

普通高等教育



计算机系列规划教材

C/C++ 程序设计 (面向过程)

★ 黄文钧 主编 ★

★ 谢宁新 刘美玲 梁 艺 副主编 ★



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



高等教育计算机系列规划教材

C/C++程序设计

(面向过程)

黄文钧 主 编

谢宁新 刘美玲 梁 艺 副主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

C/C++语言是一种通用的程序设计高级语言，C++兼容了C，并将C作为自己的前半部分——“面向过程程序设计”部分，后半部分是“面向对象程序设计”部分，本书将前半部分分离出来，单独成册，作为“C语言程序设计”或“C++面向过程程序设计”的教材。

本书共8章，内容包括第1章描述C++程序设计语言的发展、C++应用程序的开发过程等；第2章介绍C++处理的数据类型、C++使用的运算符以及表达式格式；第3章主要介绍C++结构化程序设计的三种结构：顺序结构、选择结构、循环结构；第4章~第6章分别介绍函数、数组和指针；第7章介绍结构体和共用体；第8章介绍文件的输入与输出。本书最后有附录，列出C/C++常用字符表、ASCII码表、运算符优先级表等。

本书可作为理工科大学计算机类专业、信息类相关专业的本科教材或专科教材；也很适合初学者自学。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

C/C++程序设计：面向过程 / 黄文钧主编. —北京：电子工业出版社，2016.7

普通高等教育计算机系列规划教材

ISBN 978-7-121-29338-2

I. ①C… II. ①黄… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第156135号

策划编辑：徐建军 (xujj@phei.com.cn)

责任编辑：徐建军 特约编辑：方红琴 俞凌娣

印 刷：三河市良远印务有限公司

装 订：三河市良远印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本：787×1092 1/16 印张：12.5 字数：320千字

版 次：2016年7月第1版

印 次：2016年7月第1次印刷

印 数：3000册 定价：29.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 zllts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：(010) 88254570。

前言

Preface

C/C++语言是一种通用的程序设计高级语言，内容丰富，功能强大，十分流行。C++兼容了C，并将C作为自己的前半部分——“面向过程程序设计”部分，后半部分是“面向对象程序设计”部分，C++将这两部分内容有机联系在一起，成为完整体系。由于“面向过程程序设计”部分就是完整的C内容，可以独立成册。

本书是为“C语言程序设计”和“C++面向过程程序设计”课程编写的教材，其内容选取符合教学大纲要求，同时兼顾了学科的广度和深度，适用面广。

本书主要面向高校各专业大学生，考虑到教师教学和学生自习或自学的便利，本书的内容按知识延伸和逻辑顺序进行编排，本书的每一章都附有足量的例题和习题。

本书列出的程序或例题是面向教学的，是为了帮助读者更好地理解和掌握相关概念而专门设计的，并不一定就是实际应用的程序。而一些用于实际应用的程序，往往篇幅过长，不一定适合教学。作为教学的程序，基本上对问题做了简化，尽量压缩不必要的语句，可能有些程序在专业人士看来很“幼稚”，但对于学习者而言可能就是很好的教学程序。

由于C/C++语法复杂，内容多，如果读者对它缺乏系统的了解，将难以正确应用，编写出来的程序将会错误百出，计算机编译出错，自己检查多遍仍然发现不了错误之处，事倍功半。因此，在校大学生在教师的指导下学习本书，要切实记住C/C++的语法、规则、关键字及其意义和属性、运算符及其意义和属性等基本知识，充分理解基本概念，清楚辨别相似概念的差别，初步掌握程序设计方法，能够利用C/C++编写相对简单的程序，解决一些简单问题，为以后进一步学习和应用打下坚实基础。在校大学生课后要多复习，多研读例题，对其中的各知识点要有充分的认识。此外，还要多做习题，在计算机上检验，加深自己对知识的理解。在充分理解和掌握了本书的基本知识和基本方法之后，再结合其他知识，例如“数据结构”和“计算机算法设计与分析”，可以尝试编写稍微复杂的程序，解决稍微复杂的问题。

本书由广西民族大学信息科学与工程学院的教师精心组织策划，由黄文钧担任主编，由谢宁新、刘美玲、梁艺担任副主编，参加本书编写的还有周永权、何建强、张超群、韦艳艳、汤卫东、李熹、张纲强、李香林、文勇、廖玉奎、朱健、黄帆、林国勇。在本书编写的过程中得到了学院领导的大力支持，在此一并表示感谢！

为了方便教师教学，本书配有电子教学课件，请有此需要的教师登录华信教育资源网(www.hxedu.com.cn)注册后免费进行下载，有问题时可在网站留言板留言或与电子工业出版社联系(E-mail: hxedu@phei.com.cn)。

由于编者水平有限，加之时间仓促，书中难免有疏漏之处，敬请广大读者批评指正。

编者

目录

Contents

第1章 概述	(1)
1.1 程序设计语言的发展	(1)
1.1.1 机器语言	(2)
1.1.2 汇编语言	(2)
1.1.3 高级语言	(2)
1.2 C++语言的发展	(3)
1.3 程序设计	(4)
1.4 C++程序简介	(4)
1.4.1 C++程序举例	(4)
1.4.2 C++程序的结构	(8)
1.5 C++程序的开发过程	(9)
1.5.1 数据库相关概念	(9)
1.5.2 C++程序的开发过程	(10)
1.6 本章小结	(16)
习题一	(16)
第2章 基本数据类型、运算符和表达式	(19)
2.1 C++语言的字符集和词汇	(19)
2.1.1 字符集	(20)
2.1.2 C++语言词汇	(20)
2.2 C++数据类型	(21)
2.3 常量	(23)
2.3.1 常量的定义	(23)
2.3.2 数值常量	(24)
2.3.3 字符常量和字符串常量	(25)
2.3.4 符号常量	(28)
2.4 变量	(29)
2.4.1 什么是变量	(29)

2.4.2	定义和使用变量	(29)
2.4.3	常量	(30)
2.5	运算符和表达式	(31)
2.5.1	算术运算符和算术表达式	(31)
2.5.2	赋值运算符和赋值表达式	(33)
2.5.3	逗号运算符和逗号表达式	(35)
2.5.4	位运算符和位运算表达式	(36)
2.5.5	求字节运算符	(38)
2.6	数据类型转换	(39)
2.6.1	不同类型数据混合运算时的类型转换规则	(39)
2.6.2	赋值时的类型转换规则	(40)
2.6.3	强制类型转换	(42)
2.7	本章小结	(42)
	习题二	(43)
第3章	结构化程序设计	(46)
3.1	算法	(46)
3.1.1	算法的概念和特点	(47)
3.1.2	算法的表示	(47)
3.2	结构化程序设计概述	(48)
3.2.1	结构化程序设计的概念	(48)
3.2.2	结构化程序设计的3种基本结构	(49)
3.3	C++语句	(50)
3.4	C++的输入与输出	(52)
3.4.1	输入/输出概述	(52)
3.4.2	输入/输出流的基本操作	(52)
3.4.3	在输入流与输出流中使用控制符	(55)
3.4.4	用 getchar 和 putchar 函数输入和输出字符	(57)
3.4.5	用 scanf 和 printf 函数进行输入和输出	(59)
3.5	顺序结构程序设计	(62)
3.6	关系运算和逻辑运算	(63)
3.6.1	关系运算符和关系表达式	(63)
3.6.2	逻辑常量和逻辑变量	(64)
3.6.3	逻辑运算符和逻辑表达式	(65)
3.7	选择结构	(66)
3.7.1	if 语句	(67)
3.7.2	if 语句的嵌套	(70)
3.7.3	条件运算符和条件表达式	(72)
3.7.4	switch 语句	(72)
3.7.5	选择结构程序设计	(74)
3.8	循环结构	(75)

3.8.1	while 语句	(75)
3.8.2	do-while 语句	(77)
3.8.3	for 语句	(79)
3.9	循环的嵌套	(81)
3.10	跳转语句	(82)
3.10.1	goto 语句	(82)
3.10.2	break 语句	(83)
3.10.3	continue 语句	(83)
3.11	本章小结	(87)
	习题三	(87)
第 4 章	函数	(91)
4.1	概述	(91)
4.2	引例	(92)
4.3	函数定义的一般格式	(93)
4.4	函数调用与函数声明	(94)
4.5	形式参数和实际参数的关系	(95)
4.6	内置函数	(98)
4.7	函数的重载	(98)
4.8	函数模板	(99)
4.9	函数的嵌套调用和递归调用	(99)
4.10	局部变量和全局变量	(100)
4.11	变量的存储类别	(101)
4.12	变量声明	(101)
4.12.1	auto 变量	(101)
4.12.2	用 static 声明局部变量	(102)
4.12.3	register 变量	(103)
4.12.4	用 extern 声明外部变量	(103)
4.13	本章小结	(104)
	习题四	(104)
第 5 章	数组	(106)
5.1	一维数组的定义和引用	(106)
5.1.1	一维数组的定义格式	(106)
5.1.2	一维数组元素的引用	(107)
5.1.3	一维数组的初始化	(108)
5.1.4	一维数组程序举例	(109)
5.2	二维数组的定义和引用	(109)
5.2.1	二维数组的定义	(109)
5.2.2	二维数组元素的引用	(110)
5.2.3	二维数组的初始化	(111)
5.2.4	二维数组程序举例	(112)

5.3 字符数组	(113)
5.3.1 字符数组的定义	(113)
5.3.2 字符数组的初始化	(113)
5.3.3 字符数组的引用	(114)
5.3.4 字符串和字符串结束标志	(114)
5.3.5 字符数组的输入输出	(115)
5.3.6 字符串处理函数	(116)
5.4 程序举例	(118)
5.5 本章小结	(121)
习题五	(121)
第6章 指针	(123)
6.1 指针的概念	(123)
6.2 指针变量的概念	(123)
6.3 指针变量的定义	(124)
6.4 与指针有关的基本操作	(124)
6.5 通过指针变量访问变量	(126)
6.6 指针变量作为函数参数	(127)
6.7 指针变量与一维数组	(129)
6.8 指针与字符串	(130)
6.9 指向一维数组的指针变量	(131)
6.10 指针与二维数组	(131)
6.11 多级指针与指针数组	(132)
6.12 返回指针的函数	(135)
6.13 函数指针	(135)
6.14 关于指针若干概念的总结	(137)
6.15 引用	(139)
6.16 本章小结	(140)
习题六	(140)
第7章 结构体与共用体	(142)
7.1 定义结构体的一般形式	(142)
7.1.1 结构体类型变量的说明	(143)
7.1.2 访问结构体变量的元素	(145)
7.1.3 结构体变量的赋值	(145)
7.1.4 结构体类型的数组	(146)
7.2 指向结构体类型变量的指针	(147)
7.3 结构类型指针变量作为函数参数	(149)
7.4 动态存储分配	(150)
7.5 链表的概念	(152)
7.6 链表的设计	(153)
7.7 共用体	(154)

7.8 枚举	(157)
7.9 类型定义符 typedef	(160)
7.10 本章小结	(161)
习题七	(161)
第 8 章 文件的输入和输出	(165)
8.1 文件的概念	(165)
8.2 文件指针	(166)
8.3 文件的打开与关闭	(167)
8.3.1 文件打开函数 fopen	(167)
8.3.2 文件关闭函数 fclose	(168)
8.4 文件的读写	(169)
8.4.1 字符读写函数 fgetc 和 fputc	(170)
8.4.2 字符串读写函数 fgets 和 fputs	(173)
8.4.3 数据块读写函数 fread 和 fwrite	(174)
8.4.4 格式化读写函数 fscanf 和 fprintf	(176)
8.5 文件的随机读写	(177)
8.5.1 文件定位	(177)
8.5.2 文件的随机读写	(178)
8.6 文件检测函数	(179)
8.7 本章小结	(180)
习题八	(180)
附录 A 常用字符的 ASCII 码对照表	(184)
附录 B C++运算符的优先级和结合性	(186)
参考文献	(187)

第1章

概述

➔ 本章学习目标

- 了解程序设计语言的发展历程。
- 了解 C++ 语言的发展历程和特点。
- 了解 C++ 程序的结构。
- 熟悉 C++ 程序的开发过程。

1.1 程序设计语言的发展

语言是由词汇按照一定的语法规则构成的一个符号系统。语言是人类进行思维和传递信息的工具，通过语言，人们可以交流思想。程序设计语言（Programming Language）是人与计算机交流的工具，是人们为描述问题的解决过程而设计的一种具有语法语义描述的记号，它是由词汇、词法和语法规则构成的符号系统。程序设计者使用计算机可以识别的程序设计语言来描述解决问题的步骤和方法，以命令计算机完成各项工作。

计算机系统包括硬件和软件两大部分。硬件是构成计算机的所有实体部件的集合。软件包括程序、数据及其有关的文档资料。计算机要正常工作，除了构成计算机各个组成部分的物理部件之外，还必须要有指挥计算机“做什么”和“怎么做”的程序，程序控制着计算机的工作。程序是指令的集合。而指令是指示计算机执行某种操作的命令，计算机可以识别的指令是由 0 和 1 组成的一串代码。

从世界上第一台电子计算机诞生至今，程序设计语言随着计算机技术的进步而不断发展，大致经历了三代，它们分别是机器语言时代、汇编语言时代和高级语言时代。程序设计语言按照语言级别可以分为低级语言和高级语言，其中，低级语言有机器语言和汇编语言。

1.1.1 机器语言

计算机刚诞生时，程序设计者使用的程序设计语言是机器语言。机器语言是由计算机硬件系统可以识别的二进制指令组成的语言。机器语言所有指令的记号都采用符号 0 和 1 的编码组成。例如，计算 3+5 的机器语言程序如下。

```
10110000 00000011    //将 3 送往累加器
00000100 00000101    //将 5 与累加器中的 3 相加，结果保留在累加器中
```

机器语言指令是计算机可以直接识别的指令，因此执行效率高，但是用机器语言编写程序非常困难、费时费力、编程效率低、易出差错。另外，不同计算机的机器语言是不相同的，因此，用机器语言编写的程序在不同的计算机上不能通用，程序的可移植性差。这样，当要把一个程序在另外类型的计算机系统上运行时，就需要重新编写程序代码。

1.1.2 汇编语言

机器语言晦涩难懂，不易于学习与使用。因此，为了克服机器语言的缺点，人们使用能反映指令功能的助记符来表示机器语言中的指令，称之为汇编语言。

例如，计算 3+5 的汇编语言程序如下。

```
MOV AL,03H    //将十六进制数 3 送往累加器
ADD AL,05H    //将十六进制数 5 与累加器中的 3 相加，结果保留在累加器中
```

显然，和使用机器语言编写程序比较，使用汇编语言编写程序要容易许多。当然，计算机不能直接理解和执行用汇编语言编写的程序，需要进行转换。汇编语言和机器语言基本上是一一对应的。也就是说，对大多数汇编语言中的指令来说，在机器语言中都存在一条功能相同的机器指令。因为汇编语言的指令和机器语言的机器指令存在对应关系，所以这样的转换并不困难。汇编程序就是完成这种转换工作的一种专门的程序。汇编程序是把用汇编语言编写的程序翻译为等价的机器语言程序的一种程序。

1.1.3 高级语言

机器语言和汇编语言是面向机器的语言，随机器而异，不同计算机系统的机器指令与汇编指令不同。为了克服这种缺点，人们开始研究新的程序设计语言。20 世纪 50 年代中期，世界上第一种高级语言——FORTRAN 语言诞生了，它主要用于科学和工程计算，它标志着高级语言的到来。随着计算机及其应用的发展，先后出现了多种高级语言，如 ALGOL、COBOL、BASIC、Pascal、C 等。

高级语言不再面向具体机器，而是面向解题过程，人们可以用接近自然语言和数学语言对操作过程进行描述，高级语言的表示方法比低级语言更接近于待解决问题的表示方法。高级语言的诞生使程序员不必熟悉计算机内部具体构造和熟记机器指令，而把主要精力放在算法描述上面即可。因此，和汇编语言相比，高级语言的抽象度高，易学、易用、易维护，在一定程度上与具体机器无关，求解问题的方法描述直观。

例如，计算 3+5 的 C++ 语言程序如下。

```
int sum;           //定义整型变量 sum
sum=3+5;         //将 3 与 5 的和赋值给 sum
cout<<"3+5="<<sum<<endl; //输出 3+5=8
```

按照描述问题的方式进行分类,高级语言可以分为面向过程的语言和面向对象的语言, FORTRAN、ALGOL、COBOL、BASIC、Pascal、C 等都属于面向过程的语言。程序设计是把现实世界中的问题抽象后利用计算机语言转化到机器能够理解的层次,并最终利用机器来寻求问题的解。面向过程的程序设计思想是用计算机能够理解的逻辑来描述和表达待解决的问题及其具体的解决过程,它关注的是解决问题的算法。数据结构与算法是面向过程问题求解的核心,它将问题域中要处理的对象具有的属性与对属性操作的方法分离。因此,面向过程的程序设计语言在描述包含多个相互关联过程的大型系统时非常复杂、困难。

面向对象的程序设计语言把现实世界中的事物称作对象,每个对象由一组属性和一组行为组成,并将同一类对象的共同属性和行为抽象成类。例如,“学生”这个类别的对象的属性就有学号、姓名、年龄、性别等,“学生”这个类别的对象的行为就有注册、登记考试成绩等。Smalltalk、C++、Java 等都属于面向对象语言。面向对象语言是目前最为流行的程序设计语言。

1.2 C++语言的发展

20 世纪 60 年代,剑桥大学的 Martin Richards 开发了 BCPL 语言。1970 年,美国贝尔实验室的 Ken Thompson 对 BCPL 语言进行了改进,开发了 B 语言。1972 年,美国贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 在 B 语言的基础上,设计出了 C 语言。它不是为初学者设计的,而是为计算机专业人员设计的。最初,C 语言是写 UNIX 操作系统的一种工具,在贝尔实验室内部使用。后来 C 语言不断改进,人们发现它简洁、使用灵活方便、可移植性好、功能强大,它既具有高级语言的特点,又具有汇编语言的特点,后来大多数系统软件和许多应用软件都用 C 语言编写。C 语言的出现是程序设计语言的一个重要里程碑。

C 语言具有如下优点。

- (1) 语言简洁紧凑,使用灵活方便。C 语言有 32 个关键字和 9 种控制语句,程序书写形式自由。
- (2) 运算符丰富。C 语言有 34 种运算符,运用这些运算符可以构成简洁而功能强大的表达式。
- (3) 数据结构类型丰富。C 语言具有基本数据类型——整型、实型和字符型;构造类型——数组、结构体、共用体等。另外,C 语言还提供了指针,有助于构造链表、树、栈、图等复杂数据结构。
- (4) 生成的目标代码质量高,程序执行效率高。
- (5) 可移植性好。用 C 语言编写的程序可移植性好,在一个环境下运行的程序不加修改或少许修改就可以在完全不同的环境下运行,提高程序的开发效率。

然而,随着软件规模的不断扩大,用 C 语言编写程序渐渐显得有些吃力了。C 语言是结构化的语言,当问题比较复杂、程序模块关联度大时,用结构化的程序设计语言编写程序就显得比较困难。

为了解决软件危机问题,人们迫切地需要发展新的程序设计方法,20 世纪 80 年代出现了

面向对象的程序设计思想与方法。1980年，美国贝尔实验室的Bjarne Stroustrup等人对C语言进行了改进和扩充，并引入了类的概念，把新的语言称为“带类的C”。1983年，由Rich Maseitti提议正式命名为C++。C++保留了C语言原有的所有优点，增加了面向对象的机制。

C++是由C语言发展而来的，C++是C语言的超集，与C语言兼容。用C语言编写的程序略加修改或不加修改就可以在C++的编译系统下运行或调试。C++既可用于面向过程的结构化程序设计，又可用于面向对象的程序设计，是一种功能强大的混合型程序设计语言。

面向对象程序设计是针对开发较大规模程序而提出来的，目的是提高软件开发的效率。不要把面向对象和面向过程对立起来，面向对象和面向过程不是矛盾的，而是各有用途、互为补充的。学习C++，既要会利用C++进行面向过程的结构化程序设计，也要会利用C++进行面向对象的程序设计。本书介绍C++在面向过程程序设计中的应用，C++面向对象程序设计的内容在另一册书中介绍。

1.3 程序设计

根据要解决的问题的工作步骤，使用某种程序设计语言描述出来称为程序设计。程序设计一般分为两个过程：对数据进行描述和对数据进行处理。

1. 数据描述

数据描述是指把被处理的信息描述成计算机可以接受的数据形式。例如，把被处理的信息描述成整数、实数、字符串等；也可以把被处理的信息描述成图形、声音等。

2. 数据处理

数据处理是指对数据进行输入、输出、整理、计算、存储、维护等一系列的活动。数据处理的过程要用某一种程序设计语言描述出来，即编写程序实现问题的求解。

使用程序设计语言解决问题的基本步骤如下。

- (1) 分析问题。
- (2) 确定计算或处理方法。
- (3) 描述算法：把求解问题的操作步骤描述出来。
- (4) 编写、编译、连接、执行和调试程序。

1.4 C++程序简介

1.4.1 C++程序举例

【例 1.1】 在屏幕上显示一行文字。

```
/* 文件名: egl_1.cpp
程序功能: 在屏幕上显示一行文字
*/
#include <iostream> //包含头文件 iostream
using namespace std; //使用命名空间（也称名称空间）
int main() //主函数首部
```

```

{ //函数体开始
    cout << "Hello world!\n"; //输出 Hello world!
    return 0; //向操作系统返回数值 0
} //主函数定义结束

```

程序运行后会在屏幕上输出一行字符：

```
Hello world!
```

这个程序由注释语句、预处理命令和主函数构成。

(1) 注释语句。

为了增加程序的可读性，一个好的程序可以在源程序中加上必要的注释，对程序进行注解和说明。在 C++ 程序的任何位置都可以加上注释信息。C++ 的注释有两种类型：一种是行注释；另一种是块注释。其中，行注释以 // 开头，从 // 开始到本行末尾的所有内容都是注释，不能跨行，形如：

```
//注释内容
```

块注释以 /* 开始，以 */ 结束，/* 和 */ 之间的所有内容都是注释，可以跨越多行，形如：

```
/* 注释内容 */
```

因此，一般习惯是：内容较少的简单注释常用 “//”，内容较长的常用 “/*.....*/”。

(2) 预处理命令。

以 “#” 开头的行称为预处理命令。由于程序中要用到输入输出流，所以要包含文件 `iostream`，这个文件包含了与程序输入和输出操作有关的信息。由于这类文件在程序预编译阶段被嵌入在程序的开始处，所以称之为头文件。`cout` 是 C++ 系统预定义的输出流对象，它和插入运算符 “<<” 结合使用，作用是将输出流对象 `cout` 中的内容输出到指定的输出设备（一般为显示器）。在这个程序中，`cout` 的作用是将插入运算符 “<<” 右边的字符串输出到屏幕上。’\n’ 是一个转义字符，它的作用是换行。

(3) 命名空间。

ANSI (American National Standards Institute, 美国国家标准协会) 和 ISO (International Standards Organization, 国际标准化组织) 于 1998 年联合制定了 C++ 语言的标准，以下称为标准 C++。命名空间是标准 C++ 为了解决程序中的全局标识符、编译器系统库中的标识符与第三方类及函数库中的标识符之间的同名冲突而采用的一种机制。

标准化之前的头文件是带后缀名的文件，使用 C 语言的传统方法。标准化之前包含头文件的方法如下：

```
#include <iostream.h> //头文件带后缀名.h
```

标准化之后的头文件是不带后缀名的文件，包含头文件的新方法如下：

```
#include <iostream> //头文件不带后缀名.h
using namespace std; //使用命名空间 std
```

其中，`std` 是标准 C++ 定义的命名空间，C++ 标准库中所有类、对象与函数等都定义在该命名空间中。

说明：标准 C++ 规定用户使用新版头文件，但一些编译器（如 Visual C++ 6.0）依然支持包含头文件的旧方法，以向下兼容。而另一些编译器（如 Visual Studio 2008）则遵循 ANSI C++ 的标准，因此需要使用新方法。本书的程序均遵循 C++ 的标准。

另外，为了和 C 语言兼容，C++ 标准化过程中，原有 C 语言头文件标准化后，头文件名前带个 c 字母，如 `cstdio`、`cmath`、`cstdlib`、`cstring` 等。传统方法与新方法的对比如下：

C 传统方法	C++ 新方法
<code>#include <stdio.h></code>	<code>#include <cstdio></code>
<code>#include <math.h></code>	<code>#include <cmath></code>
<code>#include <stdlib.h></code>	<code>#include <cstdlib></code>
<code>#include <string.h></code>	<code>#include <cstring></code>
	<code>using namespace std;</code>

(4) 主函数 `main()`。

`main` 是主函数的名字，由小写字母构成。注意，在 C++ 中，字母是区分大小写的。一个 C++ 程序可以由一个或多个函数组成，但必须有且只能有一个主函数，它是程序执行的入口，不管主函数在整个程序中的位置如何，都会从主函数开始执行。`main` 前面的 `int` 的作用是将 `main()` 函数的返回值类型声明为整型。

函数体用一对大括号括起来，以 “{” 开始，以 “}” 结束。每个函数都可以由若干语句组成，每条语句都以 “;” 结束。

注意：C++ 标准规定，`main` 函数的首部应该写成 `int main()`，表示 `main` 函数的返回值为整型，如果返回值为 0，则表示程序正常结束，否则表示程序异常结束。C++ 标准虽然不允许定义 `void main()`，但是在一些编译器中（如 VC++ 6.0）可以通过编译，然而并非所有编译器都支持 `void main()`。例如，在 GCC 3.2 编译器中，如果 `main` 函数的返回值不是 `int` 类型，则无法通过编译。所以，建议写成 `int main()`。

【例 1.2】 从键盘输入两个整数，求这两个整数的和。

```
#include <iostream>           //包含头文件 iostream
using namespace std;         //使用命名空间（也称名称空间）
int main()                   //主函数首部
{                             //函数体开始
    int a,b,sum;             //定义 3 个整型变量 a, b, sum
    cout<<"请输入两个整数: "; //输出语句
    cin>>a>>b;               //输入语句，从键盘获取两个整数
    sum=a+b;                 //赋值语句，将 a、b 的和赋值给 sum
    cout<<"a+b="<<sum<<endl; //输出语句
    return 0;                //程序正常结束，向操作系统返回 0
}
```

`main` 函数的第一行定义了 3 个整型变量。在 C++ 中，变量要先定义再使用。定义变量后，系统为这些变量分配内存空间，用于存储变量的值。`cin` 是 C++ 系统预定义的输入流对象，它与提取运算符 “>>” 结合使用，作用是从输入设备（如键盘）提取数据送到指定的变量中。“`sum=a+b;`” 是一条赋值语句，其作用是将 `a` 与 `b` 的和赋给变量 `sum`。函数体的第 5 行是先输出字符串 “`a+b=`”，再输出变量 `sum` 的值。`endl` 是输出操纵符，具有换行作用。

程序运行结果如下。

```
请输入两个整数: 5 7↵
a+b=12
```

注意：输入数据时，数据间用空格或回车分隔，输入完毕按回车键确认。

【例 1.3】 从键盘输入两个整数，利用独立的函数求这两个数中的大者。

```

#include <iostream> //包含头文件 iostream
using namespace std; //使用命名空间
int main() //主函数
{
    int max(int x,int y); //函数声明
    int a,b,m; //变量声明
    cout<<"请输入两个整数: ";
    cin>>a>>b;
    m=max(a,b); //调用 max 函数，将得到的值赋给 m
    cout<<"max="<<m<<"\n"; //输出大数 m 的值
    return 0;
}
/*
定义 max 函数，函数返回值为整型，形式参数 x 和 y 为整型
*/
int max(int x,int y)
{ //max 函数体开始
    int z;
    if(x>y) z=x; //if 语句，如果 x>y，则将 x 的值赋给 z
    else z=y; //否则，将 y 的值赋给 z
    return(z); //将 z 的值返回，通过 max 带回调用处
}

```

该程序由 main()和 max()两个函数组成。程序从 main()开始执行，在 m=max(a,b);行程序转向执行函数 max()，函数 max()接收两个整型参数，求出这两个整数中的大者并通过 return 语句返回结果值。

main 函数是一个特殊的函数，它是程序的入口。操作系统通过调用它开始执行程序。其他函数是在程序的运行过程中，由 main 函数或另外一些函数调用。程序按照源代码中的次序逐行执行，直到调用一个函数，此时程序转向执行被调用函数。当函数调用结束后，它把控制权交给函数调用语句的下一条语句。关于函数的概念与用法将在第 4 章详细介绍。

程序运行结果如下：

```

请输入两个整数：23 56✓
max=56

```

【例 1.4】 编写包含类的 C++程序，用于计算矩形的面积。

```

#include <iostream> //包含头文件 iostream
using namespace std; //使用命名空间
class Rectangle //声明矩形类
{ private:
    int width,height; //数据成员
public:
    void setWidth(int w) //成员函数
    { width=w; }
    void setHeight(int h) //成员函数

```



```

        {   height=h; }
        int area( )           //成员函数
        {   return width*height;   }
}; //类定义结束
Rectangle rect;           //定义矩形对象 rect
int main( )
{
    rect.setWidth(10);     //调用对象 rect 的成员函数设定矩形的宽
    rect.setHeight(6);    //调用对象 rect 的成员函数设定矩形的高
    int s=rect.area( );   //通过对象 rect 计算矩形面积
    cout<<"矩形的面积为: "<<s<<endl; //输出矩形面积
    return 0;
}

```

程序运行结果：

矩形的面积为：60

这是一个包含类的 C++ 程序。在客观世界中，对象是具有某种属性和行为的事物。而在面向对象程序设计中，同类对象的共同属性与行为构成一个类，任何对象都属于某个类。一个类包含两种成员：数据成员和成员函数，其中，数据成员用于描述对象的属性，成员函数用于描述对象的行为。也就是说，一个类是由一组数据以及对其操作的函数组成的。

1.4.2 C++程序的结构

通过以上几个程序可以看出，一个 C++ 程序单位可以包括预处理命令、主函数、其他函数和类等，其结构如下。

```

#include <>           //预处理命令，可以有多条预处理命令
#include <>
using namespace std; //使用命名空间
全局声明部分
函数定义
类定义

```

对 C++ 程序的结构说明如下。

(1) 一个 C++ 程序可以由一个程序单位或多个程序单位构成。每一个程序单位为一个文件。在程序编译时，编译系统分别对各个文件进行编译，因此，一个文件是一个编译单元。上节介绍的几个例子都是比较简单的程序，都是只由一个程序单位（即一个文件）构成的。

(2) 在一个程序单位中，可以包括以下几个部分。

① 预处理命令。`#include` 就是一个预处理命令，其作用是通知编译程序将指定的头文件包含到本程序中。头文件的内容一般包括类型声明、函数声明、全局变量定义、宏定义等，还可以包含其他头文件。上节 4 个程序中也都包括 `#include` 命令。

② 全局声明部分（在函数外的声明部分）。在这部分中包括对用户自己定义的数据类型的声明和程序中所用到变量的定义。例 1.4 中对矩形类 `Rectangle` 的声明以及对象 `rect` 的声明都属于全局声明部分。

③ 函数。函数是实现操作的部分，每个程序都可以包括一个或多个函数，其中必须有一