

高等院校计算机专业教育改革推荐教材

面向对象 程序设计基础

赵 雷 朱晓旭 编著

徐汀荣 审



机械工业出版社
CHINA MACHINE PRESS



高等院校计算机专业教育改革推荐教材

面向对象程序设计基础

赵 雷 朱晓旭 编著

徐汀荣 审

机械工业出版社

本书介绍计算机程序设计方法中的重要方法——面向对象的程序设计（OOP）。全书分为8章。第1章介绍了C++语言与C语言的关系及C++编译器的使用，以便使读者可以很快排除语法表达上的障碍。第2章介绍了面向对象的程序设计思想及其基本概念。第3章介绍C++语言的重载特性。第4章介绍了C++语言的继承性。第5章介绍了模板。上述章节是C++语言的核心，是面向对象程序设计方法最重要的部分。然后本书本着实用的宗旨，在第6章中介绍了Microsoft Visual C++使用基础。第7章介绍了Windows环境下进行面向对象的应用程序开发的基本方法。第8章介绍了Visual C++与数据库访问。

本书对面向对象程序设计所涉及的内容进行了内容的精选，重点介绍了该方法最核心的思想和内容。适合初学面向对象程序设计的人员，可以作为高校面向对象程序设计课程的专业教科书。

图书在版编目（CIP）数据

面向对象程序设计基础/赵雷,朱晓旭编著. —北京:机械工业出版社,2003.6
高等院校计算机专业教育改革推荐教材
ISBN 7-111-12182-1

I. 面... II. ①赵...②朱... III. 面向对象语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字（2003）第 037268 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：路 琳

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2003 年 6 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·13.75 印张·337 千字

0001—5000 册

定价：20.00 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

高等院校计算机专业教育改革推荐教材

编委会成员名单

主 编 刘大有

副主编 王元元

编 委 (按姓氏笔画排序)

李师贤 刘晓明 张桂芸 徐汀荣

耿亦兵 黄国兴 顾军华 薛永生

编者的话

计算机科学技术日新月异的飞速发展和计算机科学技术专业教育的相对滞后,已是不争的事实。

有两个发人深省的现象:一是,由于非计算机专业的学生既具有一门非计算机专业的专业知识,又具有越来越高的计算机应用技术水平,从而使计算机专业的学生感受到一种强烈的冲击和压力;二是,创建软件学院的工作已有近两年的历史,但软件学院的计算机专业教育的定位仍在探讨之中。

我们认为计算机科学与技术专业(以下简称计算机专业)教育的改革势在必行,正确认识和划分计算机专业教育的层次,对该专业的教育改革无疑是一个非常重要的问题。我国的计算机专业教育主要分三个层次。一般说来,这三个层次通常分布在以下三类高等院校:

第一层次主要以具有计算机一级学科博士学位授予权的教育部属重点高等院校为代表(包括具有两个博士点的大学)。这一类大学本科着重培养理论基础比较坚实、技术掌握熟练、有一定研究和开发能力的计算机专业学科型人才,其中部分学生(约本科生的10%)可攻读博士学位。

第二层次主要以具有一个计算机二级学科专业博士点的教育部属高等院校为代表。这一类高等院校本科着重培养有一定的理论基础、技术掌握比较熟练、有一定的研究或开发能力的计算机专业人才,其中一部分培养成学科型人才,另一部分培养成应用型人才,一小部分学生(约本科生的5%)可攻读博士学位。

第三层次主要以具有计算机二级学科专业硕士点的省属高等院校为代表。这一类高等院校本科面向企业应用,侧重培养对计算机技术或部分计算机技术掌握比较熟练,有一定的开发、应用能力的计算机专业应用型人才,其中很小一部分学生(约本科生的2.5%)可攻读博士学位。

国家教育部、计委批准的或省教育厅批准的示范性软件学院,就其培养目标和办学特色而言,分别与第二层次中应用型人才培养部分以及第三层次比较相近,但在如下方面有所不同:将软件工程课程作为专业教学重点;更加强调英语教学,更加重视实践能力培养,并对两者有更高的要求。

我们本着对高等院校的计算机专业状况的认识,主要面向与上述第二、第三两个层次对应的院校及与之相近的软件学院,总结多年的计算机专业的教改经验,在一定程度上溶入了ACM& IEEE CC2001和CCC2002(中国计算机科学与技术学科教程)的教改思路,组织我国一直投身于计算机教学和科研的教师,编写了这套“高等院校计算机专业教育改革推荐教材”(以下简称“推荐教材”)。自然,“推荐教材”中所贯穿的改革思路和做法,也是针对上述第二、第三两个层次对应院校的计算机专业学生。这些思路和做法可概括成以下三句话:

- 适度调整电子技术基础、计算机理论基础和系统软件的教学内容。
- 全面强化计算机工具软件、应用软件的教学要求。
- 以应用为目标大力展开软件工程的教学与实践。

电子技术基础、计算机理论基础、系统软件教学关系到学生的基本素质、发展潜力和日后的应变能力。“推荐教材”在调整它们的教学内容时的做法是:适度压缩电子线路、数字

电路和信号系统的教学内容,变三门课程为两门,并插入数字信号处理的基础内容;合并“计算机组成原理”、“微型计算机接口技术”和“汇编语言”为“计算机硬件技术基础”一门课程;注意适当放宽“离散数学”课程的知识面,使之与 CCC2002 的要求基本接轨,但适度降低其深度要求;更新系统软件课程的教学内容,以开放代码的 Linux 作为操作系统原理的讲授载体,更加关注系统软件的实践性和实用性。

为了提高计算机专业人才的计算机应用能力,全面强化计算机工具软件、实用软件的教学要求是十分重要的,这也是上述改革思路的核心。为此,“系列教材”的做法是:强化程序设计技术,强化人机接口技术,强化网络应用技术。

为强化程序设计技术,“推荐教材”支持在单片机环境、微机平台、网络平台的编程训练;支持运用程序设计语言、程序设计工具以及分布式对象技术的编程训练。大大加强面向对象程序设计课程的组合(设计了三门课程:面向对象的程序设计语言 C++,面向对象的程序设计语言 JAVA 和分布式对象技术),方便教师和读者的选择。

为强化人机接口技术,“推荐教材”设计了“人机交互教程”,“计算机图形学”和“多媒体应用技术”等可供选择的、有层次特色的课程组合。

为强化网络应用技术,“推荐教材”设计了“计算机网络技术”,“计算机网络程序设计”,“计算机网络实验教程”和“因特网技术及其应用”等可供选择的、新颖丰富的课程组合。

将软件工程课程作为专业教学重点,以应用为目标大力展开软件工程的教学与实践,是“推荐教材”改革思路的又一亮点。为改变以往软件工程课程纸上谈兵的老毛病,“推荐教材”从工程应用出发,理论联系实际,突出建模语言及其实现工具的运用,设计了“软件工程的方法与实践”,“统一建模语言 UML 导论”和“ROSE 对象建模方法与技术”等可供选择的、创新独特的软件工程课程组合。对于各类软件学院,“推荐教材”的这一特色无疑是很有吸引力的。

强调实践也是计算机学科永恒的主题,对计算机应用专业的学生来说更是如此。重应用和重实践是“推荐教材”的一个整体特点。这一特点,一方面有利于解决本文开始所指出的计算机专业学生较之非计算机专业学生,在应用开发工作中上手慢的问题;另一方面,使计算机专业的学生能在更大范围内、更高层面上掌握计算机应用技术。这一特点正是许多高等院校计算机专业教育改革追求的一个目标,也是国家教育部倡导软件学院的初衷之一。

“推荐教材”由基础知识、程序设计、应用技术、软件工程和实践环节等五个模块组成。各模块有其对应的培养目标与功能,从而构架出一个创新的、完整的计算机应用专业的课程体系。模块化的设计,使各学校可根据学生及学校的特点做自由的选择和组合,既能达到本专业的总体要求,又能体现具有特色的个性发展。整套教材的改革脉络清晰,结构特色鲜明,值得各高等院校在改革教学内容、编制教学计划、挑选教材书目时借鉴和参考。当然,很多书目也适合很多相关学科的计算机课程用作教材。

“推荐教材”的组成模块和书目详见封底。显然它不能说是完备的(实践环节模块更是如此),其改革的思路、改革的举措也可能有值得探讨的地方。我们衷心希望得到计算机教育界同仁和广大读者的批评指正。

高等院校计算机专业教育改革推荐教材
编委会

前 言

随着计算机软件技术的发展, 计算机软件应用范围越来越广, 复杂程度也越来越高, 传统的结构化程序设计思想越来越不能适应软件开发工作的需要。面向对象技术是计算机应用领域迅速发展起来的一种软件开发技术, 与传统的结构化程序设计思想相比有许多优点。采用面向对象的观点看待所要解决的问题并将其抽象为系统是极其自然的, 因为它更符合人类的思维习惯。

面向对象的程序设计方法突出的优点是代码的可重用性好, 这样就大大缩短了编程、修改和维护的时间, 可以大幅度提高软件开发的效率。由于面向对象的程序设计方法的突出优点, 目前它已经成为开发大型软件时采用的主要方法。

近年来, 许多传统的程序设计语言都引进了面向对象的程序设计思想, 如 Windows 环境下的 Visual Basic、Visual FoxPro、Visual C++、Delphi、C++Builder 等, 因此有必要让学生了解面向对象的程序设计思想, 并能够较熟练地使用一种面向对象的程序设计语言。

C++语言是一种影响较大的面向对象的程序设计语言, 它较全面地体现了面向对象的程序设计思想。C++是 C 语言的超集, 适合在 C 语言课程的基础上学习面向对象的基本思想和程序设计方法。

本书首先介绍了 C++语言与 C 语言的关系及 C++编译器的使用, 以便使读者可以很快排除语法表达上的障碍。第 2 章介绍了面向对象的程序设计思想及其基本概念。第 3 章介绍 C++语言的重载特性。第 4 章介绍了 C++语言的继承性。上述章节是 C++语言的核心, 是面向对象程序设计方法最重要的部分。

然后本书本着实用的宗旨, 以 Microsofe Visual C++为基础, 介绍了 Windows 环境下进行面向对象的应用程序开发的基本方法。

长期以来, C++语言一直被认为是比较难使用的专业开发工具, 因为它的涉及面很广。而事实上, 针对高等学校教学和爱好者自学的需要, 完全没有必要一开始就将大量琐碎的细节问题摆在学习者的面前。因此, 本书在内容上做了精选, 使得读者可以更方便地掌握面向对象思想的最精华部分。

本书由赵雷、朱晓旭编写, 第 6 章、第 7 章和第 8 章由朱晓旭编写, 其余各章由赵雷编写并由赵雷统稿。徐汀荣教授审阅了全书。另外, 本书在编写过程中得到了苏州大学计算机科学与技术学院的大力支持, 在此谨表谢意。

编 者

目 录

编者的话

前言

第 1 章 从 C 到 C++	1
1.1 C++语言的基本要素	2
1.1.1 字符集与保留字	2
1.1.2 数据类型	2
1.1.3 变量	3
1.1.4 常量	4
1.1.5 运算符和表达式	6
1.1.6 过程化语句	9
1.2 输入和输出	14
1.2.1 printf 与 scanf	14
1.2.2 I/O 流控制	15
1.3 函数	16
1.3.1 函数原型	16
1.3.2 全局与局部变量	17
1.3.3 函数调用机制	19
1.3.4 递归函数	20
1.3.5 默认参数的函数	21
1.4 引用	22
1.4.1 引用的概念	22
1.4.2 引用的操作	23
1.4.3 用引用传递参数	25
1.4.4 用引用返回值	26
1.4.5 通过引用返回多个值	28
1.5 习题	29
第 2 章 类和对象	30
2.1 类	30
2.1.1 结构与类	30
2.1.2 成员函数及其定义方法	32
2.1.3 类存取控制	35
2.1.4 类公有成员 (public)	36
2.1.5 类私有成员 (private)	37
2.1.6 类保护成员 (protected)	38
2.1.7 友元	38

2.2	构造函数	40
2.2.1	类与对象	40
2.2.2	使用构造函数的必要性	41
2.2.3	构造函数的使用	42
2.2.4	带参数的构造函数	44
2.2.5	多个构造函数	45
2.2.6	默认构造函数	47
2.2.7	复制构造函数	50
2.2.8	构造对象的次序	52
2.3	析构函数	56
2.4	举例	59
2.5	习题	64
第3章	重载	65
3.1	函数重载	65
3.1.1	非成员函数重载	65
3.1.2	成员函数重载	68
3.1.3	默认参数与函数重载	71
3.1.4	参数匹配	72
3.1.5	名字分裂规则	74
3.2	运算符重载	75
3.2.1	运算符重载的必要性	75
3.2.2	运算符用作函数调用	76
3.2.3	重载运算符的规则	76
3.2.4	运算符作为成员函数	77
3.2.5	运算符作为友元函数	80
3.2.6	值返回和引用返回	83
3.3	举例	83
3.4	习题	88
第4章	继承	89
4.1	继承的概念	89
4.2	单一继承	90
4.2.1	派生类的构造	92
4.2.2	作用域的分辨	94
4.2.3	继承与组合	95
4.3	多重继承	96
4.3.1	多重继承的实现方式	96
4.3.2	多重继承的模糊性	98
4.3.3	虚拟继承	98
4.3.4	多重继承的构造顺序	99

4.4	多态性	101
4.4.1	先期联编和迟后联编	101
4.4.2	虚函数	102
4.4.3	函数覆盖	103
4.4.4	空虚函数	105
4.4.5	纯虚函数与抽象类	106
4.5	举例	108
4.6	习题	113
第 5 章	模板	115
5.1	模板的概念	115
5.2	使用模板的必要性	116
5.2.1	函数	117
5.2.2	类	117
5.3	函数模板	118
5.4	重载模板函数	120
5.5	类模板	121
5.6	举例	124
5.7	习题	127
第 6 章	Visual C++程序设计入门	128
6.1	第一个 Windows 应用程序“Hello World!”	128
6.1.1	用 Windows API 编写“Hello World!”程序	128
6.1.2	用 MFC AppWizard 来编写“Hello World!”程序	133
6.2	对话框与常用控件	137
6.2.1	静态控件	137
6.2.2	编辑框	138
6.2.3	复选框和单选按钮	138
6.2.4	列表框和组合框	139
6.2.5	基于对话框的应用程序	140
6.2.6	有模式对话框的使用	144
6.2.7	无模式对话框的使用	148
6.3	图形设备接口 (GDI)	148
6.3.1	设备环境类	148
6.3.2	CPen 类	149
6.3.3	CBrush 类	151
6.3.4	CFont 类	153
6.3.5	CBitmap 类	155
6.3.6	库存的 GDI 对象	157
6.3.7	绘图的坐标系	157
6.4	小结	158

6.5	习题	158
第 7 章	用 Visual C++ 设计常用 Windows 程序	160
7.1	鼠标与键盘	160
7.1.1	鼠标消息的处理	160
7.1.2	键盘消息的处理	163
7.2	单文档与多文档应用程序	168
7.2.1	文档、视图与框架	168
7.2.2	单文档应用程序	169
7.2.3	多文档应用程序	176
7.3	动态链接库	180
7.3.1	动态链接库基础知识	180
7.3.2	编写一个动态链接库	181
7.3.3	在程序中使用动态链接库	183
7.4	小结	187
7.5	习题	188
第 8 章	Visual C++ 与数据库访问	189
8.1	通过 ODBC 访问数据库	189
8.1.1	ODBC 数据源	189
8.1.2	MFC 与 ODBC	190
8.1.3	通过 ODBC 类访问 ODBC 数据源应用	191
8.2	使用 DAO 技术访问数据库	198
8.2.1	MFC 与 DAO	199
8.2.2	通过 DAO 访问数据库应用	202
8.3	小结	208
8.4	习题	208

第1章 从C到C++

C语言是贝尔实验室的Dennis Ritchie在B语言的基础上开发出来的，最初的C语言是1972年在一台DEC PDP-11的计算机上实现的。然而，C语言本身是与硬件无关的。所以，经过二三十年的发展，C语言已经能够用在绝大多数的计算机上了。C语言在各种计算机上的快速推广导致了C语言有很多版本，这些版本之间通常是十分相似但又各不兼容，这对于需要代码跨平台移植的程序员来说是一个严重的问题。为了明确定义与机器无关的C语言，1989年美国国家标准协会制定了C语言的标准版本(ANSI C)。

在实际的使用中，C语言以其独有的特点令其风靡全世界：

- 语言简单、紧凑，一共只有32个关键字。
- 程序书写风格自由，使用方便、灵活。
- 具有丰富的数据类型和多样的运算符。
- C语言可以直接访问内存地址，能够进行位运算，能够用于开发操作系统等较为底层的软件。
- 可移植性好。

当然，C语言也有其不足的地方：

- C语言在编译时对类型的检查较弱，这就使得程序中一些由于类型处理不当导致的错误无法在编译时被发现。
- 在语言结构上对代码重用没有强有力的支持。
- 由于程序的书写风格相对较自由，所以对程序员的要求相对也就比较高，否则在书写程序时很可能引入不易被察觉的错误。
- 随着程序规模的扩大，程序员对程序复杂性的控制缺乏手段。

为了满足书写和管理复杂的大型程序的需要，1980年贝尔实验室的Bjame Stroustrup开始对C语言进行改进和扩充。经过十几年的努力，历经3次修订，终于在1994年制定了ANSI C++标准的草案。以后经过不断完善，成为目前的C++。

C++包含了整个C语言，包括了C语言的全部特征和优点，同时针对C语言的不足之处，增加了一些强有力的机制，其中比较著名的是加入了对面向对象编程(OOP)的完全支持。

C++语言是一种应用较广的面向对象的程序设计语言，使用它可以实现面向对象的程序设计。面向对象的设计与面向过程的设计是有很区别的，面向对象的程序设计是在面向过程的程序设计的基础上一个质的飞跃。而学习C++语言可以较为全面地认识面向对象的特性和实现面向对象的方法。

当你首次学习C++时，总会碰到一些在C语言从未见过的概念，如：类、对象、抽象、封装、继承、多态性、虚函数等等。这些概念是C++所具有的，它们与C++对面向对象程序设计方法的支持和实现有关，我们将在稍后的章节中详细介绍。而对于那些在C语言中似曾相识的基础内容，在C++中也有可能已经发生了变化。对于一个正在从C语言向C++语言过渡的程序员来讲，切不可简单地忽略了这些变化。下面我们就来介绍C++语言的基础知识。

1.1 C++语言的基本要素

1.1.1 字符集与保留字

字符集是构成程序设计语言的最基本元素，在 C++语言中，除了字符型数据的内容之外，其它所有的成分都只能由字符集中的字符构成。C++语言的字符集如表 1-1 所示。

表 1-1 C++语言的字符集

类 别	取 值							
英文字母	A-Z,a~z							
数字字符	0~9							
其他字符	空格	!	#	%	^	&	*	-
	+	=	-	~	<	>	/	\
	'	"	;	.	,	(	)	
	[	]	{	}				

保留字也称关键字，它是预先定义好的标识符，这些标识符对编译程序有着特殊的意义。表 1-2 列出了 C++语言的保留字。ANSI C 规定了 32 个保留字，ANSI C++在此基础上补充了 29 个保留字。

表 1-2 C++语言的保留字

C 中原有 保留字	auto	break	case	char	const
	continue	default	do	double	else
	enum	extern	float	for	goto
	if	int	long	register	return
	short	signed	sizeof	static	struct
	switch	typedef	union	unsigned	void
	volatile	while			
C++ 扩充 保留字	bool	catch	class	const_cast	delete
	dynamic_cast	explicit	false	friend	inline
	mutable	namespace	new	operator	private
	protected	public	reinterpret_cast	static_cast	template
	this	throw	true	try	typeid
	typename	using	virtual	wchar_t	

在程序中不能对保留字进行重新定义，也不能把保留字用作变量名、常量名等普通标识符。

1.1.2 数据类型

程序是以处理数据为目的的，一个程序在设计之前，首先必须描述其算法。在算法的描述过程中，必须对算法中所使用的数据以变量或常量的形式描述出来。每个变量和常量根据

其所代表的数据性质的不同而具有各自的数据类型。

C++的数据类型有基本数据类型和非基本数据类型（又称用户自定义类型）之分。基本数据类型是C++内部预先定义好的数据类型。非基本类型可以根据程序员的需要来定义。

C++中的基本数据类型有字符型、整型、实型和布尔型。除了这些基本类型以外，还有一些用于改变基本类型意义的数据类型修饰符。修饰符有长型（long）、短型（short）、有符号（signed）、无符号（unsigned）四种。表 1-3 列出了 C++中的基本数据类型以及它们的取值范围。

表 1-3 常用基本数据类型

类 型	说 明	长 度	取 值 范 围	备 注
char	字符型	1	-128~127	
unsigned char	无符号字符型	1	0~255	
signed char	有符号字符型	1	-128~127	
int	整型	2	-327 68~32 767	
unsigned int	无符号整型	2	0~65 535	
signed int	有符号整型	2	-32 768~32 767	
short int	短整型	2	-32 768~32 767	
unsigned short int	无符号短整型	2	0~65 535	
signed short int	有符号短整型	2	-32 768~32 767	
long int	长整型	4	-2 147 483 648~2 147 483 647	
unsigned long int	无符号长整型	4	0~4 294 967 295	
signed long int	有符号长整型	4	-2 147 483 648~2 147 483 647	
float	浮点型	4	-3.4×10^{38} ~ 3.4×10^{38}	7 位有效
double	双精度型	8	-1.7×10^{308} ~ 1.7×10^{308}	15 位有效
long double	长双精度型	10	-3.4×10^{4932} ~ 3.4×10^{4932}	19 位有效

在不同的编译环境下，同一个数据类型占有的字节数可能会不同。表 1-3 是以 16 位的编译环境为例的。在 32 位编译环境下，整型一般占 4 字节。其他一些数据类型的长度也可能有一定的变化。要准确的知道某个数据类型的长度，可以使用 sizeof 运算符。

数据类型修饰符 short 只能修饰 int，short int 可以省略为 short。

long 只能修饰 int 和 double，long int 可以省略为 long。

unsigned 和 signed 只能修饰 int 和 char。float 和 double 总是有符号的，不能用 unsinged 来修饰，也没有必要用 signed 来修饰。

1.1.3 变量

变量是存储信息的单元，它对应于某个内存空间。程序能在变量中存取值。

1. 变量定义方式

C++语言的变量定义方法是在变量类型后写上一个或多个变量名，以分号结尾，中间用逗号隔开。例如：

```
int x, y;           //2 个整型变量
long area, width, length;  //3 个长整型变量
```

从上面的例子中不难看出，在同一个变量定义语句中可以同时定义几个同类型变量，但是不同类型的变量显然无法在同一个变量定义语句中实现定义。

2. 变量赋值与初始化

可以使用赋值运算“=”符给变量赋值，可以使用单独的语句给变量赋值，也可以在变量定义语句中给变量赋值。例如：

```
int x, y;  
x = 0;  
y = 1;
```

或者：

```
int x = 0, y;  
y = 1;
```

上面例子中的做法都是可以的。

3. 变量的命名

C++语言中变量命名规则如下：

- (1) 不能是 C++ 的保留字。
- (2) 必须以字符或者下划线开始。
- (3) 中间不能有空格。
- (4) 一般不超过 31 个字符。
- (5) 变量名中只能使用大小写字母、数字和下划线。
- (6) 变量名大小写敏感。即大写和小写字母被认为是不同的字母。

变量名一般都具有描述的功能，所以命名变量时最好用一些有意义的名称，便于程序的阅读和理解。

1.1.4 常量

常量就是指在程序运行的整个过程中其值始终不变的量，也就是直接使用符号表示的值。

1. 整型常量

整型常量是以文字形式出现的整数，分为十进制、八进制和十六进制三种。它们的书写形式见表 1-4。

表 1-4 整型常量形式表

进 制	书 写 形 式	举 例	备 注
10 进制	[+/-]若干个 0~9 的数字	123	整型常量后面可以用后缀字母 L 表示长整型、U 表示无符号型，也可同时使用 L 和 U（大小写无关）
8 进制	[+/-]0 若干个 0~7 的数字	0123	
16 进制	[+/-]0x 若干个 0~9 的数字及 A~F 的字母（大小写无关）	0x123	

2. 实型常量

实型常量有两种表示形式：一般形式和指数形式。

一般形式，例如：123.456，-100.0 等。

指数形式，例如：1.23456E+2 表示 1.23456×10^2 ，-1.000E+2 表示 -1.000×10^2 等。其中字母 E 可以小写。

默认情况下，实型常量是 double 型，如果加上后缀 F（或 f）则为 float 型。

3. 字符常量

字符常量是用单引号括起来的一个字符。对于某些不可显示的字符是无法直接通过键盘输入的，C++提供了一种称为转义序列的表示方法，见表 1-5。

表 1-5 C++预定义的转义序列

字符常量形式	ASCII 码（16 进制）	含 义
\a	07	响铃
\n	0A	换行
\t	09	水平制表符
\v	0B	垂直制表符
\b	08	退格
\r	0D	回车
\\	5C	字符\"
\'	27	单引号
\"	22	双引号

同时，C++还提供了通过 ASCII 码的 8 进制形式或 16 进制形式的转义表示方法，例如：

\nnn 8 进制形式 \015（回车键）

\xnn 16 进制形式 \x61（字母'A'）

4. 字符串常量

字符串常量简称字符串，是用一对双引号括起来的字符序列。字符串常量和字符常量是不同的。在 C 和 C++中字符串常量通常由一个 '\0' 字符结束。例如：字符串"HELLO"，表面看来这个字符串由 5 个字符构成，而实际这个字符串在内存中需要占用 6 个连续的存储单元，如图 1-1 所示。

H	E	L	L	O	\0
---	---	---	---	---	----

图 1-1 字符串的存储方式

通过上面的例子不难理解，字符常量'0'和字符串常量"0"的含义和在内存中占用的容量都是不同的。

5. 布尔常量

布尔类型常量只有两个：true（真）和 false（假）。布尔类型常量在 ANSI C 中是没有的。

1.1.5 运算符和表达式

1. 表达式

表达式是构成程序语句的基础，是一组运算符、操作数和标点符号的序列。表达式可以用如下的方式来定义：

(1) 一个变量或对象（本书后续部分介绍）是一个最简单的表达式，其值是常量或对象的值。

(2) 一次函数调用是一个表达式。

(3) 用运算符连接起来的多个表达式构成一个更复杂的表达式，即所谓的表达式可以嵌套。

(4) 一般情况下，表达式根据其中的运算符的优先级和结合性进行计算，最后得到该表达式的值。少数表达式不产生值。

由于表达式本身往往由多个运算符连接而成，所以，从某种意义上讲，了解运算符可能比了解表达式本身更重要。

2. 算术运算符

表 1-6 列出了 C++ 中的算术运算符及其功能说明。

表 1-6 算术运算符

运 算 符	功 能	举 例
-	负号	-1, -1.23
+	加	x=1+2; x 的结果是 3
-	减	x=2-1; x 的结果是 1
*	乘	x=2*3; x 的结果是 6
/	除	x=1/2; x 的结果是 0（如果是 2 个整数相除，结果取整数部分）
%	取余	x=10/3; x 的结果是 1
++	自增	x++; x 的结果是 x+1
--	自减	x--; x 的结果是 x-1

值得提出的是自增和自减运算是使用方便且效率很高的两个单目运算符。这两个单目运算符都有前置和后置两种使用形式。当自增和自减运算符用于单一运算的时候效果是一样的，但是如果运算的结果还要继续参与其他计算的话，情况就完全不同了。运算符前置表示先运算后取值，运算符后置表示先取值后运算。例如：

```

x++;           //表示 x+1
++x;           //表示 x+1
y = x++;       //如果运算前 x=1，运算后 y=1
y = ++x;       //如果运算前 x=1，运算后 y=2

```

3. 赋值运算符

C++ 提供了两类赋值运算符，一类是简单赋值运算符，另一类是复合赋值运算符。简单赋值运算符就是“=”，复合赋值运算符是赋值运算符和算术运算和位运算结合而成的，共 10 个，它们是：+=，-=，*=，/=，%=，<<=，>>=，&=，^=，|=。下面通过几个例子来说明赋