



北京市高等教育精品教材立项项目

高等学校21世纪教材

GAODENG XUEXIAO 21 SHIJI JIAOCAI

# C语言 程序设计

© 孟庆昌 陈海鹏 马鸣远 刘振英 编著



人民邮电出版社  
POSTS & TELECOM PRESS



北京市高等教育精品教材立项项目

# C 语言程序设计

孟庆昌 陈海鹏 马鸣远 刘振英 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

C 语言程序设计 / 孟庆昌等编著. —北京: 人民邮电出版社, 2006.4

ISBN 7-115-14482-6

I. C... II. 孟... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 003738 号

## 内 容 提 要

本书全面、系统、循序渐进地介绍了 C 语言的基本概念、各种语法成分及其在程序设计中的应用, 并通过大量实例程序讲述 C 语言应用中的重点和难点, 引导读者掌握一般程序设计方法。每章后面有习题, 可供读者练习。

本书是北京市高等教育精品教材立项项目。全书概念清晰、结构合理、内容严谨、讲解透彻、重点突出、示例典型、实用性强, 既考虑到初学者的特点, 又能满足软件设计人员的工作需要。本书既可以作为高等院校学生以及计算机培训班学员学习 C 语言的教材, 也适合广大软件开发人员和自学人员参考阅读。

北京市高等教育精品教材立项项目

### C 语言程序设计

- 
- ◆ 编 著 孟庆昌 陈海鹏 马鸣远 刘振英  
责任编辑 潘春燕
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京隆昌伟业印刷有限公司印刷  
新华书店总店北京发行所经销
  - ◆ 开本: 800×1000 1/16  
印张: 24.75  
字数: 547 千字 2006 年 4 月第 1 版  
印数: 1—3 000 册 2006 年 4 月北京第 1 次印刷

---

ISBN 7-115-14482-6/TP · 5211

定价: 32.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

## 丛书前言

当今世界，科学技术突飞猛进，知识经济已见端倪，国际竞争日趋激烈。教育在综合国力的形成中处于基础地位，国力的强弱将越来越取决于劳动者的素质，取决于各类人才的质量和数量，这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初，我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体，是进行教学的基本工具，也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此，高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代，计算机教学内容也随之更新，尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施，对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授，精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略，组织编写好这套教材，我们在国内高校做了系统、详细的调查，对教育部制订的教育计划做了认真的研究，还对国内外已出版的教材做了理性的分析，确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中，我们多次组织了由专家及高校一线教师参加的研讨会，对现有比较出色的教材的特点及优点进行了分析，博采众长，力求实现教材权威性与实用性的完美结合。

本套教材有如下特点：

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况，在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。
2. 紧密结合当前技术的新发展，在阐述理论知识的同时侧重实用性。
3. 力求在概念和原理的讲述上严格、准确、精练，理论适中，实例丰富，写作风格上深入浅出，图文并茂，便于学生学习。
4. 为适应当前高校课程种类多、课时数要压缩的教学特点，教材不仅篇幅有很大的压缩，而且均配有电子教案，以满足现代教学新特点的需要，做到易教易学。
5. 所选作者均是国内有丰富教学实践经验的知名专家、教授，所编教材具有较高的权威性。

教育的改革将不会停止，教材也将会不断推陈出新。目前本套教材即将推出，将接受广大教学第一线教师的检验。

由于我们的水平和经验有限，这批教材在编审、出版工作中还存在不少缺点和不足，希望使用本套教材的学校师生和广大读者提出批评和建议，以便改进我们的工作，使教材质量不断提高。

## 编者的话

本书是北京市教育委员会评审并确认的北京市高等教育精品教材立项项目。本书的出版是市教委指导和支持的结果，是众多专家和同仁大力帮助的结果，是与人民邮电出版社多年合作的结果，是作者辛勤劳作的结晶。

作者从 1981 年开始率先在高校讲授 C 语言。根据多年教学和科研的经验与体会，并汲取国内外有关程序设计语言方面优秀教材的精华，结合我国国情，本着提高学生素质、培养创新意识的精神，遵循“学以致用”的原则，力求编写一本受欢迎的好教材。

C 语言从 1972 年问世，至今已历经 30 多个年头。在此期间，信息技术得到迅猛发展，诞生了众多优秀的程序设计语言。然而，C 语言仍是软件百花园中的奇葩，在系统开发、软件工程、软件测试等领域独领风骚。特别在程序设计、数据结构等教学方面已成为事实上的“标准”语言。C 语言能保持经久不衰，是与其一系列突出的优点分不开的，它简洁、高效、灵活、可移植性好、应用面广，是第三代语言的杰出代表。

本书是在原《C 语言程序设计》的基础上，经认真修改而成，对原有内容做了精简，也补充了编程知识，适当压缩篇幅，使之更适合于教学。编写时注意了以下几方面问题：

1. 概念严谨，理论系统，又不拘泥于语法讲解。对于初学者来说，建立正确的概念至关重要。本书对所讲解的对象都力求予以严谨的表述，使读者明了相关事物的实质，循序渐进地掌握 C 语言及其程序设计的知识。通过大量实例应用讲解 C 语言的重点、难点，从中体会相关知识“是什么、干什么、如何用”。

2. 注重能力提高。书中有一章集中介绍程序设计方面的知识，使读者掌握程序设计和调试的一般方法和技巧。同时，书中给出大量示例，兼顾一般知识介绍和实际应用，重点在于提高读者分析问题、确定算法、编制代码和运行调试的能力。

3. 实例丰富，难易得当。有些示例着重介绍 C 语言的某些语法现象，以加深对概念的理解；有些示例具有应用价值，以后在实际软件开发工作中也可以使用。在例题的难度上多数比较容易，适合初学者自学。也有些例题有一定的综合性，读者通过分析和运行它们，可以对 C 语言的应用有更深入地整体性了解。

4. 本书介绍的 C 语言知识遵循 ISO C 的标准，具有通用性。当然，不同的 C 语言版本会存在一些小差别。因此，在示例的选择上尽量避开对硬件环境和软件版本的依赖。但仍然可能会有个别示例的运行结果（如地址值）与读者上机运行时的结果不一样，请注意。

由于各学校培养目标、课程设置及学时安排等方面存在差异，所以在采用本教材授课时，可以对内容酌情进行取舍。可以讲授全部内容，也可对各章内容重点讲解，突出基本内容，对于较深入的算法和具体实现可以略讲或不讲，由学生自学。本书配套教材中给出了上机指

导、知识重点和各章习题的参考答案，其出发点是为了便于自学自测，提高教学效果。为便于广大教师授课，本书将提供配套电子教案。读者可通过登录人民邮电出版社网站 <http://www.ptpress.com.cn> 下载。

本书主要由孟庆昌、陈海鹏、马鸣远、刘振英编写，参加编写、整理、录入工作的人员还有孟欣、张震、刘园芳、孟平、唐伟杰等。在此对所有关心、支持本书出版的同志表示衷心感谢。

由于作者水平有限，时间仓促，对社会需求方面的了解不全面，因而书中难免有不妥甚至错误之处，恳请广大读者批评指正。

孟庆昌  
北京信息工程学院  
2005.12

# 目 录

<b>第 1 章 C 语言概述</b> .....	1
1.1 C 语言的发展历史和特点.....	1
1.1.1 C 语言的发展历史.....	1
1.1.2 C 语言的特性.....	2
1.2 C 程序示例.....	4
1.3 C 程序的编辑、编译和运行.....	9
1.3.1 一般上机解题步骤.....	9
1.3.2 在 TC 环境下运行 C 程序的步骤.....	10
1.3.3 在 UNIX/Linux 环境下运行 C 程序的步骤.....	16
习题.....	18
<b>第 2 章 基本数据类型</b> .....	20
2.1 字符集及词法约定.....	20
2.1.1 字符集.....	20
2.1.2 词法约定.....	21
2.2 常量.....	23
2.2.1 整型常量.....	24
2.2.2 浮点常量.....	26
2.2.3 字符常量.....	26
2.2.4 字符串常量.....	27
2.3 简单变量.....	28
2.4 基本数据类型及其转换.....	29
2.4.1 整型 int 及其相关类型.....	30
2.4.2 字符型 char 及其相关类型.....	31
2.4.3 浮点型 float.....	32
2.4.4 类型转换.....	33
2.5 printf 和 scanf 函数的一般使用.....	35
习题.....	38
<b>第 3 章 运算符和表达式</b> .....	40
3.1 表达式.....	40
3.2 运算符及表达式.....	40

3.2.1	算术运算符和算术表达式	41
3.2.2	赋值运算符和赋值表达式	44
3.2.3	增量运算符和增量表达式	45
3.2.4	关系运算符和关系表达式	48
3.2.5	条件运算符和条件表达式	50
3.2.6	逗号运算符和逗号表达式	51
3.2.7	逻辑运算符和逻辑表达式	52
3.2.8	位逻辑运算符和位逻辑表达式	55
3.2.9	移位运算符和移位表达式	59
3.2.10	其他运算符	60
3.3	运算符优先级和结合性	62
3.3.1	运算符汇总	62
3.3.2	运算符嵌套	64
3.3.3	表达式计算顺序	64
	习题	66
<b>第 4 章</b>	<b>程序设计知识</b>	<b>70</b>
4.1	程序的概念	70
4.1.1	程序的特性	70
4.1.2	程序设计语言	71
4.2	算法	72
4.2.1	算法的特点	72
4.2.2	算法的表示	73
4.3	结构化程序设计方法	76
4.3.1	结构化程序设计概念	76
4.3.2	结构化程序设计的三种基本结构	76
4.4	程序设计中的几种常用算法	81
4.4.1	穷举法	81
4.4.2	迭代法	83
4.4.3	递推法	86
4.4.4	递归法	86
4.5	程序调试	88
4.5.1	人工查错	88
4.5.2	程序调试	89
	习题	91
<b>第 5 章</b>	<b>语句和控制流</b>	<b>93</b>



5.1 表达式语句 .....	94
5.2 空语句 .....	94
5.3 返回语句 .....	95
5.4 复合语句 .....	96
5.5 if 语句 .....	97
5.5.1 if 语句的形式 .....	97
5.5.2 if 语句的嵌套形式 .....	99
5.6 switch 语句 .....	104
5.7 while 语句 .....	107
5.8 for 语句 .....	108
5.9 do-while 语句 .....	112
5.10 break 语句 .....	114
5.11 continue 语句 .....	116
5.12 goto 语句 .....	118
5.13 循环嵌套 .....	120
习题 .....	123
<b>第 6 章 数组 .....</b>	<b>126</b>
6.1 一维数组的定义和引用 .....	126
6.1.1 一维数组的定义 .....	126
6.1.2 一维数组元素的引用 .....	127
6.1.3 一维数组的初始化 .....	130
6.2 字符数组 .....	133
6.2.1 字符数组的定义和引用 .....	133
6.2.2 字符数组的初始化 .....	134
6.3 多维数组 .....	136
6.3.1 二维数组的定义 .....	136
6.3.2 二维数组的内部表示 .....	137
6.3.3 多维数组的定义 .....	137
6.3.4 二维数组引用 .....	138
6.3.5 二维数组的初始化 .....	140
6.4 应用举例 .....	145
习题 .....	151
<b>第 7 章 函数和变量存储类 .....</b>	<b>154</b>
7.1 函数定义 .....	154
7.2 main() 函数 .....	157

7.3	函数调用	159
7.3.1	函数调用的一般形式	160
7.3.2	函数调用的方式	161
7.3.3	函数参数	164
7.4	函数返回值和函数类型说明	167
7.4.1	函数返回值	167
7.4.2	函数类型说明	169
7.5	函数的递归调用	172
7.6	void 类型和可变参数函数	175
7.6.1	void 类型	175
7.6.2	可变参数函数	177
7.7	变量存储类	177
7.7.1	自动变量	178
7.7.2	寄存器变量	180
7.7.3	外部变量	181
7.7.4	静态变量	184
7.7.5	变量存储类汇总表	188
7.8	应用举例	188
	习题	193
<b>第 8 章</b>	<b>指针</b>	<b>197</b>
8.1	什么是指针	197
8.2	指针变量的定义	199
8.3	指针变量的引用	200
8.3.1	&运算符	200
8.3.2	*运算符	201
8.4	指针变量的运算	202
8.5	指针变量和数组	209
8.5.1	数组的指针和数组元素的指针变量	209
8.5.2	利用指针变量引用数组元素	211
8.6	指针作为函数参数	213
8.7	指向字符串的指针变量	216
8.7.1	实现字符串处理的方式	216
8.7.2	字符数组与字符指针变量的对比	219
8.8	指向多维数组的指针和指针变量	220
8.8.1	多维数组的地址表示	220

8.8.2 指向基本数组元素的指针变量	223
8.8.3 指向行数组的指针变量	224
8.9 指针数组	229
8.10 指向指针的指针	231
8.11 指向函数的指针变量	234
8.11.1 定义指向函数的指针变量	234
8.11.2 指向函数的指针变量的初始化和使用	235
8.11.3 指向函数的指针变量作为函数参数	236
8.12 返回指针的函数	238
8.13 指向 void 量的指针变量	241
8.14 动态存储分配	242
8.15 命令行参数	245
8.16 应用举例	247
习题	255
<b>第 9 章 结构与联合</b>	<b>258</b>
9.1 结构类型及其变量的定义	258
9.1.1 结构类型的定义	258
9.1.2 结构变量的定义	260
9.1.3 结构变量的内部表示	261
9.2 结构成员的引用	262
9.2.1 引用结构成员的一般方式	262
9.2.2 嵌套结构中成员的引用	264
9.3 结构变量的初始化	265
9.3.1 结构变量的一般初始化方式	265
9.3.2 有聚合成员的结构变量的初始化	266
9.4 结构数组	268
9.4.1 结构数组的定义及其应用	268
9.4.2 结构数组在内存中的表示	270
9.4.3 对结构数组的操作	270
9.5 结构和指针	272
9.5.1 指针变量作为结构成员	272
9.5.2 指向结构的指针	274
9.5.3 指向结构数组的指针	277
9.5.4 结构作为函数调用的参数	278
9.6 引用自身的结构和链表	281

9.6.1	引用自身的结构	281
9.6.2	链表	282
9.7	联合	288
9.7.1	联合变量的定义	288
9.7.2	联合变量的引用	290
9.8	应用举例	294
习题		300
<b>第 10 章</b>	<b>其他数据类型</b>	<b>303</b>
10.1	枚举	303
10.1.1	枚举类型和枚举变量的定义	303
10.1.2	枚举变量的取值	304
10.2	位段	308
10.2.1	字位标志法	308
10.2.2	位段的定义及其引用	309
10.3	用 typedef 定义类型别名	314
习题		317
<b>第 11 章</b>	<b>预处理、输入/输出和文件操作</b>	<b>318</b>
11.1	预处理功能	318
11.1.1	宏定义和宏替换	318
11.1.2	文件包含	328
11.1.3	条件编译	331
11.1.4	其他预处理功能	334
11.2	库函数使用方式	335
11.3	常用标准输入/输出函数	336
11.3.1	getchar()和 putchar()	336
11.3.2	gets()和 puts()	337
11.3.3	printf()和 scanf()	339
11.4	文件及有关操作	343
11.4.1	数据流和文件的概念	343
11.4.2	文件的打开与关闭	345
11.4.3	文件的读写	348
11.4.4	文件定位和出错检测	352
11.5	其他一些常用的函数(宏)	355
习题		361
<b>附录 A</b>	<b>C 语言语法汇总</b>	<b>363</b>

---

A.1 词法	363
A.1.1 单词	363
A.1.2 关键字	363
A.1.3 标识符	364
A.1.4 常量	364
A.1.5 字符串字面量	366
A.1.6 运算符	367
A.1.7 标点符号	367
A.1.8 前导文件名	367
A.1.9 预处理数	367
A.2 短语结构语法	368
A.2.1 表达式	368
A.2.2 说明	370
A.2.3 语句	373
A.2.4 外部定义	374
A.3 预处理指示	375
<b>附录 B 常用库函数</b>	<b>377</b>
<b>参考文献</b>	<b>380</b>

# 第 1 章 C 语言概述

C 程序设计语言（简称 C 语言）是最著名的高级程序设计语言之一，也是使用范围最广的一种计算机编程语言。它不仅可以用来编写如操作系统、数据库之类的系统软件，而且大量用于编写各种应用软件。运行 C 语言的平台从廉价个人机到功能强大的巨型机，从 DOS 系统到 Windows 2000/XP、UNIX/Linux 系统，几乎遍及所有的计算机平台。这一切都源于 C 语言的优良性质。

本章简要回顾 C 语言的发展历史及其特点，并通过示例，讲解 C 语言的一般知识及上机实习过程。

## 1.1 C 语言的发展历史和特点

### 1.1.1 C 语言的发展历史

说到 C 语言的发展历史就不得不提及 UNIX 操作系统的由来。从 1969 年开始，美国贝尔实验室的 K.Thompson（汤普森）和 D.M.Ritchie（里奇）为了在实验室内建造一种进行程序设计研究和开发的良好工作环境，着手研制、开发一种具有良好交互性能的操作系统——UNIX 操作系统。该系统最早运行在 DEC 的 PDP—7 机器上，用汇编语言编写。由于用汇编语言编写的程序不可移植，其描述问题的效率大大低于高级语言，而且可读性又差，所以 K.Thompson 在 1970 年开发出一种高级程序设计语言 B，于 1971 年在 PDP—11/20 上实现了 B 语言，并用 B 语言编写了 UNIX 操作系统和绝大多数的实用程序。B 语言的主要思想来源于 M.Richard（理查德）提出的 BCPL 语言（英国剑桥大学，1967 年）。由于 B 语言是面向字存取，而且存在功能过于简单、数据无类型、运行速度较慢等弱点，所以它并未得到广泛流行。1972 年 D.M.Ritchie 在 B 语言的基础上开发出 C 语言。C 语言克服了 B 语言的缺点，又保持了它的精炼、接近硬件的优点，同时扩充了很多适于系统设计和应用开发的功能。1973 年 K.Thompson 和 D.M.Ritchie 把 UNIX 系统用 C 语言重写了一遍，并加入多道程序设计等新的功能，这就是 UNIX 的第五版。在初期阶段，C 语言还局限在贝尔实验室内部使用。到 1975 年，随着 UNIX 第六版的公布并且免费推向世界，C 语言的一系列优点才引起人们的广泛关注。伴随 UNIX 系统从研究室、院校走向商业社会，登上世界舞台，

C 语言也逐步为世人所认识、欢迎和推广使用。特别是 1977 年“可移植 C 语言编译程序”推出之后，大大促进了 C 语言独立于 UNIX 系统。目前，几乎在各种硬件平台上，在形形色色的软件环境下，C 语言都是程序设计人员解决各类问题的最主要的编程语言之一。图 1-1 所示为 C 语言的派生关系。



图 1-1 C 语言的派生关系

1978 年, B.W.Kernighan 和 D.M.Ritchie 合著了《The C Programming Language》一书, 书中讲述的 C 语言语法成为后来广泛使用的 C 语言版本的基础, 一般把它称为**经典 C**。随着微型机的迅速普及, 出现了一大批 C 语言系统。虽然它们之间具有很好的兼容性, 但由于没有统一的标准, 必然存在若干差异, 从而给不同平台间的程序移植带来困难。为了改变这种局面, ANSI (American National Standards Institute) 于 1983 年设立了一个委员会, 专门制定 C 语言标准, 这就是常说的**ANSI C**。ANSI C 与经典 C 相比, 有了很大的发展。B.W.Kernighan 和 D.M.Ritchie 在 1988 年对他们的经典著作《The C Programming Language》进行了修订, 使之符合 ANSI C 标准。随着制定以 UNIX 为基础的操作系统标准 POSIX (Portable Operating System Interface for Computer Environments) 这一工作的开展, ISO 于 1990 年通过了 C 程序设计语言的国际标准, 称之为**标准 C**。它是以 ANSI C 为基础, 吸收了各国有关专家和协会的意见后制定的。从此, C 语言就有了统一的国际标准, 这标志着 C 语言进入一个崭新的成熟时代。

### 1.1.2 C 语言的特性

在众多程序设计语言中, C 语言能独树一帜, 长盛不衰, 展示出强大的生命力, 这源于

C语言具有很多显著的优点。C语言的主要特点如下。

### 1. 语言表达能力强

C语言是面向结构化程序设计的语言，有良好的通用性，可以在各种硬件平台上运行；它可以直接处理字符、数字和地址，可以完成通常由硬件设备实现的算术、逻辑运算（如位运算、字节运算，对地址的有关操作等）；可以充分地反映出当前计算机的性能，能取代汇编语言编写各种系统软件和应用软件。由于C语言既具有高级语言（如PASCAL、Fortran、Ada等）的功能，又具有低级语言（如汇编语言）的很多特性，所以人们往往把C语言称为“中级语言”。但这并不意味着C语言功能差，恰恰相反，它表明C语言把高级语言的基本结构与低级语言的实用性两者有机地结合起来。

### 2. 语言简洁，使用方便、灵活

C语言在表示方式上力求简单易行，如：用一对花括号“{ }”表示复合语句的BEGIN、END，利用赋值运算符（如+=、-=、\*=、/=等）表示进行相应运算并且将结果赋给左值（赋值号左边的变量），等等。另外，C语言把一般语言的许多成分都通过显式函数调用来完成，使得编译程序相对小而精。例如，C语言本身没有提供输入/输出机制，也没有并行操作、同步或协同程序等复杂控制，而是提供了大量而有效的库函数来实现输入/输出、字符串处理及存储分配等功能。对库函数可以根据需要方便地予以扩充。这样，就使得C语言编译程序小巧、紧凑。

### 3. 运算符丰富

C语言是一种表达式语言，它有功能很强的运算符（见附录A），如：增1运算符（++）、减1运算符（--）、取地址运算符（&）、间接运算符（\*）等，用这些运算符可构成书写简洁而功能很强的表达式，从而提高软件生产效率。由于运算符类型极其丰富，从而使表达式的类型灵活、多样，在其他高级语言中难以实现的运算，在C语言中能很容易地办到。

### 4. 生成的代码质量高

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，除了要求该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。如果代码质量低，系统开销就大，那就失去实用价值。许多试验表明，针对同一问题，用C语言编写的程序一般所生成的目标代码的效率仅比用汇编语言生成的目标代码的效率低10%~20%。由于描述问题算法时用高级语言比用汇编语言简单、快捷、工作量小，可读性好，易于调试、修改和移植，因而C语言就成为人们描述系统软件和应用软件比较理想的工具。在代码质量方面，C语言确实可与汇编语言媲美。这是其他高级语言尚无法与之匹敌的。

### 5. 具有良好的可移植性

用C语言编写的程序很容易进行移植。在一种工作平台上开发的软件只需稍作修改，甚至不做任何修改，就可以在其他工作平台上运行，从而使开发的软件独立于具体的计算机体系结构和软件环境，达到“共用”的目标，提高了软件的生产效率，节省了用户的投资。



## 6. 是结构化语言

虽说 C 语言不是严格定义的“模块结构语言”，但其结构类似于 ALGOL、Pascal 和 Modula-2 等语言，具有结构化语言的一系列特征。所以，通常仍将 C 语言称为结构化语言。结构化语言的一个显著特点是实现代码和数据的隔离，从而使程序之间很容易实现程序段的共享。C 语言的主要结构成分是函数。利用分隔化的函数，调用者仅需知道它实现什么功能，而不必知道其功能是如何实现的（当然，过多使用外部变量会产生某些副作用！）。C 语言具备现代化语言的各种数据结构，既有简单数据类型，如整型、字符型、浮点型、双精度型等，又有聚合类型，如数组、结构、指针等。C 语言具有结构化的控制语句，直接支持多种循环结构。在书写格式上允许采用逐层缩进形式，增加了程序的可读性。在 C 语言中使用复合语句也同样支持实现程序的结构化和分隔化。当今人们普遍认为，C 语言层次清晰，结构紧凑，比非结构化语言（如 Fortran、BASIC）更易于使用和维护。

C 语言还有其他一些优点，读者可在学习及实践中体会。当然，C 语言也和其他语言一样，存在不足之处，如：某些运算符优先顺序与习惯不完全一致；类型转换比较随便，不是强类型语言等。尽管如此，相比之下，C 语言仍不愧为优秀的程序设计语言之一。

## 1.2 C 程序示例

下面用几个具体的 C 程序作示例，简要介绍 C 程序的结构、各个成分的构成及其作用，使读者对 C 程序有一个感性认识。对这些内容，读者只需做一般了解，先不必予以深究，在以后学到相关内容时，就很容易理解了。

**例 1-1** 计算两个给定的整数之和。

```
#include <stdio.h>
main()
{
    int a, b, sum;
    a=8;
    b=1000;
    sum=a+b*2;
    printf("sum=%d\n", sum);
    return(0);
}
```

这个程序执行后，输出的结果是：

```
sum=2008
```

现在对这个程序进行一些解释：

第 1 行 `#include <stdio.h>` 是文件包含行。它表示：本程序中所用到的某些常量或宏的定