0
---	---	---	---	---	----

要特别注意字符串与字符串常量的区别,除了表示形式不同外,其存储性质也不相同,字符 'A' 只占 1 个字节,而字符串常量 "A" 占 2 个字节。

3. 转义字符

转义字符是 C++语言中表示字符的一种特殊形式。通常使用转义字符表示 ASCII 码字符集中不可打印的控制字符和特定功能的字符,如用于表示字符常量的单引号 (')、用于表示字符串常量的双引号 (") 和反斜杠 (\) 等。

转义字符用反斜杠 "\" 后面跟一个字符或一个八进制或十六进制数表示。表 1-4 给出了 C++语言中常用的转义字符。

字符常量中使用单引号和反斜杠以及字符串常量中使用双引号和反斜杠时,都必须使用



转义字符表示，即在这些字符前加上反斜杠。

表 1-4 转义字符

转 义 字 符	意 义	ASCII 码值（十进制）
\a	响铃（BEL）	7
\b	退格（BS）	8
\f	换页（FF）	12
\n	换行（LF）	10
\r	回车（CR）	13
\t	水平制表（HT）	9
\v	垂直制表（VT）	11
\\	反斜杠	92
\?	问号字符	63
\'	单引号字符	39
\"	双引号字符	34
\0	空字符（NULL）	0
\ddd	任意字符	三位八进制
\xhh	任意字符	二位十六进制

在 C 程序中使用转义字符或者“\xhh”可以方便灵活地表示任意字符。\\ddd 表示斜杠后面跟三位八进制数，该三位八进制数的值即为对应的八进制 ASCII 码值。\\x 后面跟两位十六进制数，该两位十六进制数为对应字符的十六进制 ASCII 码值。

使用转义字符时需要注意以下问题：

- 转义字符中只能使用小写字母，每个转义字符只能看作一个字符。
- \\v 垂直制表和\\f 换页符对屏幕没有任何影响，但会影响打印机执行响应操作。
- 在 C 程序中，使用不可打印字符时，通常用转义字符表示。

4. 预定义常量

C++语言允许将程序中的常量定义为一个标识符，称为预定义常量。预定义常量一般使用大写英文字母表示，以区别于一般用小写字母表示的变量。预定义常量在使用前必须先定义，定义的形式是：

```
#define <预定义常量名> <常量>
```

例如：

```
#define PI 3.1415926
#define TRUE 1
#define FALSE 0
#define STAR '*'
```

这里定义 PI、TRUE、FLASE、STAR 为预定义常量，其值分别为 3.1415926，1，0，'*'。“#define”是 C++语言的预处理命令，它表示经定义的预定义常量在程序运行前将由其对应的常量替换。

定义预定义常量的目的是为了提高程序的可读性，便于程序的调试和修改。因此在定义符号常量名时，应使其尽可能地表达它所代表的常量的含义，例如前面所定义的预定义常量名 PI，表示圆周率 3.1415926。此外，若要对一个程序中多次使用的预定义常量的值进行修改，只须对预处理命令中定义的常量值进行修改即可。

5. 字符变量

字符变量用来存放字符常量，注意只能存放一个字符，不要以为在一个字符变量中可以放字符串。字符变量的定义形式如下：

```
char c1,c2;
```

它表示 c1 和 c2 为字符变量，各放一个字符。因此可以用下面语句对 c1、c2 赋值：

```
c1='a'; c2='b';
```

在这一节中，谈到了不同类型的数量与常量，为加深读者对这些知识的了解，下面给出一个简单的例程。

例程 1-1 使用常量与变量。

```
#include "stdafx.h"
int main(int argc, char* argv[])
{
    //整型变量的使用
    int a=10;
    int b;
    b=a*13;
    printf("a=%d,b=%d\n",a,b);
    //字符变量的使用
    char c1,c2;
    c1='a';
    c2='b';
    c1=c1-32;c2=c2-32;
    printf("%c%c\n",c1,c2);
    return 0;
}
```

可以想像一下，例程 1-1 的输出将会是什么。如果读者对例程 1-1 的具体含义不是很清楚，请接着阅读后面的内容。

1.2.4 C++语言的运算符

C++语言的内部运算符很丰富，运算符是告诉编译程序执行特定算术或逻辑操作的符号。C++语言有三大运算符：算术、关系与逻辑、位操作。另外，还有一些特殊的运算符，用于完成一些特殊的任务。

1. 算术运算符

在 C++语言中，算术运算符主要包括“+”（加）、“-”（减）、“*”（乘）和“/”（除）、



“%”（求模）、“++”（自增）、“--”（自减）七种。

C++语言的算术运算符的用法与大多数计算机语言相同，几乎可用于所有 C++语言内定义的数据类型。

当“/”被用于整数或字符时，结果取整。例如，在整数除法中， $10/3=3$ 。模运算符“%”在 C++语言中也同它在其他语言中的用法相同。需要注意的是，模运算取整数除法的余数，所以“%”不能用于 float 和 double 类型。请见下列的简单代码：

```
int x,y;
x=10;
y=3;
printf("%d",x/y);          /*显示 3*/
printf("%d",x%y);          /*显示 1,整数除法的余数*/
x=1;
y=2;
printf("%d,%d",x/y,x%y); /*显示 0,1*/
```

最后一行打印一个 0 和一个 1，因为 $1/2$ 整除时为 0，余数为 1，故 $1\%2$ 取余数 1。

C++语言中有两个很有用的运算符，通常在其他计算机语言中是找不到它们的：自增和自减运算符，++和--。运算符“++”是操作数加 1，而“--”是操作数减 1，例如：

```
x=x+1;    /*同++x;*/
x=x-1;    /*同--x;*/
```

自增和自减运算符可用在操作数之前，也可放在其后，例如： $x=x+1$ ；可写成 $++x$ ；或 $x++$ ；但在表达式中这两种用法是有区别的。如果自增或自减运算符在操作数之前，则在引用操作数之前就先执行加 1 或减 1 操作；如果运算符在操作数之后，则先引用操作数的值，然后再进行加 1 或减 1 操作。例如：

```
x=10;
y=++x;
```

此时， $y=11$ 。如果程序改为：

```
x=10;
y=x++;
```

则 $y=10$ 。在这两种情况下， x 都被置为 11，但区别在于设置的时刻，这种对自增和自减发生时刻的控制是非常有用的。在大多数 C 编译程序中，为自增和自减操作生成的程序代码比等价的赋值语句生成的代码要快得多，所以尽可能采用加 1 或减 1 运算符是一种好的选择。

2. 关系和逻辑运算符

关系运算符中的“关系”二字指的是一个值与另一个值之间的关系，逻辑运算符中的“逻辑”二字指的是连接关系的方式。因为关系和逻辑运算符常在一起使用，所以将它们放在一起讨论。

C++语言的关系运算符分为“>”（大于）、“>=”（大于等于）、“<”（小于）、“<=”（小