



QUANGUOGAODENGZHIYE
JIAOYUJIAOCAICONGSHU

全国高等职业教育教材丛书

C 语言 程序设计

高福成 编著 边莫英 主审

CYUYAN
CHENGXUSHEJI

南开大学出版社

全国高等职业教育教材丛书

C 语言程序设计

高福成 编著

边奠英 主审

南开大学出版社

天 津

图书在版编目(CIP)数据

C 语言程序设计 / 高福成编著. —天津: 南开大学出版社, 2001. 11

(全国高等职业教育教材丛书)

ISBN 7-310-01576-2

I. C... II. 高... III. C 语言—程序设计—高等教育: 技术教育—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 040718 号

出版发行 南开大学出版社

地址: 天津市南开区卫津路 94 号

邮编: 300071 电话: (022) 23508542

出版人 肖占鹏

承印 天津宝坻第二印刷厂印刷

经销 全国各地新华书店

版次 2001 年 11 月第 1 版

印次 2001 年 11 月第 1 次印刷

开本 787mm×1092mm 1/16

印张 16.75

字数 416 千字

印数 1—5000

定价 23.00 元

内容提要

本书针对高等职业技术教育和高等教育自学考试的特点，以应用为目的，以程序设计为主线，系统介绍了 C 语言及其程序设计技术。全书共 11 章，包括 C 语言的基本数据类型、数据运算、程序控制结构、数组、函数、指针、复合数据类型、文件和编译预处理等，最后简要介绍了 C++ 和面向对象程序设计。书中突出了重点，分解了难点，配以大量的应用实例和多种类型的习题，方便自学。本书既是新高职计算机专业教材，也可供参加全国计算机等级考试二级 C 语言的考生参考。

高等职业教育教材编审委员会名单

主任委员：

乔丽娟

委员：（以姓氏笔画为序）

丁桂芝 王松岭 边奠英 刘凤桐

李占伦 李维祥 吴功宜 赵雅兴

徐宝强 徐娟敏 葛洪贵

序

全国高等教育自学考试指导委员会副主任

中国职业技术教育学会副会长 王明达

中国高等教育大众化目标的实现必然伴随着高等教育形式和结构多样化的变革。单纯以学术水平为追求目标的高等教育无法满足社会对于多种专门人才的需求,因此要大力发展高等职业教育,培养社会需要的各类专门人才,以适应我国经济和社会发展的要求。

什么是高等职业教育?职业教育的特征不在于办学形式,主要体现在培养目标上。培养生产、服务、管理第一线的实用型人才的教育即为职业教育。按照专业所需接受教育的年限达到相当于普通高等教育学习年限的职业教育即为高等职业教育。

高等职业教育如何实现培养实用人才的目标?首要的就是专业设置。既然培养的是生产第一线的实用型人才,所设专业就一定是直接与社会生产、生活相联系的,社会生产、生活中最必需的。这与普通高等学校开设专业的思路有着本质的区别。其次是教学内容的安排和教学计划的制定。接受高等职业教育的学生其学习内容必须是成熟的技术和管理规范,教学计划、课程设置应该按照职业岗位群的职业能力要求来确定,而不应从学科体系出发。再次,为使学生毕业就能基本顶岗工作,要求增大实习训练所占的比例,在校期间就基本完成上岗前的实践训练。为了保证实践训练得到社会认可,要实行学历证书与职业资格证书“双证书”制度,同时要求双师型教师任教。只有按部就班实现以上要求的高等职业教育才会被社会认同,也才会有生命力。

办出特色是高等职业教育生命力的源泉。学生毕业即能顶岗是职业教育区别于其他教育的一个突出特点。要想做到这一点,一方面学习理论知识要以“必需”和“够用”为度,让学生掌握基本理论和知识;另一方面要全方位开辟实习基地,保证充足的实训时间。高等职业教育的水准主要是通过专业设置、课程内容,以及实训能力的培养体现的。

为落实第三次全教会“完善自学考试制度”、“大力发展高等职业教育”的改革思路,1999年全国高等教育自学考试指导委员会决定在天津市开展高等教育自学考试职业技术专业的试验工作。

天津市高等教育自学考试委员会在深入调查研究的基础上,从职业岗位群的知识技能需求出发,以能力本位教育(CBE)为理论依托,设计了12个职业技术专业,于2000年面向社会开考。

高等教育自学考试开考职业技术专业的试验,在完善高等教育自学考试专业建设、拓

展自学考试教育功能方面，在探索开放式教育、培养应用型高级人才方面，在职业教育课程体系方面，以及在实践技能考核的研究、管理方面，对于我国高等教育自学考试制度的完善和高等职业教育的发展都具有重要意义。

天津市高等教育自学考试委员会将根据职业技术专业试验工作的需要陆续出版有关考试课程的教材。教材编撰者多为具有职业教育经验的学科专家和职业教育专家，他们根据职业教育的专业培养目标重新整合了学科知识体系，尽力体现理论知识必需、够用的原则。当然，由于认识水平的局限和时间的紧迫，这些教材还需要继续完善提高。尽管如此，这迈出的第一步是十分可贵的。我深信，高等教育自学考试职业技术专业的试验工作一定能取得成功。

2001年1月于北京

前 言

C语言是计算机程序设计语言中的小字辈，但由于其卓越的性能，使它风靡全世界，成为最受欢迎的语言之一。当今流行的面向对象语言 C++及 Internet 上的 Java 语言就来源于 C 语言。

学习计算机语言的惟一目的是应用，而应用要通过程序设计来体现。进行程序设计，需要很强的逻辑思维能力，是一种极富创造性的智力劳动。可以这样认为，语言是一种技能，程序设计是一门科学。因此，任何计算机语言及其程序设计的基本特点就是理论性和实践性并重，教学上应强调科学训练与技能培养并存。基于这一认识，本书的宗旨是以应用为目的、以程序设计为主线，系统介绍 C 语言及其程序设计技术，让读者在大量的程序设计实践中自然而然地熟悉和掌握语言。

本书是根据天津市高等教育自学考试委员会制定的《C 语言程序设计大纲》的要求编写的，全书共 11 章。第 1 章介绍简单的 C 程序设计；第 2、5、8 章从易到难介绍 C 语言的各种数据结构；第 3 章专门介绍 C 语言的运算功能；第 4 章集中介绍 C 语言的控制结构，体现结构化程序设计的特点；第 7、9 章介绍函数和文件，这是模块化程序设计的需要；第 6 章介绍指针，这是 C 语言的精华和特色，也是学习的难点；第 10 章介绍 C 语言环境中的特殊问题，即编译预处理和分割编译；第 11 章简单介绍 C++，以便于读者进一步学习和应用 C++ 语言进行面向对象的程序设计。附录给出了习题(选择题、填空题和程序改错题)参考答案。

在本书的编写和出版过程中，得到天津市高等教育自学考试委员会的大力支持，南开大学出版社的领导和编辑付出了大量的辛勤劳动，在此表示由衷的感谢。

编者

2001 年 5 月

目 录

第1章 C语言概述..... (1)	
1.1 C语言的起源和发展..... (1)	
1.2 C语言的特点..... (2)	
1.3 C语言程序的基本结构..... (4)	
1.3.1 一个简单的C语言程序..... (4)	
1.3.2 C语言程序的基本结构 (4)	
1.4 简单的C程序设计..... (8)	
1.4.1 格式输出函数 printf()	
简介..... (9)	
1.4.2 格式输入函数 scanf()	
简介..... (10)	
1.4.3 简单的赋值语句..... (11)	
1.4.4 简单程序设计举例..... (11)	
1.5 C程序的开发过程和Turbo C集成	
环境..... (12)	
1.5.1 C程序的开发过程..... (12)	
1.5.2 Turbo C 2.0 集成环境..... (13)	
本章小结..... (18)	
练习题..... (19)	
第2章 基本数据类型、常量和变量..... (21)	
2.1 基本数据类型..... (21)	
2.1.1 五种基本数据类型..... (21)	
2.1.2 基本数据类型的存储方式	
和取值范围 (21)	
2.1.3 基本数据类型的修饰..... (23)	
2.2 常量及其类型..... (23)	
2.2.1 整数 (23)	
2.2.2 实数 (24)	
2.2.3 字符常量..... (25)	
2.2.4 字符串常量..... (26)	
2.3 变量及其类型..... (27)	
2.3.1 变量的定义..... (27)	
2.3.2 变量的初始化..... (31)	
2.4 符号常量 (32)	
2.4.1 宏定义 (32)	
2.4.2 const 修饰..... (33)	
2.5 不同类型数据的输入输出..... (34)	
2.5.1 printf()函数..... (34)	
2.5.2 scanf()函数..... (36)	
2.5.3 字符输出函数 putchar()	
和输入函数 getchar() (37)	
2.5.4 单字符输入函数	
getche()和 getch() (38)	
本章小结..... (39)	
练习题..... (39)	
第3章 数据运算..... (44)	
3.1 算术运算 (45)	
3.1.1 算术运算符..... (45)	
3.1.2 算术表达式..... (45)	
3.1.3 强制类型转换..... (48)	
3.2 赋值运算 (49)	
3.3 逗号运算(顺序运算) (51)	
3.4 关系运算和逻辑运算..... (53)	
3.4.1 关系运算符和逻辑	
运算符 (53)	
3.4.2 关系表达式、相等表达式和	
逻辑表达式 (54)	
3.5 位运算 (56)	
3.5.1 按位逻辑运算..... (56)	
3.5.2 移位运算 (57)	
3.5.3 由位操作运算符组成	
的复合赋值运算符..... (58)	
3.5.4 位操作的应用举例..... (59)	
3.6 测试数据长度运算符 sizeof..... (60)	
3.7 数学函数 (61)	
本章小结..... (62)	
练习题..... (62)	
第4章 流程控制 (66)	
4.1 结构化程序设计的概念..... (66)	
4.2 选择结构 (67)	

4.2.1 if-else 语句.....	(67)	定义	(122)
4.2.2 if-else 语句的嵌套.....	(69)	6.1.3 指针的运算.....	(124)
4.2.3 if-else if-else 语句.....	(70)	6.2 用指针访问变量.....	(127)
4.2.4 条件表达式	(72)	6.3 用指针访问一维数组.....	(127)
4.2.5 switch 语句和 break 语句.....	(73)	6.4 用指针处理字符串.....	(128)
4.3 循环结构	(77)	6.5 用指针访问二维数组.....	(130)
4.3.1 for 语句.....	(77)	6.5.1 将二维数组视为一维数组 处理.....	(130)
4.3.2 while 语句.....	(79)	6.5.2 行指针.....	(131)
4.3.3 do-while 语句	(81)	6.5.3 指针数组.....	(133)
4.3.4 循环的嵌套	(82)	6.6 二级指针.....	(136)
4.4 转移控制语句.....	(83)	6.6.1 一级指针和二级指针.....	(136)
4.4.1 break 语句.....	(83)	6.6.2 用二级指针访问一维 数组.....	(136)
4.4.2 continue 语句.....	(85)	6.6.3 用二级指针访问二维 数组.....	(137)
4.4.3 exit() 函数.....	(85)	6.7 用指针进行内存动态分配.....	(139)
4.4.4 goto 语句和标号.....	(86)	6.7.1 内存动态分配的含义.....	(139)
4.5 综合举例	(88)	6.7.2 内存动态分配函数.....	(139)
4.5.1 枚举法的应用.....	(88)	本章小结.....	(141)
4.5.2 归纳法的应用	(90)	练习题.....	(141)
本章小结.....	(93)	第7章 函数.....	(147)
练习题.....	(93)	7.1 C 语言程序的模块结构.....	(147)
第5章 数组和字符串.....	(100)	7.1.1 模块化程序设计的 特点.....	(147)
5.1 数组的定义和初始化.....	(100)	7.1.2 C 语言程序的模块 结构.....	(147)
5.1.1 数组的定义.....	(100)	7.2 C 语言函数的定义.....	(148)
5.1.2 数组的存储结构.....	(101)	7.3 C 语言函数的调用.....	(150)
5.1.3 数组的初始化.....	(101)	7.4 函数间数据传递方式之一 ——哑实结合.....	(152)
5.2 数组的赋值.....	(103)	7.4.1 变量的哑实结合.....	(153)
5.3 数组的输入和输出.....	(104)	7.4.2 数组的哑实结合.....	(156)
5.3.1 数值型数组的输入和 输出	(104)	7.4.3 函数的哑实结合 (函数指针)	(159)
5.3.2 字符型数组的输入和 输出	(105)	7.5 函数间数据传递方式之二 ——函数返回值.....	(162)
5.3.3 字符串处理函数.....	(107)	7.6 函数间数据传递方式之三 ——全局变量.....	(164)
5.4 数组的应用举例.....	(108)	7.7 函数的嵌套调用和递归调用.....	(165)
本章小结.....	(113)		
练习题.....	(113)		
第6章 指针.....	(119)		
6.1 地址和指针.....	(119)		
6.1.1 地址.....	(119)		
6.1.2 指针的概念和指针的			

7.7.1 函数的嵌套调用.....	(165)	9.3.2 字符串读写函数 fgets() 和 fputs().....	(206)
7.7.2 函数的递归调用.....	(166)	9.3.3 格式读写函数 fscanf() 和 fprintf().....	(208)
7.8 main() 函数的参数和返回值.....	(168)	9.3.4 数据块读写函数 fread() 和 fwrite().....	(210)
本章小结.....	(170)	9.4 文件检测函数.....	(211)
练习题.....	(170)	9.5 文件的存取.....	(211)
第8章 结构、联合和枚举.....	(177)	9.5.1 文件的定位.....	(211)
8.1 结构类型和结构变量.....	(177)	9.5.2 文件的顺序存取和随机 存取.....	(214)
8.1.1 结构类型和结构变量的 定义.....	(177)	本章小结.....	(217)
8.1.2 结构变量的初始化、赋值 和结构成员的访问.....	(179)	练习题.....	(217)
8.1.3 嵌套结构.....	(181)	第10章 编译预处理和分割编译.....	(223)
8.1.4 结构的存储.....	(182)	10.1 宏定义(#define).....	(223)
8.2 结构数组.....	(183)	10.2 文件包含(#include).....	(227)
8.3 结构指针.....	(183)	10.2.1 文件包含的形式和功能.....	(227)
8.4 结构在函数间的传递.....	(186)	10.2.2 文件包含的嵌套.....	(229)
8.4.1 结构在函数间的传递(哑实 结合).....	(186)	10.3 条件编译.....	(230)
8.4.2 结构型函数和结构指针型 函数(函数返回值方式).....	(189)	10.4 分割编译.....	(232)
8.5 联合(共用体).....	(191)	本章小结.....	(235)
8.6 枚举.....	(195)	练习题.....	(235)
本章小结.....	(196)	第11章 C++语言简介.....	(238)
练习题.....	(196)	11.1 C++语言与面向对象程序 设计.....	(238)
第9章 文件.....	(200)	11.2 C++语言简介.....	(240)
9.1 文件的概念.....	(200)	11.2.1 C++语言对C语言的 扩充.....	(240)
9.2 文件的打开与关闭.....	(202)	11.2.2 用类实施数据封装和数据 隐藏.....	(241)
9.2.1 文件指针.....	(202)	11.2.3 继承和派生类.....	(245)
9.2.2 文件的打开.....	(202)	11.2.4 多态性和虚函数.....	(248)
9.2.3 文件的关闭.....	(204)	附录 练习题参考答案.....	(251)
9.3 文件读写函数.....	(204)		
9.3.1 字符读写函数 getc() 与 fgetc()、 putc() 与 fputc().....	(204)		

第 1 章 C 语言概述

内容提要和学习指导

本章介绍 C 语言的起源、发展和特点, C 程序的基本结构和 C 程序的开发过程, 以及 Turbo C 2.0 集成环境的使用。通过本章的学习, 读者应了解 C 语言产生的历史背景, 掌握 C 语言的特点和 C 程序的开发过程及上机操作步骤, 能使用简单的赋值语句和输入/输出函数编制简单的 C 程序, 并在 Turbo C 集成环境中进行 C 程序的调试和运行。

1.1 C 语言的起源和发展

C 语言的祖先是 ALGOL60(Algorithm Language)。ALGOL60 是 1960 年由国际计算机委员会设计的一种面向过程的高级语言。它是一种结构化语言, 用它编写的程序具有可读性和可移植性好的特点。但是, 它不能直接对硬件进行操作, 不宜用来编写系统程序。系统程序主要用汇编语言编写, 而汇编语言是面向机器的, 用它编写的程序可读性和可移植性都比较差。为此, 人们开始考虑设计一种集高级语言和低级语言特点于一身的语言, 以便使用它来编写可读性和可移植性都比较好的系统程序。

1963 年, 英国剑桥大学和伦敦大学首先将 ALGOL60 发展成 CPL(Combined Programming Language)语言。该语言已比较接近于硬件, 但规模较大, 难以实用。

1967 年, 剑桥大学的 Martin Richards 将 CPL 改制成 BCPL(Basic Combined Programming Language)。BCPL 比 CPL 大为简化, 既具有结构化程序设计语言的特点, 也能直接处理与硬件相关的数据, 被软件人员用作系统程序的描述语言。

1970 年, 美国贝尔实验室的 Ken Thompson 将 BCPL 修改成 B 语言(Boiling CPL down to its basic good feature), 并用 B 语言记述和开发了第一个高级语言 UNIX 操作系统, 在 DEC 公司的 PDP-7 小型机上实现。

1972 年, Ken Thompson 和 Dennis M. Ritchie 将 B 语言修改设计成 C 语言。C 语言既保持了 BCPL 和 B 语言的精练和接近于硬件的特点, 也克服了它们过于简单、数据无类型等缺点。1973 年, Ken Thompson 和 Dennis M. Ritchie 又合作将 1969 年用汇编语言编写的 UNIX 操作系统改用 C 语言编写, 其中 C 语言代码占 90% 以上, 只保留了少量汇编语言代码。这就使得 UNIX 操作系统向其他类型的机器上移植变得相当简单。到了 20 世纪 70 年代中期, UNIX 操作系统和 C 语言作为软件设计师的良好工具传遍了贝尔实验室, 接着也传遍了几乎所有的美国大学校园。随后, UNIX 和 C 语言开始风靡世界。

1978 年, 以 UNIX 第 7 版中的 C 编译程序为基础, Brian W. Kernighan 和 Dennis M. Ritchie 合著了影响深远的《The C Programming Language》。书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础, 称为 K&R C 语言。其后的十几年中, 适用于不同机种和不同操作系统的 C 语言编译系统相继问世, 从而把 C 语言的应用推向了更加广泛普及的阶段。尽管 C

语言源程序的兼容性很强，但由于没有统一的标准而在各种版本之间略有差异。为此，美国国家标准局于 1983 年制定了 C 语言标准。这个标准不断完善，并从 1987 年开始实施，称为 ANSI C。1988 年，K&R 修改了他们的经典著作《The C Programming Language》，按 ANSI C 标准重新编写了该书。现在一般称 ANSI C 为新标准，称 K&R C 为旧标准。此后陆续出现的各种 C 语言编译版本，如 Microsoft C、Turbo C 和 Quick C 等都是与 ANSI C 兼容的新版本。它们的语法和语句功能是一致的，其差异表现在各自的标准函数库中收纳的函数种类、格式和功能上，尤其是图形库的差异更大一些。

C 语言是一种面向过程的语言，意思是用 C 语言编程时可以不必了解计算机的内部逻辑，而主要考虑解题、算法的逻辑和过程的描述。过程的实现必须按照算法逐条语句编写，通知计算机一步一步该怎么做。然而，当一个项目达到一定的规模，C 语言便受到了限制。根据工程项目的要求，25 000 到 100 000 行的程序变得很难管理，因为它很难面面俱到。于是，在 1980 年，当时在贝尔实验室工作的 Bjarne Stroustrup 将许多支持面向对象的编程功能扩充到 C 语言中，起初称之为“带类的 C 语言”，1983 年定名为 C++。C++ 既支持面向过程，也支持面向对象。面向对象程序设计(Object-Oriented Programming, OOP)是一种新的程序设计范例。它不像面向过程那样把精力集中于分析和逐步求精输入输出之间的关系，也不是描述事物之间的逻辑关系，而是从分析问题领域中实体的属性和行为及其相互关系入手，在计算机中建立它们的直接对应物——对象，用对象对应于问题领域中的实体，用消息传递来表达实体间的相互作用、相互关系。对象与关系描述清楚后，就转向解决对象生成的问题即定义所需要的类(class)。在定义类时，通过类继承，可以充分利用系统已有的类，来拼凑所需要的类，之后进行类的实例化，从而得到所期望的对象。面向对象的程序设计可以管理规模庞大的程序，它已经成为当代程序设计的主流。所以，C++ 也成为开发复杂大系统的具有吸引力的语言。由于 C++ 语言以及由 C++ 派生出来的另一个很有吸引力的 Java 语言都是以 C 语言为基础的，所以，学习 C 语言也是进一步学习 C++ 或 Java 语言的重要阶段。

本书介绍的 C 语言将以 ANSI C 为基础，以即“美国信息系统的国家标准——程序设计语言 C，X3.159——1989”为依据。书中的程序是在 Turbo C 2.0 环境下调试的，使用其他 C 编译系统的读者请注意参考有关手册。

1.2 C 语言的特点

在程序设计语言的大家族中，C 语言技压群芳，具有强大的生命力，这主要是 C 语言本身的优势所造成的，是 C 语言的特点决定的。

1. C 语言是一种“中级”语言

C 语言是一种“低级”的高级语言，也是一种“高级”的低级语言，它除了具有一般高级语言所能处理的算术和逻辑运算外，还可以直接进行通常由硬件实现的位、字节和地址(指针)等操作，这一特性使它很适合系统级编程，有“高级汇编语言”之称，以致足以取代汇编语言来编制各种系统软件和应用软件。与汇编语言相比，C 语言通用性强，不局限于某个机器或某个操作系统。

2. C 语言生成的目标代码体积小、质量高

高级语言能否用来描述系统软件，特别是操作系统、编译程序等，除要求语言表达能力

强之外，很重要的一个因素是语言生成的目标代码质量如何。如果目标代码质量低、系统开销大，就没有实用价值。许多实验表明，针对同一问题，用C语言编写的程序，其目标代码的效率仅比汇编语言低 10%~20%。由于用高级语言描述问题比汇编语言迅速、工作量小，可读性和可移植性好，易于调试和修改，所以C语言就成了人们描述系统软件和应用软件的比较理想的工具。

3. C语言基本上是一种结构化程序设计语言

C语言具有顺序、选择和循环三种基本控制结构。它的各种控制语句如 if、while、for、switch、break 等，功能灵活，足以描述结构良好的程序。它还为被广为责难的 goto 语句提供了一席之地，兼顾了程序效率和简洁的要求。C语言的代码和数据分隔特性(使用全局变量和局部变量)、代码块分隔特性(将一组语句放在花括号中)在功能上有助于数据隐藏；C函数的独立性有助于进行模块化程序设计。当然，C语言不允许在函数中定义函数，所以它不像 Pascal 那样是真正的块结构语言。

4. 语言简洁，使用灵活，易于学习和使用

构成C语言命令的关键字仅有 32 个(Turbo C 增加了 11 个，以扩充功能)。C语言在表示方法上力求明了易懂、简单易行。例如，C语言的书写格式没有限制，用赋值运算符(如+=、-=、*=、/=、++、--等)实现既运算又赋值；提供了预处理能力，包括宏定义、文件蕴含、条件编译等，可以在源程序编译时方便、灵活地进行修改；C语言还把一般语言的许多成分通过显式调用来完成，使得编译程序小而精。C语言并未提供依赖于机器硬件的输入/输出语句，也没有并行操作、同步等复杂控制，而是提供大量而有效的函数库来实现输入/输出、字符串处理、数学计算、动态存储分配等功能，函数库可根据需要方便地扩充。这些不仅保证了C语言移植性好，而且运行时所需支持少，存储空间占用也小。

5. C语言具有很强的数据处理能力

C语言具有 5 种基本数据类型，允许几乎所有的类型转换，例如，整型和字符型可以在绝大多数表达式中自由混用。还可以在基本数据类型的基础上按层次产生各种构造类型和集合类型，如数组、结构、联合、枚举等，利用它们可以处理各种复杂的数据结构。

6. 可移植性好

C语言是伴随 UNIX 操作系统诞生的。1978 年以后，C语言已先后移植到大、中、小型和微型计算机上，已独立于 UNIX 了。现在C语言编译系统几乎可以在所有的操作系统及各种类型的机器上运行。统计资料表明，不同机器上的C编译程序有80%的代码是公用的，因此移植起来很简便。

7. 绘图功能强

C语言在庞大的图形程序库的支持下，可进行二维、三维图形和动画的设计，也可以用于计算机辅助设计。

C语言具有上述诸多优点，因而具有旺盛的生命力。但也有一些缺点和不足，如运算符的优先级太多，不便于记忆，有些运算符的优先级还与常规约定不同；类型转换比较随便；检测手段太弱(如C程序运行时不执行数组下标越界、变量类型的可兼容性等检查)，安全性较差，要求程序员对程序设计更熟练、查错能力更强。但是，瑕不掩瑜，多种程序设计的应用已经证明C语言是一种极为有效且表达能力很强的语言。

1.3 C 语言程序的基本结构

1.3.1 一个简单的 C 语言程序

我们先通过一个简单的 C 语言程序，了解一个 C 语言程序是什么样子。

【例 1.1】输入圆的半径，计算圆的周长和面积。

解：程序中先定义了一个符号常量 PI(代表 π)，然后定义三个浮点型变量，分别代表半径、面积和周长，并根据给定的半径 r 值依次计算面积和周长，最后输出结果。

```
#include <stdio.h>
#define PI 3.14159
main()                /* 函数名 */
{                    /* 函数体开始 */
    float r,a,c;      /* r:半径; a:面积; c:周长 */
    r=2.5;            /* 给定 r 的值 */
    a=PI*r*r;         /* 计算 a */
    c=2*PI*r;         /* 计算 c */
    printf("r=%f,a=%f,c=%f\n",r,a,c); /* 输出 r,a,c */
}                    /* 函数体结束 */
```

以上程序运行结果如下：

r=2.500000, a=19.634956, c=15.707965

上面的程序中，main 是主函数名，C 语言规定必须用 main 作主函数名，其后的一对圆括号不能省略；函数体是程序的定义和执行部分，左花括号表示函数体的开始，右花括号表示函数体的结束；函数体中有五条 C 语句，这是程序的功能所在。

在程序的开始，还有两个用“#”开头的命令行，它们的含义将在下面介绍。

1.3.2 C 语言程序的基本结构

下面再通过一个复杂一点的例子，了解 C 语言程序的基本结构。

【例 1.2】计算二整数相乘的程序

解：我们把这个大问题分解成两个子问题：一个是子函数，解决任意两个整数相乘的问题；另一个是主函数，用来输入两个变量的值，并调用子函数，最后打印结果。这样就形成了一个含有主函数和子函数的 C 语言程序。

```
#include <stdio.h>                /* 文件蕴含 */
main()                            /* 主函数名 */
{                                /* 函数体开始 */
    int x,y,z;                   /* 局部变量类型定义 */
    scanf("%d%d",&x,&y);          /* 输入变量值 */
    z=product(x,y);              /* 调用计算乘积的函数 */
    printf("x=%d,y=%d\n",x,y);  /* 打印 x 和 y 的值 */
}
```

```

        printf("x*y=%d\n",z);          /* 打印乘积值 */
    }                                  /* 函数体结束 */
product(int a,int b)                  /* 子函数及其形式参数 */
{
    int c;                            /* 定义局部变量 */
    c=a*b;                            /* 计算乘积 */
    return(c);                        /* 返回值 */
}

```

程序运行时，用户若从键盘输入 5 和 8(5 和 8 用空格隔开)并按回车键后，程序输出为

```

x=5,y=8
x*y=40

```

C 语言程序的基本结构是函数。

1. 一个 C 语言程序由一个或多个 C 函数组成，函数也叫模块，是完成某个整体功能的最小单位，其中必须有且只能有一个用“main”命名的主函数。
2. 一个 C 函数由若干条 C 语句组成，C 语句是完成某种程序功能(如赋值、输入、输出等)的最小单位。
3. 一条 C 语句由若干个基本单词组成，基本单词是构成语句的最小单位。

C 语言程序的一般格式如图 1.1 所示，其中 f1()到 fn()表示用户定义的函数。

```

全局变量说明
main()
{
    局部变量说明
    语句序列
}
f1(形式参数表)
{
    局部变量说明
    语句序列
}
f2(形式参数表)
{
    局部变量说明
    语句序列
}
⋮
fn(形式参数表)
{
    局部变量说明
    语句序列
}

```

图 1.1 C 语言程序的一般格式

1.3.2.1 基本单词

C 语言共有五种基本单词，即关键字(亦称保留字)、标识符、常数、操作符和分隔符。

1. 关键字

关键字是 C 语言中有特定意义和用途、不得作它用的字符序列，其中 ANSI C 标准规定的关键字有 32 个：

基本数据类型关键字：char int float double void

存储类型关键字：auto register static extern

类型修饰关键字：long short signed unsigned const volatile

流程控制关键字：if else switch case default break continue for
while do return goto

构造数据类型关键字：struct union enum

类型定义关键字：typedef

求数据类型长度关键字：sizeof

所有 C 语言关键字都必须小写。在 C 语言中，大写字母和小写字母是有区别的。

2. 标识符

标识符是 C 语言中用来表示变量名、数组名、函数名、指针名、结构名、联合名、枚举常量名、用户定义的数据类型名及语句标号等用途的字符序列，可由 1~32 个字符组成，第一个字符必须是字母或下划线，后面的字符可以是字母、数字或下划线。例如 AB, Ab, aB, ab, A_b, _ab, ab_, s2d, W_length 等都是正确的标识符，而 A+B, A'B, A.B, 2abc, α, β, d%等都是错误的标识符。

标识符不能与 C 语言关键字相同，并区分大小写。例如 AB 和 Ab 是两个不同的标识符，ELSE 可以作为标识符，而不是 C 语言关键字。

3. 常数

常数包括数值常数(如 123、-23.5、1.2E4 等)、字符常数(如 'a'、'B'、'c' 等)、字符串常数(如 "xyz"、"good morning" 等)、符号常数以及枚举常数(见第 8 章)。

4. 操作符

操作符包括各种运算符(如 +、-、*、/)，有特定意义的标点符号(如花括号、方括号、圆括号、逗号)等。

5. 分隔符

分隔符用来分隔相邻的标识符、关键字和常数，最常用的分隔符是空格，此外，制表符、换行符、换页符等也可用作分隔符。

例如语句

```
float r,a,c;
```

中，float 是关键字，代表数据类型；r、a 和 c 是标识符(这里表示变量名)；又如语句

```
a=PI*r*r, c=2*PI*r;
```

中，“=”、“*”和“,”是运算符；“2”是常数，PI 是符号常数；在“a=PI*r*r”和“c=2*PI*r”之间可以用一个或多个空格分开，这些空格是分隔符。

1.3.2.2 C 语句

C 语句是组成 C 语言程序的基本单位，具有独立的程序功能。所有的 C 语句都以分号