

 普通高等教育计算机规划教材

新编C语言 程序设计教程

钱雪忠 吕莹楠 高婷婷 主编



提供电子教案

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



普通高等教育计算机规划教材

新编 C 语言程序设计教程

钱雪忠 吕莹楠 高婷婷 主编

宋威 吴秦 程建敏 王雪茹 参编



机械工业出版社

本书在编者多年教学实践的基础上编写而成,在有所创新的同时,希望能做到:概念清晰但不烦琐;例题精选又不失通用性;从实际操作出发更注重应用编程能力;把握语言知识点又敢于面对能力考核。

本书内容全面,重点突出,共含13章,主要内容包括:C语言概述、结构化程序设计与算法、数据类型及其运算、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组及其应用、函数及其应用、指针及其应用、自定义类型及其应用、文件及其应用、预处理命令、位运算等。

本书既可作为高等院校理工科各专业“C语言程序设计”类课程的教材,也可供参加自学考试人员、应用系统开发设计人员、工程技术人员及其他对程序设计感兴趣的读者参阅。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 2399929378, 电话: 010-88379753)。

图书在版编目(CIP)数据

新编 C 语言程序设计教程 / 钱雪忠, 吕莹楠, 高婷婷主编.

—北京: 机械工业出版社, 2013.7

普通高等教育计算机规划教材

ISBN 978-7-111-43012-4

I. ①新… II. ①钱… ②吕… ③高… 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(2013)第 136250 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 和庆娣 王 凯

责任印制: 李 洋

中国农业出版社印刷厂印刷

2013 年 7 月第 1 版·第 1 次印刷

184mm×260mm·18.75 印张·465 千字

0001—3000 册

标准书号: ISBN 978-7-111-43012-4

定价: 39.90 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

教材网: <http://www.cmpedu.com>

销售一部: (010) 68326294

机工官网: <http://www.cmpbook.com>

销售二部: (010) 88379649

机工官博: <http://weibo.com/cmp1952>

读者购书热线: (010) 88379203

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- 1) 反映计算机技术领域的新发展和新应用。
- 2) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、多媒体光盘、课程设计和毕业设计指导等内容。
- 3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- 4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- 5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

C 语言程序设计是国内外广泛使用的计算机程序设计语言，是高等院校相关专业重要的专业基础课程。由于 C 语言具有功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好等特点，自 20 世纪 90 年代以来便迅速在全世界普及推广。目前，它仍然是最优秀的程序设计语言之一。

本书是编者在一线教学实践的基础上，为适应当前本科教育教学改革创新的要求，更好地践行语言类课程注重实践教学与创新能力培养的需要，组织新编而成的教程。教程编写中融合了同类其他教材的优点，并力求创新，新编教程具有以下特点：

1) 突出 C 语言实用的重点概念，在重点概念讲明白的情况下，并不求语法概念的详尽与全面，而只求轻快明晰、循序渐进、通俗易懂、深入浅出。

2) 精选例题，引入了大量趣味性、游戏性应用实例，注重与加强程序阅读、参考、编写和上机调试实践的能力，重在编程思想的培养与训练。

3) 从实际操作出发，发现问题解决问题，举一反三，一题多解，增强实用能力。

4) 明晰 C 语言各语言成分的意义与价值，以“数据+算法”为核心，提高读者编程能力。

5) 基本知识学习、上机实验、典型习题与知识点把握等多方面相结合，扎实掌握相关知识，敢于面对 C 语言能力考核。

6) 语言编程环境以 Visual C++ 6.0 为主，同时能兼顾 Turbo C、Win-TC 等传统简易编程环境，比较不同编程环境程序运行差异，能更好地了解语言程序与编译器的依存关系。

本书内容充实全面，每章除基本知识外，还有章节要点、应用实例、本章小结、适量习题、本章实验等，以配合对知识点的掌握。教师在课程讲授时可根据学生、专业、课时等情况对内容适当取舍。

本书由钱雪忠、吕莹楠、高婷婷主编，由江南大学、黑龙江东方学院相关师生合作编写，参与编写的还有宋威、吴秦、程建敏、王雪茹。参与程序调试的有钱恒、任看看、马亮、施亮、邓杰等。编写中还得到了江南大学物联网工程学院智能系统与网络计算研究所同仁们的大力协助与支持，使编者获益良多，谨此表示衷心的感谢。

由于时间仓促，编者水平有限，书中难免有疏漏之处，敬请广大读者与同行专家批评指正。

编 者

目 录

出版说明

前言

第 1 章 C 语言概述	1	2.5 本章小结	27
1.1 程序设计语言简介	1	2.6 习题	27
1.2 C 语言发展过程	3	实验 2 熟悉 VC++ 6.0 环境及 算法	29
1.3 C 语言是优秀的程序语言	4	第 3 章 数据类型及其运算	30
1.4 C 语言与 C++ 的关系	5	3.1 数据类型	30
1.5 初识简单的 C 语言程序	6	3.2 常量与变量	31
1.6 C 程序的结构特点	9	3.2.1 常量	31
1.7 C 程序的书写规则	10	3.2.2 变量	32
1.8 C 语言字符集与词汇	10	3.3 整型数据	32
1.9 运行 C 程序	12	3.3.1 整型常量	32
1.10 本章小结	13	3.3.2 整型变量	33
1.11 习题	14	3.4 实型数据	37
实验 1 初识运行环境和运行 过程	15	3.4.1 实型常量	37
第 2 章 结构化程序设计与算法	18	3.4.2 实型变量	37
2.1 初识算法	18	3.5 字符型数据	40
2.1.1 算法的概念	18	3.5.1 字符常量	40
2.1.2 算法举例	19	3.5.2 转义字符	41
2.1.3 算法的特征	20	3.5.3 字符变量	41
2.2 结构化程序设计	20	3.5.4 字符数据的存储与使用	41
2.2.1 结构化程序设计方法	20	3.5.5 字符串常量	42
2.2.2 结构化程序设计方法的 原则	21	3.6 变量赋初值	43
2.2.3 结构化程序设计的三种基本 结构	21	3.7 数据类型的转换	43
2.3 算法的表示方法	22	3.8 算术运算符和表达式	45
2.3.1 用自然语言表示算法	23	3.8.1 运算符简介	45
2.3.2 用流程图表示算法	23	3.8.2 算术运算符和表达式	45
2.3.3 用 N-S 流程图表示算法	24	3.9 赋值运算符和表达式	48
2.3.4 用伪代码表示算法	25	3.10 逗号运算符和表达式	49
2.3.5 用计算机语言表示算法	26	3.11 应用实例	50
2.4 结构化程序设计应用举例	26	3.12 本章小结	52
		3.13 习题	53
		实验 3 数据类型及其运算	54

第 4 章 顺序结构程序设计	56	6.7 break 和 continue 语句	100
4.1 C 语言语句概述	56	6.7.1 break 语句	100
4.2 C 语言赋值语句	57	6.7.2 continue 语句	100
4.3 数据输入和输出的概念	58	6.8 应用实例	101
4.4 字符数据的输入和输出	58	6.9 本章小结	111
4.4.1 putchar 函数	58	6.10 习题	112
4.4.2 getchar 函数	59	实验 6 循环结构程序设计	116
4.5 格式数据的输入和输出	60	第 7 章 数组及其应用	118
4.5.1 格式输出函数 printf	60	7.1 一维数组的定义和引用	118
4.5.2 格式输入函数 scanf	64	7.1.1 一维数组的定义	118
4.6 应用实例	68	7.1.2 一维数组元素的引用	119
4.7 本章小结	72	7.1.3 一维数组的初始化	120
4.8 习题	72	7.1.4 一维数组程序举例	121
实验 4 顺序结构程序设计	74	7.2 二维数组的定义和引用	123
第 5 章 选择结构程序设计	75	7.2.1 二维数组的定义	123
5.1 关系运算符和表达式	75	7.2.2 二维数组元素的引用	124
5.1.1 关系运算符及其优先级	75	7.2.3 二维数组的初始化	125
5.1.2 关系表达式	75	7.3 字符数组	127
5.2 逻辑运算符和表达式	76	7.3.1 字符数组的定义	127
5.2.1 逻辑运算符及其优先级	76	7.3.2 字符数组的初始化	127
5.2.2 逻辑运算及其取值	77	7.3.3 字符数组的引用	127
5.2.3 逻辑表达式	77	7.3.4 字符串和字符串结束标志	128
5.3 if 语句	78	7.3.5 字符数组的输入和输出	128
5.3.1 if 语句的三种形式	78	7.3.6 字符串处理函数	129
5.3.2 if 语句的嵌套	81	7.4 应用实例	132
5.3.3 条件运算符和表达式	83	7.5 本章小结	139
5.4 switch 语句	83	7.6 习题	139
5.5 应用实例	85	实验 7 数组及其应用	142
5.6 本章小结	89	第 8 章 函数及其应用	144
5.7 习题	90	8.1 C 语言函数概述	144
实验 5 选择结构程序设计	92	8.2 函数定义的一般形式	145
第 6 章 循环结构程序设计	94	8.3 函数的参数和函数的值	147
6.1 概述	94	8.3.1 形式参数和实际参数	147
6.2 goto 语句	94	8.3.2 函数的返回值	148
6.3 while 语句	95	8.4 函数的调用	149
6.4 do-while 语句	96	8.4.1 函数调用的一般形式	149
6.5 for 语句	97	8.4.2 函数调用的方式	149
6.6 循环的比较及其嵌套	99		

8.4.3 被调用函数的声明	150	9.7 指向指针的指针	206
8.5 函数的嵌套调用	151	9.8 main 函数参数	207
8.6 函数的递归调用	152	9.9 应用实例	208
8.7 数组作为函数参数	155	9.10 本章小结	211
8.8 局部变量和全局变量	158	9.11 习题	213
8.8.1 局部变量	159	实验 9 指针及其应用	216
8.8.2 全局变量	160	第 10 章 自定义类型及其应用	218
8.9 变量的存储类别	161	10.1 如何定义结构体	218
8.9.1 auto 变量	162	10.2 结构体类型变量的说明	219
8.9.2 用 static 声明局部变量	162	10.3 结构体变量	220
8.9.3 register 变量	163	10.3.1 结构体变量成员的 表示方法	220
8.9.4 用 static 声明全局变量	164	10.3.2 结构体变量的赋值	221
8.10 内部函数和外部函数	166	10.3.3 结构体变量的初始化	221
8.11 函数与模块化程序设计	166	10.3.4 指向结构体变量的指针	222
8.12 应用实例	167	10.4 结构体数组	223
8.13 本章小结	173	10.4.1 结构体数组的定义	223
8.14 习题	173	10.4.2 指向结构体数组的 指针	224
实验 8 函数及其应用	176	10.5 结构体指针变量作函数 参数	225
第 9 章 指针及其应用	178	10.6 C 语言动态存储分配	226
9.1 指针的基本概念	178	10.7 C 语言链表的概念	227
9.2 指针变量	179	10.8 共用体	229
9.2.1 指针变量的定义	179	10.8.1 共用体的定义	230
9.2.2 指针变量的引用	180	10.8.2 共用体变量的说明	230
9.2.3 指针变量作为函数参数	183	10.8.3 共用体变量的赋值和 使用	231
9.2.4 指针变量的几个问题	186	10.9 C 语言枚举类型	232
9.3 指针与数组	189	10.9.1 枚举类型的定义及其变量 说明	232
9.3.1 指向数组元素的指针	189	10.9.2 枚举类型变量的赋值和 使用	232
9.3.2 通过指针引用数组	189	10.10 C 语言类型定义符 typedef 详解	234
9.3.3 数组名作函数参数	191	10.11 应用实例	234
9.3.4 指向多维数组的指针	195	10.12 本章小结	241
9.4 指针与字符串	197	10.13 习题	242
9.4.1 字符串的表示形式	197		
9.4.2 字符(串)指针变量与 字符数组	200		
9.5 指针与函数	201		
9.5.1 函数指针变量	201		
9.5.2 指针型函数	202		
9.6 指针数组	203		

实验 10 自定义类型及其应用	245	12.1 C 语言预处理概述	265
第 11 章 文件及其应用	247	12.2 宏定义	265
11.1 C 语言文件概述	247	12.2.1 无参宏定义	266
11.2 文件指针	248	12.2.2 有参宏定义	268
11.3 文件的打开与关闭	249	12.3 文件包含命令	272
11.3.1 文件打开函数 fopen	249	12.4 C 语言条件编译	272
11.3.2 文件关闭函数 fclose	251	12.5 应用实例	274
11.4 文件的读写	251	12.6 本章小结	276
11.4.1 字符读写函数 fgetc 和 fputc	251	12.7 习题	276
11.4.2 字符串读写函数 fgets 和 fputs	254	第 13 章 位运算	278
11.4.3 数据块读写函数 fread 和 fwrite	255	13.1 C 语言位运算符	278
11.4.4 格式化读写函数 fscanf 和 fprintf	256	13.2 C 语言位域 (位段)	280
11.5 文件的随机读写	256	13.3 应用实例	282
11.5.1 文件定位	257	13.4 本章小结	284
11.5.2 文件的随机读写	257	13.5 习题	284
11.6 文件检测函数	258	附录	286
11.7 库文件	258	附录 A ASC II 与扩展 ASC II 编码表	286
11.8 应用实例	259	附录 B C 语言关键字大全	287
11.9 本章小结	260	附录 C C 语言运算符及其 优先级	288
11.10 习题	260	附录 D VC++6.0 程序调试常见 错误信息	289
实验 11 文件及其应用	263	附录 E C 语言库函数	291
第 12 章 预处理命令	265	参考文献	292

第 1 章 C 语言概述

C 语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有低级语言的许多特点，又具有高级语言的优点，既适于编写系统软件，又能方便地用来编写应用软件。20 世纪 90 年代以来，C 语言迅速在全世界普及推广。目前，它仍然是最优秀的程序设计语言之一。

学习重点和难点：

- C 语言介绍与语言的特点
- C 语言程序结构
- 程序运行环境

读者在学习本章后，将对 C 语言及 C 语言程序有初步认识，并能开展 C 语言程序的运行与实践。

1.1 程序设计语言简介

语言？程序？程序设计？

自从第一台计算机诞生以来，程序设计语言和程序设计方法不断发展。

语言是思维的载体。人和计算机打交道，必须要解决“语言”沟通的问题。计算机并不能理解和执行人们使用的自然语言，而只能接受和执行二进制的指令。计算机能够直接识别和执行的这种指令，称为机器指令，这种机器指令的集合就是机器语言指令系统，简称为机器语言。为了解决某一特定问题，需要选用指令系统中的某一些指令，将其按要求选取并组织起来就组成一个“程序”。如下程序是 8086 指令系统对应的二进制代码程序，能完成两个十六进制数相加的功能：

```
10111000 00111111100001011
10001110 11011000
10100001 0000000000000010
00000001 00000110 0000000000000000
10110100 01001100
11001101 00100001
```

换言之，一个程序是完成某一特定任务的一组指令序列，或者说，为实现某一算法的指令序列称为“**程序**”，机器世界中真正存在的就是这样的二进制程序。

用机器语言编制的程序虽然能够直接被计算机识别、执行，但是机器语言本身随不同类型的机器而异，所以可移植性差，而且机器语言本身难学、难记、难懂、难修改，给使用者带来极大的不便。于是，为了绕开机器指令，克服机器指令程序的缺陷，人们提出了程序设计语言的构想，即使用人们熟悉、习惯的语言符号来编写程序，最好是直接使用人们交流的自然语言来编程。在过去的几十年中，人们创造了许多介于自然语言和机器指令之间的程序设计语

言。按语言的级别大致可分为：汇编语言（低级）和高级语言（第三代、第四代、……）。

汇编语言的特点是使用一些“助记符号”来替代那些难懂难记的二进制代码，所以汇编语言比机器指令便于理解和记忆，但它和机器语言的指令基本上又是一一对应的，两者都针对特定的计算机硬件系统，可移植性差，因此称它们都是“面向机器的低级语言”。为了直观地了解汇编语言程序，如下给出一段实现 X、Y 两个 16 位二进制数相加的 8086 汇编程序：

```
;X, Y 分别为 16 位二进制数，程序实现 X=X+Y（不考虑 溢出）。
DATA SEGMENT                                ;定义数据段开始
X DW 123H                                    ;定义一个字变量（16 位）X
Y DW 987H                                    ;定义一个字变量（16 位）Y
DATA ENDS                                    ;定义数据段结束
CODE SEGMENT                                ;定义代码段开始
ASSUME CS:CODE,DS:DATA                      ;建立段寄存器与段名之间的映射关系
START:MOV AX,DATA                            ;取 DATA 段地址送 AX 寄存器
MOV DS,AX                                    ;将数据段段地址送数据段寄存器 DS
MOV AX,Y                                     ;取变量 Y 值送给寄存器 AX
ADD X,AX                                     ;将 X 的值与 AX 的内容相加，结果送给 X，实现 X=X+Y
MOV AH,4CH                                   ;将 DOS 调用的 4CH 功能号送 8 位寄存器 AH
INT 21H                                     ;执行 DOS 功能调用，退出程序，回到 DOS
CODE ENDS                                    ;定义代码段结束
END START                                    ;源程序结束，主程序从标号 START 开始
```

高级语言类似自然语言（主要是英语），由专门的符号根据词汇规则构成单词，由单词根据句法规则构成语句，每种语句有确切的语义并能由计算机解释。高级语言包含许多英语单词，有“自然化”的特点；高级语言书写计算式子接近于熟知的数学公式的规则；高级语言与机器指令完全分离，具有通用性，一条高级语言语句常常相当于几条或几十条机器指令。所以，高级语言的出现，给程序设计从形式和内容上都带来了重大变革，大大方便了程序的编写，提高了可读性。例如：BASIC、C、Visual Basic（简称 VB）、Visual C++（简称 VC++）、VB.NET、C#.NET、Java 等都是高级语言。高级语言一般能细分为第三代高级语言、第四代高级语言、……，分类依据是高级语言的逻辑级别、表达能力、接近自然语言的程度等。如 Turbo C 2.0（简称 TC）为第三代高级语言，而 VB 6.0、VC++ 6.0、C#、VB.NET、Java 等可认为是第四代高级语言。第四代高级语言一般是具有面向对象特性、具有快速或自动生成部分应用程序能力的高级语言，它表达能力强，编写程序效率高，更接近人的使用语言，高一级别的语言一般具有低一级别语言的语言表达能力。如下是输入两个整数并随即显示两整数之和的 Turbo C 2.0 语言程序：

```
#include <stdio.h>                          /* Turbo C 2.0 在 DOS 环境运行 */
main()
{ int num1,num2;
  printf("Input two numbers: ");             /* 屏幕上显示输入提示 */
  scanf("%d %d",&num1,&num2);                 /* 通过键盘读两个整数 */
  printf("The sum is %d\n",num1+num2)        /* 屏幕上显示两整数之和 */
}
```

显然，高级语言程序要比面向机器的低级语言易懂、明了、简短得多。

应该看到的是，高级语言是不断发展变化的，不断有新的更好的语言产生，同时也有旧的且功能差而不再实用的语言消亡。但 C 语言自产生以来，已历经 40 余年，依然具有强大的生命力与活力，该语言依然是当今最热门、最实用的高级语言之一。

1.2 C 语言发展过程

在学习 C 语言之前，首先来介绍一下 C 语言的历史。

C 语言是一门通用的、模块化、程序化的编程语言，被广泛应用于操作系统和应用软件的开发。由于其高效和可移植性，适应于不同硬件和软件平台，深受开发人员的青睐。

1. C 语言早期发展

1969~1973 年，AT&T 贝尔实验室开始了 C 语言的最初研发。根据 C 语言的发明者丹尼斯·里奇（Dennis Ritchie）说，C 语言最重要的研发时期是在 1972 年。

1970 年，美国贝尔实验室的 Ken Thompson 以 BCPL（Basic Combined Programming Language）语言为基础，设计出很简单且很接近硬件的 B 语言（取 BCPL 的首字母），并且用 B 语言写了第一个 UNIX 操作系统。

1972 年，美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二字母作为这种语言的名字，即 C 语言。为此，C 语言的祖先是 BCPL 语言。

C 语言的诞生是和 UNIX 操作系统的开发密不可分的，原先的 UNIX 操作系统都是用汇编语言写的，1973 年 UNIX 操作系统的核心用 C 语言改写，从此以后，C 语言成为编写操作系统的主要语言。

2. ANSI C 标准

20 世纪 70~80 年代，C 语言被广泛应用，从大型主机到小型微机，也衍生了 C 语言的很多不同版本。

为统一 C 语言版本，1983 年美国国家标准局（American National Standards Institute，简称 ANSI）成立了一个委员会，来制定 C 语言标准。1989 年 C 语言标准被批准，被称为 ANSI X3.159-1989 “Programming Language C”。这个版本的 C 语言标准通常被称为 ANSI C（C89）。目前，几乎所有的开发工具都支持 ANSI C 标准，它是 C 语言用得最广泛的一个标准版本。

3. C99 标准

在 ANSI C 标准确立之后，C 语言的规范在很长一段时期内都没有大的变动。1995 年，WG14 小组对 C 语言进行了一些修改，成为后来的 1999 年发布的 ISO/IEC 9899:1999 标准，通常被称为 C99 标准。但是各个公司对 C99 标准的支持所表现出来的兴趣不同。当 GCC 和其他一些商业编译器支持 C99 标准的大部分特性时，微软和 Borland 却似乎对此不感兴趣。

4. ISO 发布 C 语言标准新版本

ISO（International Organization for Standardization）于 2011 年 4 月正式公布 C 语言新的国际标准草案，之前被命名为 C1X 的新标准将被称为 ISO/IEC 9899:2011（C11 版）。新的标准修订了 C11 版本，提高了对 C++ 的兼容性，并将新的特性增加到 C 语言中。

新功能包括支持多线程，基于 ISO/IEC TR 19769:2004 规范下支持 Unicode，提供更多用

于查询浮点数类型特性的宏定义和静态声明功能等。根据草案规定，最新发布的标准草案修订了许多特性，支持当前的编译器。

5. C 语言对其他语言的影响

很多编程语言都深受 C 语言的影响，比如 C++（以前是 C 语言的一个扩展）、C#、Java、PHP、Javascript、Perl、LPC 和 UNIX 的 C Shell。也正因为 C 语言的影响力，掌握 C 语言的人再学其他编程语言，大多能很快上手，触类旁通。

6. 目前 C 语言的商用版本

目前最流行的 C 语言有以下几种：① Microsoft C 或称 MS C；② Borland C（简称 BC）或 Turbo C；③ Win-TC；④ AT&T C；⑤ Objective-C。这些 C 语言版本不仅实现了 ANSI C 标准（C89 版），而且在此基础上各自作了一些扩充，使之更加方便与实用。

1.3 C 语言是优秀的程序语言

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 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。

1. C 语言的优点

（1）语言简洁，使用方便灵活

C 语言是现有程序设计语言中规模最小的语言之一，而小的语言体系往往能设计出较好的程序。C 语言的关键字很少，ANSI C 标准一共只有 37 个关键字，9 种控制语句，压缩了一切不必要的成分。C 语言的书写形式比较自由，表达方法简洁，使用一些简单的方法就可以构造出相当复杂的数据类型和程序结构。

（2）可移植性好

用过汇编语言的读者都知道，即使是功能完全相同的一种程序，对于不同的单片机，必须采用不同的汇编语言来编写，这是因为汇编语言完全依赖于单片机硬件。而现代社会中新硬件的更新换代速度非常快，如果每接触一种新的单片机就要学习一次新的汇编语言，那么我们将没有多少时间真正用于产品开发。

C 语言是通过编译得到可执行代码的，统计资料表明，不同机器上的 C 语言编译程序 80%的代码是公共的，C 语言的编译程序便于移植，从而使在一种单片机上使用的 C 语言程序可以不加修改或稍加修改即可方便地移植到另一种结构类型的单片机上。

（3）数据结构类型丰富，表达能力强

C 语言具有丰富的数据结构类型，可以根据需要采用整型、实型、字符型、数组类型、指针类型、结构类型、共用体类型、枚举类型等多种数据类型来实现各种复杂数据结构的运算。C 语言还具有多种运算符，灵活使用各种运算符可以实现其他高级语言难以实现的运算。

（4）运算符多，表达方式灵活

C 语言提供了多达 34 种运算符，并把括号、赋值、逗号等都作为运算符处理，它们可以组成各种表达式，还可采用多种方法来获得表达式的值，从而使用户在程序设计中具有更大的灵活性。C 语言的语法规则不太严格，程序的书写格式自由灵活，主要用小写字母来编写，而小写字母是比较容易阅读的，这些都充分体现了 C 语言灵活、方便和实用的特点。

(5) 可进行结构化程序设计

C 语言以函数作为程序设计的基本单位，C 语言程序中的函数相当于汇编语言中的子程序。C 语言对于输入和输出的处理也是通过函数调用来实现的。各种 C 语言编译器都会提供一个函数库，其中包含有许多标准函数，如各种数学函数、标准输入、输出函数等。此外，C 语言还具有自定义函数的功能，用户可以根据自己的需要编写自定义函数。实际上 C 语言程序就是由许多个函数组成的，一个函数相当于一个程序模块，因此 C 语言可以很容易地进行结构化程序设计。

(6) 可以直接操作计算机硬件

C 语言具有直接访问单片机物理地址的能力，可以直接访问片内或片外存储器，还可以进行各种位 (bit) 操作。

(7) 生成的目标代码效率高

众所周知，汇编语言程序目标代码的效率是最高的，这就是为什么汇编语言仍是编写计算机系统软件的重要工具的原因。但是统计表明，对于同一个问题，用 C 语言编写的程序生成代码的效率仅比用汇编语言编写的程序低 10%~20%。

2. C 语言的不足

尽管 C 语言具有很多的优点，但和其他任何一种程序设计语言一样也有其自身的缺点，如不能自动检查数组的边界，各种运算符的优先级别太多，某些运算符具有多种用途，比学习其他高级语言要困难一些等。

但总的来说，C 语言的优点远远超过它的缺点。所以，C 语言有很广阔的应用领域。

3. C 语言的应用领域

1) 应用软件。Linux 操作系统中的应用软件都是使用 C 语言编写的，安全性非常高。

2) 对性能要求严格的领域。一般对性能有严格要求的领域都是用 C 语言编写的，比如网络程序的底层和网络服务器端底层、地图查询等。

3) 系统软件和图形处理。C 语言具有很强的绘图能力和可移植性，并且具备很强的数据处理能力，可以用来编写系统软件、制作动画、绘制二维图形和三维图形等。

4) 数字计算。相对于其他编程语言，C 语言是数字计算能力超强的高级语言。

5) 嵌入式设备开发。手机、PDA 等时尚消费类电子产品内部的应用软件、游戏等，很多都是采用 C 语言进行嵌入式开发的。

6) 游戏软件开发。利用 C 语言可以开发很多游戏，比如推箱子、贪吃蛇等。

1.4 C 语言与 C++的关系

C 语言是 C++语言的基础，C++是在 C 的基础上增加了新的面向对象理论，两者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效果。

C 语言是一种结构化语言，它的重点在于算法和数据结构。C 程序的设计首要考虑的是如何通过一个过程，对输入（或环境条件）进行运算处理得到输出（或实现过程（事务）控制）。

C++首要考虑的是如何构造一个对象模型，让这个模型能够契合与之对应的问题域，这样就可以通过获取对象的状态信息得到输出或实现过程（事务）控制。所以 C 与 C++的最大

区别在于它们用于解决问题的思想方法不一样。之所以说 C++比 C 更先进，是因为“设计这个概念已经被融入到 C++之中”。

C 语言与 C++的区别有很多，下面是简要概述：

1) 全新的程序设计思维，C 语言是面向过程的，而 C++是面向对象的。

2) C 语言有标准的函数库，它们是松散的，只是把功能相同的函数放在一个头文件中；而 C++对于大多数的函数都是紧密集成的，特别是 C 语言中没有 C++中的 API，这些 API 是对 Window 系统的大多数 API 的有机组合，是一个集体。

3) C++中的图形处理和 C 语言的图形处理有很大的区别。C 语言中的图形处理函数基本上是不能用在 C++中的，C 语言标准中也不包括复合或复杂的图形处理功能。

4) C 和 C++中都有结构的概念，但是在 C 语言中结构只有成员变量，无成员方法，而在 C++的结构中，它可以有自己的成员变量和成员函数。C 语言中结构的成员变量是公共的，都可以访问；而在 C++中没有加限定符的均为私有的，而非公共的。

5) C++与 C 语言相比，可以写更多更好的程序，比如写基于 DOS 的程序、写 DLL、写控件、写系统。

6) C 语言对程序文件的组织是松散的，几乎是全要程序处理；而 C++对文件的组织是以工程为总领，各文件分类明确并汇集于工程中。

7) C++中的 IDE（集成开发环境）很智能，和 VB 一样，有的功能可能比 VB 还强。C++可以自动生成想要的程序结构，节省很多时间。C++中的附加工具也有很多，可以进行系统分析、查看 API 和控件等。C++调试功能也很强大，并且方法多样。

但是，C++功能虽强，它是以 C 语言为基础的，也就是说学习掌握 C 语言才是首要的。

1.5 初识简单的 C 语言程序

为了说明 C 语言程序结构的特点，先看以下几个程序。这几个程序由易到难，表现了 C 语言程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中直观地了解到组成一个 C 语言程序的基本部分和书写格式等。

【例 1-1】 显示“Hello, World!”。

注意：程序行后用“/* ...*/”或“//”引出的内容为注释部分，起到说明语句或程序的作用，程序不执行此部分。

```
#include <stdio.h> /*include 称为文件包含命令，扩展名为 h 的文件称为头文件*/
main()
{   printf("Hello, World !\n"); //显示 Hello, World !信息
}
```

说明：main 是主函数的函数名，表示这是一个主函数。每一个 C 语言程序都必须有且只能有一个主函数（main 函数）。调用 printf 函数的功能是把要输出的内容在显示器上显示。printf 函数是一个由系统定义的标准函数（即库函数），可在程序中直接调用。

【例 1-2】 输入数 x，计算 sin(x)的值。程序的每个语句功能由注释可知。

```
#include <math.h> /*程序要使用到数学库函数（sin 函数）*/
#include <stdio.h>
```

```

main()                                /*main 函数首部*/
{   double x,s;                       /*定义两个实数变量，以被后面程序使用*/
    printf("input number:\n");        /*显示提示信息*/
    scanf("%lf",&x);                 /*从键盘获得一个实数 x*/
    s=sin(x);                         /*求 x 的正弦，并把它赋给变量 s*/
    printf("sin(%lf) = %lf\n",x,s);    /*显示程序运算结果*/
}                                       /*main 函数体结束*/

```

说明：此段程序的功能是从键盘输入一个数 x ，求 x 的正弦值，然后输出结果。

在 `main()` 之前的两行称为预处理命令（详见后面）。预处理命令还有其他几种，这里的 `include` 称为文件包含命令，其意义是把尖括号 `<>` 或引号 `" "` 内指定的文件包含到本程序中，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为 `.h`，因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型，因此凡是在程序中调用一个库函数时，一般都要包含该函数原型所在的头文件。在本例中，使用了 3 个库函数：输入函数 `scanf`、正弦函数 `sin`、输出函数 `printf`。`sin` 函数是数学函数，其头文件为 `math.h` 文件，因此在程序的主函数前用 `include` 命令包含了 `math.h`。`scanf` 函数和 `printf` 函数是标准输入/输出函数，其头文件为 `stdio.h`，因此在主函数前也用 `include` 命令包含了 `stdio.h` 文件。

需要说明的是，C 语言规定对 `scanf` 和 `printf` 这两个函数可以省去对其头文件的包含命令。所以，在本例中也可以删去第二行的包含命令 `#include <stdio.h>`。

在例题中的主函数体中又分为两部分，一部分为说明部分（可选），另一部分为执行部分。说明部分包括对变量的类型定义或函数声明等。【例 1-1】中未使用任何变量，因此无说明部分。

C 语言规定，**源程序中所有用到的变量都必须先说明，后使用**，否则将会出错。这一点是编译型高级程序设计语言的一个共同特点，与解释型的 BASIC 语言不同。说明部分是 C 语言程序结构中很重要的组成部分。

【例 1-2】中使用了两个变量 x 与 s ，用来表示输入的自变量和 `sin` 函数值。由于 `sin` 函数要求这两个量必须是双精度浮点型，故用类型说明符 `double` 来说明这两个变量。

说明部分后的 4 行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一行是输出语句，调用 `printf` 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值。第二行为输入语句，调用 `scanf` 函数，接受键盘上输入的数并存入变量 x 中。第三行是调用 `sin` 函数并把函数值赋予变量 s 。第四行是用 `printf` 函数输出变量 s 的值，即 x 的正弦值。程序结束。

在前两个例子中用到了 C 语言输入函数（`scanf`）和输出函数（`printf`），这里简单介绍一下它们的格式，以便下面使用。

`scanf` 和 `printf` 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入、输出值。因此，这两个函数在括号中的参数表都由以下两部分组成：“**格式控制串**”，“**参数表**”。其中，“**格式控制串**”是一个字符串，必须用双引号括起来，它表示输入、输出量的数据类型。各种类型的格式表示法可参阅第 4 章。在 `printf` 函数中还可以在格式控制串内出现非格式控制字符，这时它们在显示屏幕上将原文照印。“**参数表**”中给出了输入或输出的量，当有多个量时，用逗号间隔。

例如：“`printf("sin(%lf) = %lf\n",x,s);`”，其中 `%lf` 为格式控制字符串，表示按双精度浮点数处理。它在格式串中出现两次，对应 x 和 s 两个变量。其余字符为非格式字符，则照原样

输出到屏幕上。

【例 1-3】 从文件 `infile.txt` 或键盘输入 `x`, `y` 两整数, 经处理后, 输出两数、两数中的最大值与最小值到屏幕和输出文件 `outfile.txt` 中。

注意: 一个 C 语言程序可由一个、两个或多个 C 源程序文件组成, 如下程序设计分成两个源程序文件 (若运行包含多文件的程序有困难时, 可以先作为一个文件运行)。

```
// file1.c——C 源程序文件 1, 含 main()与 max()函数
#include<stdio.h>
main()                                /*主函数*/
{  int x,y,z;                          /*变量说明*/
    FILE *fpi,*fpo;
    int max(int a,int b);              /*内部函数声明*/
    extern int min(int a,int b);       /*外部函数声明*/
    if((fpi=fopen("infile.txt","r"))==NULL) /*打开 infile.txt 输入文件*/
    {  printf("Input two numbers:\n");   /*若打开失败, 则提示从键盘输入*/
        scanf("%d%d",&x,&y);           /*从键盘输入 x,y 值*/
    }
    else                               /*若打开文件成功, 则从 infile.txt 文件读两数*/
    {  fscanf(fpi,"%d%d",&x,&y);         /*从文件输入 x,y 值*/
        fclose(fpi);                  /*关闭文件指针 fpi*/
    }
    z=max(x,y);                        /*调用 max 函数*/
    fpo=fopen("outfile.txt","w");       /*打开 outfile.txt 输出文件*/
    fprintf(fpo,"Two numbers are %d, %d\n",x,y); /*输出两数到文件*/
    fprintf(fpo,"Max number is %d\n",z);     /*输出最大值到文件*/
    fprintf(fpo,"Min number is %d\n", min(x,y)); /*输出最小值到文件*/
    fclose(fpo);                        /*关闭文件指针 fpo*/
    printf("Two numbers are %d, %d\n",x,y);  /*输出两数到显示屏*/
    printf("Max number is %d\n",z);          /*输出两者间最大值到显示屏*/
    printf("Min number is %d\n", min(x,y));  /*输出两者间最小值到显示屏*/
}
int max(int a,int b)                   /*定义 max 函数*/
{  if(a>b) return a;                  /*把结果按条件返回主调函数*/
    else return b;
}

/* file2.c——C 源程序文件 2, 含 min()函数 */
int min(int a,int b)                  /*定义 min 函数*/
{  int result;
    result= a<b?a:b;                  /*使用?:条件运算符得到最小数*/
    return result;                    /*返回最小值*/
}
```

说明: 本程序由 3 个函数组成, `main` 主函数、`max` 函数和 `min` 函数, 且函数之间是并列关系, 可从主函数中调用其他函数。`max` 函数的功能是比较两个数, 然后把较大的数返回给主函数, 而 `min` 函数把较小的数返回给主函数。`max` 函数和 `min` 函数是用户自定义函数。因此, 在主函数中要给出函数声明 (程序第 6、7 行), 其中 `min` 函数还来自另一个源文件, 为此是外部函数声明。可见, 在程序的说明部分中, 不仅可以有变量说明, 还可以有内部和外部函数的声明。本程序还涉及对文件的打开、输入、输出和关闭等操作。关于函数与