



21世纪高等学校计算机
基础实用规划教材

C程序设计基础 (第2版)

◎ 王珊珊 臧浏 张志航 皮德常 编著



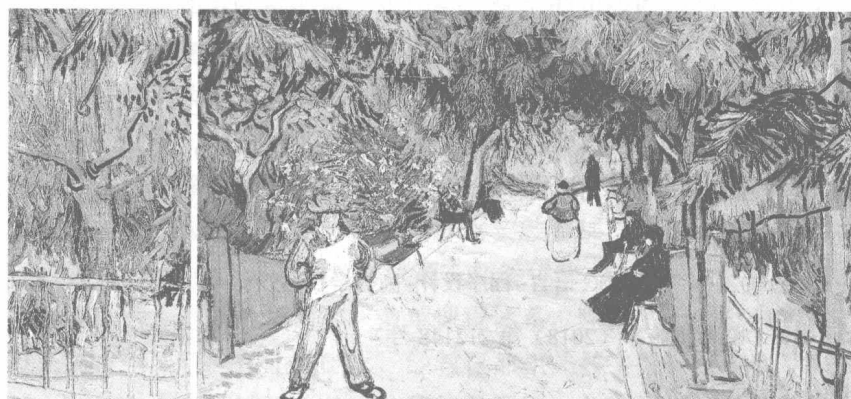
清华大学出版社



21世纪高等学校计算机
基础实用规划教材

C程序设计基础 (第2版)

◎ 王珊珊 臧浏 张志航 皮德常 编著



清华大学出版社

北京

内 容 简 介

本书全面介绍标准 C 语言的相关知识, 内容包括: C 语言概述, 基本数据类型、运算符和表达式, 标准设备的输入输出, C 语言的流程控制, 函数, 编译预处理, 数组, 结构体、共用体和枚举类型, 指针, 链表及其算法, 数据文件的使用等。

本书力求概念严谨, 同时做到深入浅出、通俗易懂; 通过大量的例题和习题帮助程序设计初学者掌握必需的基本语法和常用算法。

本书适合作为高等学校计算机专业和非计算机的理工科各专业的程序设计基础课程教材, 也可以作为广大计算机爱好者的自学教材。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C 程序设计基础/王珊珊等编著.—2 版.—北京: 清华大学出版社, 2019

(21 世纪高等学校计算机基础实用规划教材)

ISBN 978-7-302-51263-9

I. ①C… II. ①王… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2018) 第 212148 号

责任编辑: 付弘宇 张爱华

封面设计: 刘 键

责任校对: 时翠兰

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 三河市君旺印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 20 字 数: 486 千字

版 次: 2007 年 9 月第 1 版 2019 年 2 月第 2 版 印 次: 2019 年 2 月第 1 次印刷

印 数: 1~1500

定 价: 49.00 元

产品编号: 044412-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建设,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要;正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程可以有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家, 择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时, 要引入竞争机制, 通过申报、评审确定主题。书稿完成后要认真实行审稿程序, 确保出书质量。

繁荣教材出版事业, 提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度, 希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材

联系人: 魏江江 weijj@tup.tsinghua.edu.cn

前 言

C 语言是在由 UNIX 的研制者（美国贝尔实验室的 Dennis Ritchie 和 Ken Thompson）于 1970 年研制出的 B 语言的基础上发展和完善起来的。1972 年，在 DEC PDP-11 计算机上实现了最初的 C 语言，此后 C 语言伴随着计算机的发展一直走到了今天。

C 语言是一种支持过程化的、实用的程序设计语言，是高校学生学习程序设计的一门必修基础课程，同时也是编程人员广泛使用的工具。学好 C 语言，可以触类旁通其他语言，如 C++、Java、C# 和 VB 等。本书是编者在总结过去 20 多年的教学和实践经验的基础上编写而成的，适合作为高等学校计算机专业和非计算机的理工科专业的程序设计基础课程教材，也可供读者自学使用。本书目前被用作南京航空航天大学理工科各专业的程序设计语言课程教材。

本书作者主张的教学理念是注重程序设计的算法的教学，注重对学生算法思路的逻辑训练。本书讲述力求概念严谨，同时做到深入浅出、通俗易懂。各章节配有大量的例题和习题，主要是针对各章的教学难点和重点以及各种算法而设计的。在选择例题和习题时，尽量涵盖目前程序设计语言课程的各种算法类型，使初学者见到习题后，能够在教材的例题中找到相似的例子，这样对初学者来说，解题就不是一件非常困难的事情。建议教师在进行课本教学外，根据实际情况安排课程设计，选用适合不同层次学生的课程设计题目，强化训练学生动手编写较大规模程序的能力。

2007 年本书的第 1 版《程序设计语言——C》出版，至今已使用多年，在学生中取得了良好的反响。第 1 版的程序运行环境是 Turbo C 2.0，为了适应计算机软硬件环境的变化，本书的程序运行环境改为 Visual Studio 2010（简称 VS2010）。在教材内容上，本书在第 1 版的基础上，修正了部分不足，文字描述更加准确；本书对第 1 版全部源程序例子做了少量增删，并对源程序主函数的书写格式做了调整，调整前和调整后的格式如下：

```
main()                //调整前的程序书写格式，主函数没有明确指定返回值
{
    //...语句
}

int main()            //调整后的程序书写格式，主函数的返回值类型指定为 int
{
    //...语句
    return 0;         //调整后，增加 return 语句
}
```

本书主要具有以下几点特色。

(1) 整体考虑计算机和非计算机专业的教学要求,适用于计算机专业和非计算机的理工科各专业。

(2) 本书比较“瘦”,页码相对较少。笔者曾调研过一些高校,该课程的课时约为40~56。既然学时有限,那么书的厚度也应相应配套。

(3) 在内容顺序的安排上更加合理,方便计算机专业和非计算机专业的学生在学习时进行内容上的取舍。例如提前讲解结构体、枚举等内容,这为学生在后续的学习中使用这些内容进行实验做了铺垫。此部分内容在其他许多同类教材中都是最后讲解。又如,在介绍链表时分别讲解了不带头结点和带头结点的链表算法,满足不同专业的教学需要。

(4) 给出部分算法的来历和数学证明(如筛选法求素数以及汉诺塔问题),增加趣味性。

(5) 在作业安排上,从易到难,环环相扣。有许多学生学过C语言,却不会编程。编者在教学中认识到了这一点,因此设计了许多与实际有关的习题,并且这些习题都是彼此相关的。

(6) 本书通俗易懂,深入浅出,将复杂的概念采用浅显的语言讲述,便于读者理解和掌握。

本书第1~3章、第9~11章由王珊珊执笔(其中10.3.3节由皮德常执笔),第5~8章由臧渊执笔,第4章由张志航执笔。全书由王珊珊负责统稿。皮德常仔细通读了全书,在基本概念及文字描述上做了把关,并给出部分算法的来历和数学证明。参加本书编写工作的还有朱敏、张定会、钱忠民、潘梅园、张卓莹、刘佳、朱玉莲等。

讲述本书全部内容的建议学时为:理论教学48学时,课程设计16~32学时(内容另行安排),上机实验50学时。本书的实验环境是Visual Studio 2010,书中全部例题和习题均已在该环境中通过编译和运行。书中标题前加“*”的章节为选学内容。

本书提供所有例题的源代码和习题的答案,同时向选用本书作教材的教师提供讲课用的PowerPoint格式电子教案。读者可以直接从清华大学出版社网站(<http://www.tup.com.cn>)下载这些资源。资源下载的相关问题请联系 fuhy@tup.tsinghua.edu.cn。

与本书配套的实践教材《C语言程序设计上机实验及学习指导》已由南京大学出版社出版。该实践教材的主要内容是与本书配套的上机实验、课程设计以及针对每章知识点的练习题及解析,是本书的补充。两本教材形成了完整的教材体系,是保障教学质量的基础。

由于编者的水平有限,本书中难免存在疏漏之处,恳请专家和广大读者指正。可以通过电子邮件与编者联系: shshwang@nuaa.edu.cn(王珊珊), zangliwen@nuaa.edu.cn(臧渊), zzh20100118@qq.com(张志航), dc.pi@nuaa.edu.cn(皮德常)。

编者

2018年6月

于南京航空航天大学

目 录

第 1 章 C 语言概述	1
1.1 计算机语言与程序	1
1.1.1 机器语言与程序	1
1.1.2 汇编语言与程序	1
1.1.3 高级语言与程序	2
1.2 C 语言的发展及其特点	3
1.3 简单的 C 程序	4
1.4 程序开发的步骤	6
习题 1	7
第 2 章 数据类型、运算符和表达式	8
2.1 保留字和标识符	8
2.1.1 保留字	8
2.1.2 标识符	8
2.2 C 语言的基本数据类型	9
2.3 常量和变量	10
2.3.1 常量	10
2.3.2 符号常量	13
2.3.3 变量	14
2.4 基本运算符和表达式	15
2.4.1 C 语言运算符及表达式简介	15
2.4.2 算术运算符和算术表达式	16
2.4.3 运算符的优先级和结合性	16
2.4.4 关系运算符和关系表达式	16
2.4.5 逻辑运算符和逻辑表达式	17
2.4.6 位运算符和位运算表达式	18
2.4.7 自增、自减运算符和表达式	21
2.4.8 赋值运算符和赋值表达式	21
2.4.9 逗号运算符和逗号表达式	22
2.4.10 sizeof()运算符和表达式	23

2.4.11 逻辑表达式运算优化的副作用	23
2.5 类型转换	23
2.5.1 赋值时的自动类型转换	23
2.5.2 各种类型运算量混合运算时的自动类型转换	26
2.5.3 强制类型转换	26
习题 2	27
第 3 章 标准设备的输入输出	29
3.1 输入输出的基本概念	29
3.2 格式化输入输出函数的使用	29
3.2.1 格式化输出函数 printf()	29
3.2.2 格式化输入函数 scanf()	35
3.3 字符输入输出函数的使用	40
3.4 顺序结构程序设计举例	41
习题 3	43
第 4 章 C 语言的流程控制	47
4.1 算法概述	47
4.1.1 算法及其效率	47
4.1.2 算法的设计原则	48
4.1.3 算法的表示工具	49
4.1.4 结构化程序设计中基本结构的表示	51
4.2 C 程序的结构和语句概述	53
4.3 选择结构语句	55
4.3.1 if 语句	55
4.3.2 if 语句的嵌套使用	59
4.3.3 条件运算符	61
4.3.4 switch 语句	62
4.4 循环结构语句	67
4.4.1 goto 语句及标号的使用	67
4.4.2 while 语句	68
4.4.3 for 语句	69
4.4.4 do-while 语句	71
4.4.5 break 语句和 continue 语句	72
4.4.6 循环的嵌套	74
4.5 控制语句应用举例	75
习题 4	84

第 5 章 函数	86
5.1 函数概述	86
5.2 函数的定义与调用	87
5.2.1 函数的定义	87
5.2.2 函数的调用	89
5.2.3 函数的参数	91
5.3 函数的原型声明	92
5.4 函数的嵌套调用和递归调用	94
5.4.1 函数的嵌套调用	94
5.4.2 函数的递归调用	99
5.5 使用 C 系统函数	103
5.6 作用域和存储类别	106
5.6.1 作用域	106
5.6.2 存储类别	109
5.6.3 全局变量的作用域的扩展和限制	113
5.7 程序的多文件组织	116
5.7.1 内部函数和外部函数	116
5.7.2 多文件组织的编译和连接	118
习题 5	118
第 6 章 编译预处理	121
6.1 宏定义	121
6.1.1 不带参数的宏定义	121
6.1.2 带参数的宏定义	123
6.2 文件包含	126
*6.3 条件编译	127
习题 6	130
第 7 章 数组	132
7.1 数组的定义及应用	132
7.1.1 一维数组的定义及使用	132
7.1.2 一维数组作函数参数	136
7.1.3 多维数组的定义及使用	148
7.1.4 二维数组作函数参数	151
7.2 字符数组的定义及应用	157
7.2.1 字符数组的定义	158

7.2.2	字符数组的初始化	158
7.2.3	字符数组的使用	159
7.2.4	字符串和字符串结束标志	160
7.2.5	字符数组的输入输出	161
7.2.6	字符串处理函数	163
7.2.7	字符数组应用举例	167
习题 7		170
第 8 章	结构体、共用体和枚举类型	172
8.1	结构体的定义及应用	172
8.1.1	结构体类型的定义	172
8.1.2	结构体类型变量的定义	173
8.1.3	结构体类型变量及其成员的引用	175
8.1.4	结构体数组	178
8.2	共用体的定义及应用	181
8.2.1	共用体类型及其变量的定义	181
8.2.2	共用体类型变量的引用	182
8.2.3	共用体类型数据的特点	183
8.3	枚举类型	185
8.3.1	枚举类型的定义	185
8.3.2	枚举类型变量的定义	185
8.3.3	枚举类型变量的使用	185
8.4	用 typedef 定义类型	189
习题 8		191
第 9 章	指针	193
9.1	指针和指针变量	193
9.1.1	指针的概念	193
9.1.2	指针变量的定义	193
9.1.3	与指针有关的运算符&和*	194
9.1.4	指针变量赋值	194
9.1.5	直接访问和间接访问	194
9.1.6	地址值的输出	198
9.2	指针作函数参数	198
9.2.1	基本类型量作函数参数	199
9.2.2	指针变量作函数参数	200
9.3	指针和指向数组的指针	202

9.3.1	指针和一维数组	202
9.3.2	一维数组元素指针作函数参数	206
9.3.3	指针和字符串	211
9.3.4	二维数组与指针	216
9.3.5	获得函数处理结果的几种方法	222
9.4	指针数组	224
9.4.1	指针数组的定义和使用	224
9.4.2	使用指针数组处理二维数组	225
9.4.3	利用字符指针数组处理字符串	227
9.4.4	main()函数的参数	229
9.5	指向指针的指针	231
9.6	指针和函数	232
9.6.1	函数指针	232
9.6.2	返回指针的函数(指针函数)	236
9.7	指针小结	237
习题 9		240
第 10 章	链表及其算法	243
10.1	存储空间的动态分配和释放	243
10.2	结构体及指针	245
10.3	链表及算法	247
10.3.1	链表概念的引入	247
10.3.2	不带头结点的链表的常用算法	249
*10.3.3	带头结点的链表的常用算法	257
习题 10		263
第 11 章	数据文件的使用	265
11.1	输入输出概述	265
11.2	文件的基本概念和分类	265
11.3	缓冲的概念	266
11.4	文件的读写过程	267
11.4.1	文件和数据流	267
11.4.2	文件类型指针	267
11.4.3	文件的打开与关闭	268
11.5	文件的读写	270
11.5.1	fgetc 函数、fputc 函数和 feof 函数	271
11.5.2	fgets 函数和 fputs 函数	274

11.5.3	fscanf 函数和 fprintf 函数	279
11.5.4	fread 函数和 fwrite 函数	281
11.5.5	getw 函数和 putw 函数	284
11.6	文件的随机读写	285
11.6.1	文件读写位置指针的定位	285
11.6.2	文件的随机读写	287
11.7	文件的出错检测	290
*11.8	输入输出重定向	291
*11.9	通过命令行参数指定待读写的文件名	293
	习题 11	294
附录 A	ASCII 码表	297
附录 B	常用库函数	298
	参考文献	304

1.1 计算机语言与程序

人类语言是人与人之间交流信息的工具，而计算机语言是人与计算机之间交流信息的工具。用计算机解决问题时，人们必须首先将解决该问题的方法和步骤按一定序列和规则用计算机语言描述出来，形成计算机源程序，经过编译、连接，生成可执行程序，然后计算机就可自动执行该程序，完成所需要的功能。

计算机语言与程序经历了以下三个阶段的发展。

1.1.1 机器语言与程序

机器语言是第一代计算机语言。

我们知道，在计算机内部采用二进制表示信息。指挥计算机完成一个基本操作的指令也是由二进制代码构成的，称为机器指令。每一条机器指令的格式和含义都是由设计者规定的，并按照这个规定设计和制造硬件。一个计算机系统的全部机器指令的总和称为指令系统，也就是机器语言。用机器语言编写的程序为如下形式：

```
0000 0100 0001 0010
0000 0100 1100 1010
0001 0010 1111 0000
1000 1010 0110 0001
    :
```

上面每一行都是一条机器指令，代表一个具体的操作。用机器语言编写的程序能直接在计算机上运行，运行速度快、效率高，但必须由专业人员编写程序。机器语言程序依赖于硬件，程序的可移植性差。所谓移植，是指在一个计算机系统下编写的程序经过改动可以在另一个计算机系统中运行，并且运行结果一致。改动越少，表示可移植性越好；反之，表示可移植性越差。

1.1.2 汇编语言与程序

汇编语言是第二代计算机语言。

汇编语言是一种符号语言，它将难以记忆的二进制指令用有意义的英文单词缩写来代替，英文单词缩写被称为助记符，每一个助记符代表一条机器指令。例如，用助记符 ADD 表示“加”操作，用 SUB 表示“减”操作。用汇编语言编写的程序为如下形式：

```
MOV AL 12D          /* 表示将十进制数 12 送往累加器 AL */
```

```

SUB AL 18D      /* 表示从累加器 AL 中减去十进制数 18 */
:
HLT            /* 表示停止执行程序 */

```

汇编语言提高了程序的可读性和可写性，使编程者在编写程序时稍微轻松了一点。但汇编语言程序不能在计算机上直接运行，必须把它翻译成相应的机器语言程序才能运行。将汇编语言程序翻译成机器语言程序的过程称为汇编。汇编过程是计算机运行汇编程序自动完成的，如图 1-1 所示。汇编程序是一种计算机系统软件。

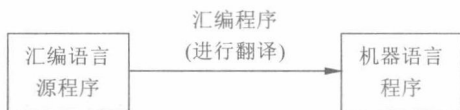


图 1-1 汇编过程

1.1.3 高级语言与程序

如上所述，机器语言和汇编语言都是面向机器的语言，受计算机硬件机型的限制，通用性差，不便于学习，一般只适用于专业人员。为了从根本上解决这个问题，人们创造了高级程序设计语言，简称高级语言。高级语言采用类似于人类自然语言和数学语言的方式描述问题、编写程序。例如，用 C 语言编写的一个程序片段如下：

```

int a, b, c;           /* 定义变量 a, b, c */
scanf("%d%d", &a, &b); /* 输入变量 a, b 的值 */
c = a + b;            /* 将变量 a, b 的值相加，结果赋给变量 c */
printf("c=%d\n", c); /* 输出求和结果 */

```

该程序片段每条语句的功能已给出说明。用高级语言编写程序时，不需要了解计算机的内部硬件结构，只要告诉计算机“做什么”即可。至于计算机用什么机器指令去完成（即“怎么做”），编程者不需要关心。也就是说，编写高级语言程序时，不需要考虑具体的计算机硬件系统。

但是，计算机无法直接执行高级语言程序，必须将高级语言程序翻译成机器语言程序才能执行。翻译过程分成两步完成，即编译和连接，如图 1-2 所示。



图 1-2 编译、连接过程

高级语言源程序经编译后得到目标程序（一种形式的机器语言程序），再经过与库程序（包含系统提供的通用函数目标代码）的连接生成可执行的机器语言程序。编译程序和连接程序分别完成编译和连接工作，它们也是计算机系统软件。

高级语言是第三代计算机语言。高级语言不仅易学易用，通用性强，而且具有良好的

可移植性。因为不同的计算机系统有不同的编译程序（也称编译器），将高级语言源程序重新编译（在编译之前有时需要对源程序稍加改动，称为移植）后，即可在不同的计算机系统中运行。

目前世界上有数百种高级语言，应用于不同领域，而 C 语言作为其中的优秀代表，得到了普遍应用。

1.2 C 语言的发展及其特点

C 语言与 UNIX 操作系统关系密切。最初的 UNIX 是采用汇编语言编写的，而后出现了 B 语言版本的 UNIX（它是第一个用高级语言编写的 UNIX）。C 语言是美国贝尔实验室的 Dennis M. Ritchie 在 B 语言版的 UNIX 基础上开发出来的，1972 年在 DEC PDP-11 计算机上实现了最初的 C 语言。当时开发 C 语言的目的就是为了编写新版 UNIX 操作系统，新版 UNIX 操作系统中 90% 的代码用 C 语言编写，10% 的代码用汇编语言编写。随着 UNIX 操作系统的广泛应用，C 语言也被人们认识和接受。

20 世纪 80 年代前后，C 语言在各种计算机上的快速推广产生了许多 C 语言版本。这些版本虽然是类似的，但通常互不兼容。显然人们需要一个与开发平台和机器无关的标准 C 语言版本。从 1983 年开始，美国国家标准协会（American National Standard Institute, ANSI）着手制定 C 语言的规范，到 1989 年完成了 ANSI C 标准的制定。Brian W. Kernighan 和 Dennis M. Ritchie 编著的 *The C Programming Language—Second Edition*（1988 年，第 2 版）介绍了 ANSI C 的全部内容，该书被称为 C 语言的圣经（C Bible）。该书的第 1 版在 1978 年完成，有兴趣的读者可参考阅读。

目前国际标准化组织（ISO）推出的最新的、成熟的 C 语言标准有 C99 和 C11 等，因为本书是 C 语言的入门教材，讲授的是 C 语言的基本知识，所以本书基本上遵循 ANSI C 标准。有兴趣的读者可自行参阅相关的 C 语言新标准。

C 语言具有如下优点。

（1）语言简洁、紧凑，使用方便、灵活。C 语言只有 32 个关键字，程序书写形式自由。

（2）具有丰富的运算符和数据类型。

（3）C 语言可以直接访问内存地址，进行位操作，完成类似于汇编语言的操作，使其能够胜任开发系统软件的工作。有时 C 语言也被称为中级语言，其含义是它将高级语言的优点与汇编语言的实用性结合在一起。

（4）生成的目标代码质量高，程序运行效率高。

（5）可移植性好。即程序可以很容易地被改写并运行在不同的计算机上。

但是，C 语言也具有如下局限性。

（1）C 语言的数据类型检查机制较弱，这使得程序中的一些错误不能在编译时被发现。

（2）C 语言本身几乎没有支持代码重用的语言结构，因此一个程序员精心设计的程序很难为其他程序所用。

（3）当程序达到一定规模时，程序员很难控制程序的复杂性。

C语言既适用于开发大型的复杂系统软件,也可以开发一般的应用软件。

C语言的发展方向是C++语言。在C语言的基础之上,为了满足管理程序的复杂性以及代码重用的需要,1980年贝尔实验室的Bjarne Stroustrup博士及其同事对C语言进行了改进和扩充,最初的成果称为“带类的C”,而后称为“新C”。1983年,由Rick Mascitti提议,正式命名为C++(C Plus Plus)。因为在C语言中,运算符“++”是对变量进行增值运算,那么C++的喻义是对C语言进行“增值”。1994年制定了ANSI C++草案。此后又经过不断完善,成为目前的C++。C++语言仍然在不断的发展中。有兴趣的读者可以在学完本书内容后,继续学习C++语言。

1.3 简单的C程序

下面举一个简单的例子来说明C程序的基本结构。

例 1-1 第一个简单的C程序,源程序名为Li0101.c。

```
/* -----  
    程序名: Li0101.c  
    程序功能: 输出一行信息  
    开发者: XXX  
    开发时间: 2015年2月10日  
    修改时间: 2017年6月16日  
    其他说明: ...  
-----  
*/  
  
#include<stdio.h>  
  
int main()  

```

上述程序由注释语句和主函数构成。

C程序的注释语句从“/*”开始,到“*/”结束,在“/*”和“*/”之间对程序的功能、语句的功能以及一些程序的设计思想做说明。注释分成两种:一种是块状的序言性注释;另一种是书写在一行中的功能性注释。本例中的注释是块状注释,一般写在程序的开头部分,跨多行书写,内容包括源程序名称、总体功能、程序开发者、开发日期、修改日期等。一般在编写大型正式程序时给出序言性注释,在学习C程序设计基础时忽略该注释。在下面的例1-2中可以看到在一行中书写的功能性注释,一般写在一行的后半部分,对本行语句做说明。注释语句是给程序开发者看的,对产生程序的可执行代码没有影响。C语言对源程序正式编译之前,会将注释语句从源程序中删除,参见第11章中例11-6。

任何C程序必须包含一个主函数,函数名是main,它是主控函数,表示程序从该函数开始执行。例1-1中,int main()是函数的首部,紧接着是由花括号{}括起来的函数体。在