

© 21 世纪高等学校教材



C/C++程序设计

(第2版)

吴国凤 宣善立 主编



高等教育出版社
Higher Education Press

21 世纪高等学校教材

C/C++程序设计

(第2版)

吴国凤 宣善立 主 编

高等教育出版社

内容提要

本书为《C/C++程序设计》第2版,全面地讲述了C/C++语言程序设计的基础知识、程序设计方法和解决实际问题的技巧。全书内容丰富、结构精练、讲解深入浅出、实用性强,并将知识点融入各章典型例题中,同时也注重培养学习者的计算机程序设计能力和良好的程序设计习惯。

全书共分为12章,主要内容包括:C语言概述、基本程序设计、程序控制结构、数组、函数、指针、结构体与共用体、位运算、文件、C++基础知识、面向对象程序设计及应用程序实例等。每章精选了多种类型的案例和习题,并有配套的《C/C++程序设计实验指导与题解》(第2版)一书,帮助学习者开阔思路和提高程序设计能力。

本书可作为高等院校理工科各专业C/C++程序设计课程教材,也可作为计算机等级考试的参考用书。

本书电子课件及案例源代码等相关教学资源均可以从中国高校计算机课程网上下载,网址为:<http://computer.cncourse.com>。

图书在版编目(CIP)数据

C/C++程序设计/吴国风,宣善立主编.—2版.—北京:

高等教育出版社,2009.9

ISBN 978-7-04-028054-8

I. C… II. ①吴…②宣… III. C语言—程序设计
IV. TP312

中国版本图书馆CIP数据核字(2009)第127498号

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100120
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京明月印务有限责任公司

开 本 787×1092 1/16
印 张 24
字 数 580 000

购书热线 010-58581118
咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2006年6月第1版
2009年9月第2版
印 次 2009年9月第1次印刷
定 价 28.30元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 28054-00

第2版前言

当今信息化时代,大学生不仅要掌握计算机基础知识,而且要学会使用计算机进行程序设计。C/C++语言以其功能丰富、表达能力强、应用面广等特点,深受广大程序开发人员的青睐,因而也成为各高校计算机专业和非计算机专业的一门必修课。通过学习,让学生不仅掌握程序设计语言的知识、编程技术、调试方法和基本算法等,更重要的是在实践中逐步掌握程序设计的思想和方法,具备利用计算机求解实际问题的能力,并能灵活应用程序设计语言解决专业领域中的具体问题。为了满足程序设计基础教学改革的需要,我们组织了长期从事计算机程序设计课程教学,具有丰富教学实践经验的老师编写了《C/C++程序设计》一书,本书为修订后的第2版,供广大高校的师生们选用。为适合各类学生的学习和掌握,我们在编写教材的同时,一并编写了配套的实验教材《C/C++程序设计实验指导与题解(第2版)》、CAI课件、电子教案及相应的程序设计素材,同时也开发了C/C++程序设计教学网站,以实现教学互动、网上作业和网上考试。

本书以程序设计为主线,突出C/C++语言实用的重点概念,由案例、知识点和实验组成,加强程序阅读、编写和调试能力,强化实践训练。同时遵守循序渐进的原则,讲解深入浅出,并通过大量的案例讲解C/C++程序设计方法。本书的主要内容包括:基本数据类型、运算符和表达式、程序控制结构、数组和字符串处理、函数和模块化程序设计、指针、结构体和共用体、位运算、文件、C++基础知识、面向对象程序设计及应用程序实例等。其内容丰富、概念清晰、组织精练、实用性强,并将知识点融入各章典型例题中,同时也注重培养学习者计算机程序设计能力和良好的程序设计风格。书中精选了典型的案例和习题,帮助学习者开阔思路和提高程序设计能力。书中的完整程序代码均在Microsoft Visual C++ 6.0环境下调试通过。

本书由吴国凤、宣善立任主编,姚合生、黄晓梅任副主编,各章主要编写人员分工如下:第1、9章由吴国凤编写,第2章由偶春生编写,第3、12章由宣善立编写,第4、8章由黄晓梅编写,第5章由王金玲编写,第6、7章由姚合生编写,第10、11章由红光编写,全书由吴国凤统编定稿。参加本书编写的还有李明、王卫、林杰华、方乃义、娄彦山、冷金麟、黎杰等。

在本书的编写过程中得到了兄弟院校从事计算机教育的专家、老师的支持和帮助以及教研室各位同仁的协作,在此一并表示最真诚的谢意。

由于编者水平有限,书中难免有疏漏之处,恳请各位专家和读者批评指正。

编 者

2009年4月

第1版前言

本书是根据教育部最新提出的《关于进一步加强高等学校计算机基础教学的意见》中有关“计算机程序设计基础”课程的教学要求而编写的。

为了培养高素质人才，各高等院校均已将计算机基础教育放到了很重要的位置，计算机基础教育不仅包括大学计算机基础，还包括计算机程序设计基础。通过程序设计语言的学习，学习者可以掌握程序设计的方法与技术，培养程序设计的能力。为了满足程序设计基础教学改革的需要，我们组织了有关高校长期在教学第一线工作的教师编写了《C/C++程序设计》一书，供广大高校的师生们选用。为适应教学方式的变革，除了编写主教材之外，还编写了配套的辅助教材、课件、电子教案及相应的程序设计素材。

本书共12章，主要内容包括：C语言概述、基本程序设计、程序控制结构、数组、函数、指针、结构体与共用体、位运算、文件、C++基础知识、面向对象程序设计及应用程序实例等。本书全面、系统地讲述了C/C++语言的基本概念、编写程序的方法以及程序设计的技巧。本书内容丰富，概念清晰，组织精练，系统性强，同时突出了学生程序设计能力的培养。另外，每章还配备了大量的习题，便于学生练习掌握主要知识点。

本书由吴国凤、宣善立任主编，姚合生、黄晓梅任副主编。各章编写分工如下：第1、9章由吴国凤编写，第2章由谢文佩编写，第3、12章由宣善立编写，第4、8章由黄晓梅编写，第5章由王金玲编写，第6、7章由姚合生编写，第10、11章由红光编写，全书由吴国凤、宣善立统编定稿。参加本书编写的还有李明、偶春生、王卫、林杰华、姜彦山、方乃义、冷金麟、黎杰等。

在本书的编写过程中得到了合肥工业大学计算机与信息学院胡学钢、王浩教授和高等教育出版社领导和相关编辑的大力支持以及许多同行的支持和帮助，在此一并表示最真诚的谢意。

由于编者水平有限，书中难免有疏忽、错误之处，恳请专家和读者批评指正。

编 者

2006.6

目 录

第1章 C语言概述	1	2.2.2 实型常量	26
1.1 程序与程序设计	1	2.2.3 字符常量	27
1.1.1 程序的概念	1	2.2.4 字符串常量	27
1.1.2 程序设计语言	2	2.2.5 符号常量	28
1.1.3 程序设计方法	3	2.3 变量及其类型	28
1.2 C/C++语言简介	5	2.3.1 变量及其说明	28
1.2.1 C/C++语言的由来	5	2.3.2 整型变量	29
1.2.2 C语言的特点	6	2.3.3 实型变量	32
1.3 C程序初识	8	2.3.4 字符变量	33
1.3.1 C程序实例	8	2.4 运算符和表达式	34
1.3.2 C程序基本构成	10	2.4.1 算术运算符和算术表达式	34
1.3.3 程序编写规范与风格	12	2.4.2 关系运算符和关系表达式	36
1.4 C语言的语法	12	2.4.3 逻辑运算符和逻辑表达式	37
1.4.1 字符集	12	2.4.4 赋值运算符和赋值表达式	39
1.4.2 关键字	13	2.4.5 其他运算符及表达式	40
1.4.3 标识符	14	2.4.6 运算的优先级与结合性	42
1.4.4 常量	14	2.4.7 数据类型转换	42
1.4.5 运算符	14	2.5 数据的输入/输出	44
1.4.6 分隔符	14	2.5.1 数据输入/输出概念	44
1.5 算法	15	2.5.2 字符数据的输入/输出	44
1.5.1 算法的基本特征	15	2.5.3 格式输入/输出	45
1.5.2 算法的表示	15	2.6 典型例题精解	51
1.6 C程序的编程环境与学习方法	17	本章小结	52
1.6.1 C程序的实现过程	17	习题2	52
1.6.2 VC++ 6.0编程环境	18	第3章 程序控制结构	57
1.6.3 学习C语言的方法	20	3.1 顺序结构	57
1.7 典型例题精解	21	3.1.1 C语言的语句	57
本章小结	22	3.1.2 顺序结构程序设计	59
习题1	22	3.2 分支结构	60
第2章 基本程序设计	24	3.2.1 if (if-else) 语句	60
2.1 C语言数据类型	24	3.2.2 switch 语句	65
2.2 常量及其类型	25	3.3 循环结构	67
2.2.1 整型常量	25	3.3.1 while 语句	68

II 目录

3.3.2 do-while 语句	68	5.3.1 形式参数和实际参数	127
3.3.3 for 语句	69	5.3.2 变量作为函数参数	128
3.3.4 转移语句	71	5.3.3 数组作为函数参数	129
3.3.5 循环的嵌套	73	5.4 函数的嵌套调用与递归调用	133
3.4 典型例题精解	75	5.4.1 函数的嵌套调用	133
本章小结	82	5.4.2 函数的递归调用	134
习题 3	82	5.5 变量的作用域和存储类型	136
第 4 章 数组	89	5.5.1 变量的生存期与作用域	136
4.1 数组的概念	89	5.5.2 变量的存储类型	138
4.2 一维数组	90	5.5.3 内部函数与外部函数	144
4.2.1 一维数组的定义	90	5.6 编译预处理	145
4.2.2 一维数组元素的引用	90	5.6.1 宏定义	145
4.2.3 一维数组的初始化	91	5.6.2 文件包含	148
4.2.4 一维数组的应用	93	5.6.3 条件编译	150
4.3 多维数组	95	5.7 典型例题精解	151
4.3.1 多维数组的概念	95	本章小结	157
4.3.2 二维数组的说明及引用	96	习题 5	157
4.3.3 二维数组元素的存储顺序	97	第 6 章 指针	164
4.3.4 二维数组的初始化	97	6.1 指针的基本概念	164
4.3.5 二维数组的应用	99	6.1.1 内存、地址和指针	164
4.4 字符数组和字符串	101	6.1.2 指针变量的定义与初始化	165
4.4.1 字符数组的定义	102	6.1.3 指针的运算及引用	166
4.4.2 字符数组的初始化	102	6.1.4 指针变量作为函数参数	170
4.4.3 字符串的输入/输出	103	6.1.5 多级指针的概念	171
4.4.4 字符串处理函数	104	6.2 指针与数组	173
4.5 典型例题精解	109	6.2.1 指针与一维数组	173
本章小结	113	6.2.2 指针与二维数组	177
习题 4	113	6.2.3 指针与字符串	180
第 5 章 函数	119	6.2.4 指针数组	184
5.1 概述	119	6.3 指针与函数	188
5.1.1 函数的概念	119	6.3.1 指向函数的指针	188
5.1.2 函数的分类	120	6.3.2 返回指针的函数	190
5.2 函数的定义和调用	121	6.3.3 带参数的主函数	191
5.2.1 函数的定义	121	6.4 典型例题精解	192
5.2.2 函数的调用	122	本章小结	196
5.2.3 函数的原型声明	124	习题 6	198
5.2.4 标准库函数	126	第 7 章 结构体与共用体	205
5.3 函数的参数及传递方式	127	7.1 结构体	205

7.1.1 结构体类型的定义	205	9.3.1 文件的打开	272
7.1.2 结构体变量的说明	206	9.3.2 文件的关闭	274
7.1.3 结构体变量的初始化	209	9.4 文件的读/写	275
7.1.4 结构体变量的引用	209	9.4.1 字符输入/输出函数	275
7.1.5 结构体数组	211	9.4.2 文件的字符串输入/输出函数	277
7.1.6 结构体指针	214	9.4.3 文件的格式化输入/输出函数	279
7.1.7 结构体与函数	216	9.4.4 文件的数据块输入/输出函数	281
7.1.8 动态内存分配与链表	220	9.4.5 整数输入/输出函数	283
7.2 共用体	229	9.5 文件的定位操作	284
7.2.1 共用体类型的定义	229	9.6 文件的错误检测	287
7.2.2 共用体变量的说明	229	9.7 典型例题精解	289
7.2.3 共用体变量的引用	231	本章小结	290
7.3 枚举类型	234	习题 9	290
7.3.1 枚举类型的定义	234	第 10 章 C++基础知识	295
7.3.2 枚举变量的说明及引用	234	10.1 面向对象程序设计概述	295
7.3.3 枚举类型的应用	236	10.1.1 面向对象的基本概念	295
7.4 用户定义类型	236	10.1.2 面向对象程序设计的特点	296
7.5 典型例题精解	238	10.2 C++程序结构	297
本章小结	243	10.2.1 一个简单的 C++程序	297
习题 7	243	10.2.2 C 程序与 C++程序的区别	298
第 8 章 位运算	252	10.3 类和对象	304
8.1 位运算的基本概念	252	10.3.1 类和对象的定义	304
8.2 计算机内的数据表示	252	10.3.2 构造函数与析构函数	307
8.3 位运算	253	10.3.3 类中的 const 修饰符	309
8.3.1 逻辑位运算	253	10.3.4 静态成员	311
8.3.2 移位运算	257	10.3.5 友元	312
8.4 位域	260	10.4 典型例题精解	313
8.4.1 位域的定义及位域变量的说明	260	本章小结	317
8.4.2 位域变量的使用	261	习题 10	317
8.5 典型例题精解	263	第 11 章 面向对象程序设计	321
本章小结	265	11.1 继承与派生	321
习题 8	265	11.1.1 基类和派生类	321
第 9 章 文件	268	11.1.2 单一继承	322
9.1 概述	268	11.1.3 多重继承	323
9.1.1 文件的基本概念	268	11.2 多态性与虚函数	323
9.1.2 文件的分类	269	11.2.1 函数重载	323
9.2 文件类型指针	270	11.2.2 运算符重载	325
9.3 文件的打开与关闭	272	11.2.3 虚函数	327

IV 目录

11.2.4 抽象类.....	330	12.1.5 编码.....	342
11.3 典型例题精解.....	330	12.1.6 学生成绩管理系统源码.....	343
本章小结.....	334	12.2 通讯录管理系统.....	351
习题 11.....	334	12.2.1 需求分析.....	351
第 12 章 应用程序实例.....	338	12.2.2 系统设计.....	352
12.1 学生成绩管理系统设计.....	338	12.2.3 通讯录系统源代码.....	354
12.1.1 问题定义.....	338	附录 1 常用字符与 ASCII 代码对照表.....	370
12.1.2 设计方法.....	339	附录 2 运算优先级和结合性总表.....	371
12.1.3 系统模块分解.....	340	参考文献.....	372
12.1.4 模块设计.....	340		

【内容概述】

本章从认识程序入手，主要介绍程序设计的方法与结构化程序设计的特征以及 C/C++ 语言的发展、特点；并以简单案例出发，分析 C 语言程序的基本结构及其组成，归纳了程序、函数、语句的特点与规则及 C 程序中的基本词汇，说明了算法的几种方法及在 VC++ 6.0 的环境下编程的基本步骤。

【教学目标】

1. 熟悉程序、程序设计语言和方法；
 2. 掌握结构化程序设计的 3 种基本控制结构；
 3. 了解 C/C++ 语言的特点；
 4. 掌握 C 程序的基本结构及组成；
 5. 掌握 C 程序中的基本词汇；
 6. 熟悉程序设计的算法概念；
 7. 掌握在 VC++ 6.0 环境中编辑、编译、连接和运行 C 程序的基本步骤和方法。
-

1.1 程序与程序设计

1.1.1 程序的概念

1. 程序

计算机最吸引人的地方在于它能自动执行指定的程序。所谓程序，就是用语言、文字、图表等方式表达解决某个问题的方法和步骤，是计算机解决某些特定问题所需的代码化指令序列。程序设计者根据预先制定的功能和规则，编写一系列完整指令，由计算机执行，实现预定的功能和任务，这就是计算机程序。

2. 程序的描述

一个程序应包括两方面的内容：对数据的描述和对操作的描述。

- ① 对数据的描述：在程序中指定数据的类型和数据的组织形式，即数据结构。数据结构是

2 第1章 C语言概述

数据与数据间存在的一种或多种特定关系。在程序设计语言中，与数据结构密切相关的便是数据的类型和数据的存储。在C语言中，系统提供的数据结构是以数据类型形式出现的。

② 对操作的描述：操作步骤，也就是算法。算法是为了解决一个问题而采取的方法和步骤，是程序的灵魂。为此，著名计算机科学家沃思（Nikiklaus Wirth）提出了一个公式：

$$\text{程序} = \text{数据结构} + \text{算法}$$

由此可知，算法和数据结构是一个整体。要编写一个程序，首先要掌握一种程序设计语言并熟悉其开发环境，同时也要了解所研究的问题并对问题进行求解的方法（即程序设计方法）。其中，算法是灵魂，数据结构是加工对象，语言是工具，编程需要采取合适的方法。

1.1.2 程序设计语言

程序设计语言是用户用来编写程序以操作计算机的语言系统。程序设计语言的发展从低级语言到高级语言经历以下几个发展过程。

1. 机器语言

机器语言是用二进制指令代码表示的指令集合，是计算机能直接识别和执行的语言。用机器语言编写的程序运行效率高，占用内存少。但由于面向机器，因机而异，通用性差，而且程序不直观，程序的编写和维护都很困难。

2. 汇编语言

为了便于记忆，人们引进一些符号来表示这些指令。例如，ADD（Addition）表示加法，即用助记符号代替了以二进制表示的机器指令。人们把这种改进的、用符号来描述的指令系统称为汇编语言。

用汇编语言编写的程序称为源程序，计算机不能直接识别和执行，必须翻译成机器语言的目标程序后才能执行。汇编语言翻译软件称为汇编程序，它可以将程序员编写的助记符直接转换为机器指令，然后再由计算机去识别和执行。

3. 面向过程语言

高级语言是采用命令和语句的语言。高级语言表达方式接近被描述的问题，接近于自然语言和数学表达式，它屏蔽了机器的细节，提高了语言的抽象层次，易于被人们接受和掌握。随着计算机技术的发展，形成了为数众多的高级语言，目前应用较广的有FORTRAN、BASIC、Pascal、C语言等。

4. 面向对象语言

面向对象语言是从面向过程的语言发展起来的，用这种语言编程，人们只需告诉计算机“做什么”，而无须告诉计算机“怎样做”。面向对象分析与设计方法提供了抽象类型机制，将数据与数据的操作封装在一起，共同完成信息和处理的双重模块化，它的封装和继承很好地体现了现代软件工程的特点。当前使用较多的面向对象语言有C++、C#、Java、Visual Basic、Object Pascal等。

5. 程序执行方式

由于计算机只能识别机器语言，因此，用高级语言编写的程序（或称为源程序）必须经过翻译生成可执行文件才能执行。翻译方式有两种：解释方式和编译方式。

解释方式是通过运行“解释器”来一条一条地解释语句并执行，不生成目标代码，速度较慢，如 BASIC、ASP、JavaScript、VBScript 等语言。

编译方式是用编译语言编写源程序，编译、连接生成目标代码后，再运行目标代码得到结果，这种运行方式效率较高，如 C、C++、C#、Java 等，其编译过程如图 1-1 所示。



图 1-1 编译语言的执行过程

重点

- ◀ 由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。
- ◀ 用助记符代替机器指令的符号语言称为汇编语言。
- ◀ 高级语言所编制的程序不能被计算机识别，必须经过翻译生成可执行文件才能执行。

1.1.3 程序设计方法

用计算机解决一个实际应用问题时的整个处理过程称为程序设计。程序设计需要一定的方法来指导，以便提高程序设计的可读性、可维护性、稳定性及编程效率。目前有两种重要的程序设计方法：结构化程序设计方法和面向对象程序设计方法。

1. 结构化程序设计方法

结构化程序设计(Structured Programming)是荷兰学者于 1969 年提出的一种程序设计方法，这种方法建立在经典的结构定理基础之上。结构定理指出：任何程序逻辑都可以用顺序、选择和循环 3 种基本结构来表示，如图 1-2 所示。实践证明：结构化程序设计方法确实使程序执行效率提高，并且由于减少了程序出错率，从而大大减少了维护的费用。

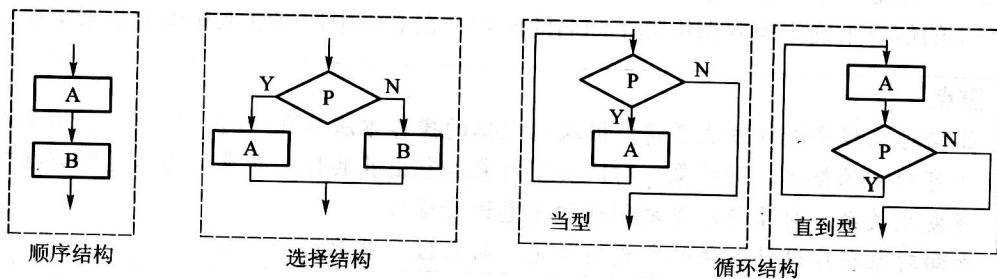


图 1-2 3 种基本控制结构

结构化程序设计有两个主要特征。

① 自顶向下、逐步求精和模块化设计是结构化程序设计方法中最典型、最具有代表性的方法。将大型任务从上向下划分为多个功能模块，每个模块又可以划分为若干子模块，然后分别进行模块程序的编写。

② 程序总是由3种基本结构组成：顺序结构、选择结构和循环结构。这3种结构的共同特点是单入口、单出口、无死语句、无死循环。

结构化程序设计是面向过程的程序设计方法，其主要思想是把复杂问题分解成若干简单的子问题并逐步求精。一个结构化程序就是用高级语言表示的结构化算法，这种程序具有易读、易理解、易于修改和维护等优点，同时也提高了程序的可靠性，保证了程序的质量。

重点

- ◀ 结构化程序设计的思路是：自顶向下、逐步求精和模块化设计。
 - ◀ 每一模块内部均由顺序、选择和循环3种基本结构组成。
-

2. 面向对象程序设计方法

面向对象的程序设计是另一种重要的程序设计方法，它能有效地改进结构化程序设计方法的不足。面向对象程序设计方法和结构化程序设计方法的不同点是：结构化程序设计方法采用函数（或过程）来描述对数据的操作，但又将函数与其操作的数据分离开来；而面向对象程序设计方法将数据和对数据的操作封装在一起，作为一个整体来处理。

面向对象程序设计语言的3个主要特征是：封装性、继承性和多态性。

① 封装性：也叫数据隐藏，是面向对象方法的一个重要原则。封装是把对象的属性和操作结合在一起，构成一个独立的对象。用户无须知道其内部工作流程，只需要知道接口和操作即可，一般用类来实现封装。

② 继承性：继承是指子类可以拥有父类的属性和行为。这是一种支持重用的思想，用现有的类型派生出新的子类。例如，新型轿车在原有型号的基础上增加若干种功能，新型轿车类是从原有轿车类派生出来的，它继承了原有轿车类的属性，并增加了新的功能。

③ 多态性：是指当多种事物继承一种事物时，所表现出的不同行为。多态性通常分为编译时多态和运行时多态。编译时多态是指在程序编译阶段由编译系统根据参数确定与哪个同名的函数相联系，在C++中通过函数重载和运算符重载来实现。运行时多态是指在程序的运行阶段才根据产生的信息来确定需要调用哪个同名的函数，在C++中通过继承和虚函数来实现。

重点

- ◀ 面向对象程序设计方法是将数据及对数据的操作方法放在一起，作为一个相互依存、不可分割的整体，即对象。对同类型对象抽象出其共性，即类。类通过外部接口与外界发生关系，对象与对象之间通过消息进行通信。
 - ◀ 面向对象程序设计方法的主要特征是：封装性、继承性和多态性。
-

1.2 C/C++语言简介

1.2.1 C/C++语言的由来

1. C 语言的由来

C 语言是目前世界上最流行和实用的一种计算机高级程序设计语言。它适合于作为系统描述语言,既可以用来编写系统软件,也可以用来编写应用软件。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的,后随其宿主系统 UNIX 的普及而引起注意,又因其本身的高效、灵活、功能丰富、表达力强、移植性好等特殊优点,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 语言的原型是 ALGOL 60 语言。1963 年,剑桥大学将 ALGOL 60 语言发展成为 CPL (Combined Programming Language, 复合程序设计语言)。CPL 语言更接近硬件层,但规模较大,难以实现。1967 年剑桥大学的马丁·理查德对 CPL 进行了简化,推出了 BCPL (基本复合程序设计语言)。1970 年美国贝尔实验室的肯·汤普逊对 BCPL 进行了进一步简化,突出了硬件处理能力,并取“BCPL”的第一个字母“B”作为新语言的名称,且用 B 语言编写了第一个 UNIX 操作系统。但 B 语言过于简单,功能有限,在 1972 年贝尔实验室的布朗·W.卡尼汉和丹尼斯·M.利奇对 B 语言进行了完善和扩充,在 B 语言的基础上最终设计出一种新的语言,他取“BCPL”的第二个字母“C”作为这种语言的名字,这就是 C 语言。C 语言保持了 BCPL 语言和 B 语言精练、接近硬件的优点,又克服了它们过于简单、数据无类型、功能有限的缺点。后经多次改进,在贝尔实验室内部使用。1978 年以后, C 语言被移植到各种机型上。由于它作为系统描述语言既可以用来编写系统软件,又可以用来编写应用软件,所以很快受到欢迎。许多程序员纷纷转为使用 C 语言编程,从而使 C 语言在当时的软件开发中几乎一统天下。

1977 年,为了让 C 语言脱离 UNIX 操作系统,成为在任何计算机上都能运行的通用计算机语言,布朗·W.卡尼汉和丹尼斯·M.利奇(K&R)合著了著名的《C 程序设计语言》一书,对 C 语言的语法进行了规范化的描述,成为当时的标准,该书也被简称为《K&R》标准。但是,在《K&R》中并没有给出一个完整的标准 C 语言定义,后来由 ANSI (American National Standards Institute, 美国国家标准协会)在此基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C。

C 语言广泛应用于专业程序设计,程序员无须知道硬件的细节就可以编制应用程序。同时, C 语言也有一些与硬件直接相关的“低级”语句,使程序员可以编制直接访问硬件的程序。目前使用的 C 语言编译系统有 Microsoft C、Turbo C、Quick C、Borland C 等版本,各种版本略有差异,因此,在使用一个系统之前,要了解所用的计算机系统配置的 C 编译系统的特点和规定。

2. C++语言的由来

C++是从 C 语言发展演变而来的。1983 年,美国 AT&T 公司贝尔实验室的 Bjarne Stroustrup 博士在 C 语言的基础上,扩充了 Simula 67 语言中面向对象的概念,发明了 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++语言的标准化工作从1989年开始，于1994年制定了ANSI C++标准草案。目前流行的最新版本是C++ Borland 6.0和VC++ 6.0。

C++提出了一些更为深入的概念，它所采用的面向对象程序设计方法，很好地解决了日益复杂的程序问题，它所支持的面向对象的概念也能方便地将问题空间直接映射到程序空间，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。但这也增加了语言的复杂性，掌握起来有一定难度。不过，C是C++的基础，C++语言和C语言在很多方面是兼容的。因此，掌握了C语言，再进一步学习C++就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。

1.2.2 C语言的特点

C语言同时具有汇编语言和高级语言的优势，它可以对内存地址进行读/写，可直接对硬件、二进制位进行操作。它把高级语言面向过程和低级语言与硬件关系密切的优点有机地结合起来，广泛应用于系统软件、应用软件、数据处理以及科学计算等领域。C语言之所以成为目前世界上使用较广泛的程序设计语言，并被选作C++的基础语言，是由C语言的诸多突出优点所决定的。

1. 语言简洁紧凑，使用方便

ANSI C只有32个关键字和9种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分（与Pascal语言的比较），如表1-1所示。

表 1-1 C语言与 Pascal 语言比较

C 语 言	Pascal 语言	含 义
{ }	BEGIN...END	复合语句
if(e) s;	IF(e) THEN s	条件语句
int i;	VAR i:INTEGER	定义 i 为整型变量
int a[10];	VAR a:ARRAY[1..10] OF INTEGER	定义整型数组 a
int f();	FUNCTION f():INTEGER	定义 f 为返回值整型的函数
int*p;	VAR p: ↑ INTEGER	定义 p 为指向整型变量的指针
i+=2;	i=i+2	赋值 i+2⇒i
i++,++i;	i=i+1	i 自增值

结论：C语言比Pascal语言简练，源程序短。

2. 运算符极其丰富

C语言共有44种运算符（见附录2）。C语言把括号、赋值、强制类型转换等都作为运算符处理，从而使得C语言的运算符极其丰富，表达式类型多样化。灵活使用各种运算符可以实现其他高级语言中难以实现的运算。例如：

算术运算符+、-、*、/、%、（加、减、乘、除、求余），++、--（自增、自减）；逻辑运算符!、&&、!!（逻辑非、逻辑与、逻辑或）；位运算符<<=、>>=（左移、右移）等。

3. 数据结构类型丰富

C语言具有整型、实型、字符型、数组、指针、结构体、共同体等数据类型。能方便地构造更加复杂的数据结构（如使用指针构造链表、树、栈）。为处理各种复杂数据类型提供了实用的手段。

4. 模块化结构

C语言是一种模块化的程序设计语言。C程序是由一系列函数构成的，这些函数可以自己设计，也可以由标准函数提供。C语言程序是由最顶层的函数 `main()` 按照一定的组织层次调用标准函数或自己设计的函数而形成的。

5. 具有结构化的控制语句

C语言具有多种结构化的控制语句，提供了 `if-else` 语句、`while` 语句、`do - while` 语句、`switch` 语句、`for` 语句。可以很容易地实现结构化的各种基本结构，用来设计结构化程序。

6. 允许直接访问物理地址，进行位操作

能实现汇编语言的大部分功能，可以对硬件进行操作。

① C语言可以直接操作计算机硬件，如寄存器、各种外设 I/O 端口等。

② C语言的指针可以直接访问内存物理地址。

③ C语言类似汇编语言的位操作可以方便地检查系统硬件的状态。

因此，C语言既具有高级语言的功能，又具有低级语言的许多功能，可以用来编写系统软件。C语言的这种双重性，使得其既可以作为系统描述语言，又可以作为程序设计语言。

7. C语言对语法限制不严格，程序设计灵活

C语言不检查数组下标越界，不限制对各种数据的转换（编译系统可能对不合适的转化进行警告，但不限制），不限制指针的使用，程序正确性由程序员保证。

实践中，C语言程序编译时会提示“警告错误”或“严重错误”。“警告错误”表示用户使用的语法可能有问题，但是有时可以忽略，程序仍然可以完成编译工作，然后运行（但是一般情况下“警告错误”往往意味着程序真的有问题，应该认真地检查）。“严重错误”是不能忽略的，编译系统发现严重错误后，不会产生目标代码。

灵活和安全是一对矛盾，对语法限制的不严格也是C语言的一个缺点，如黑客使用越界的数组攻击其他用户的计算机系统。

8. 生成的目标代码质量高，程序执行效率高

C语言的有些操作是直接对应于实际计算机硬件的，在很多方面具有汇编语言的特性，因此能充分反映计算机硬件的功能，所生成的代码效率极高。有资料显示C语言只比汇编代码效率低10%~20%。

9. 可移植性好

由于C语言的标准化以及C程序的输入/输出、内存管理等操作均采用C库函数实现,这不仅使得C编译程序很容易在不同的计算机上实现,而且使得用户编写的C程序可以不做修改或作少量修改就能在不同的计算机上运行。

重点

- ◀ C语言具有强大的功能,既可编写系统软件,也可编写应用软件。熟练掌握C语言是计算机开发人员的一项基本功。

1.3 C程序初识

C语言的程序是由一个或多个函数构成的。为了让读者对编程语言有初步认识,下面通过几个简单的C程序案例,分析C语言程序的基本结构及组成、特点和相关的规则。

1.3.1 C程序实例

【例 1.1】输出字符串。

```
#include <stdio.h>

void main(void)
{
    printf("welcome!\n");
}
```

程序运行结果如图 1-3 所示。

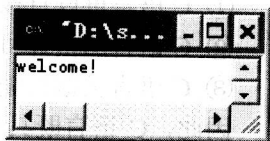


图 1-3 例 1.1 程序运行结果

程序点拨

这是一个最简单的C语言程序,作用是将字符串“welcome!”显示在屏幕上。

① 第一行,include是预处理命令,其功能是把指定的头文件(.h)包含到本程序中。C语言中没有输入/输出语句,程序中的输入/输出是通过C语言本身提供的标准库函数来实现的,它的原型放在<stdio.h>头文件中。程序中的printf()函数在stdio.h中声明。

② 第二行,main()是函数名,称为主函数。在任何一个C程序中必须有且只有一个main()函数,任何程序总是从主函数开始执行,最终在主函数中结束。

③ void是系统定义的关键字,表示空类型。函数名左边的void代表本函数无返回值,函数名右边的void代表本程序的主函数无参数。

④ 第三行和第五行{ }中的内容是函数体。左括号{表示函数体开始,右括号}表示函数体结束。函数体中可以包含一个或多个语句,每个语句末尾一定要加上分号“;”,表示语句结束。

⑤ 第四行,printf()是C语言提供的标准输出函数,其功能是在屏幕上将双引号之间的内容原样输出。“\n”是转义符,它的作用是将光标移到下一行的开始处。

【例 1.2】求a的平方根。

```
#include <stdio.h>
```