高等院校计算机基础教育改革推荐教材

计算机软件技术基础

陆　勤　王庆瑞　编著

机械工业出版社
CHINA MACHINE PRESS

高等院校计算机基础教育改革推荐教材

计算机软件技术基础

陆 勤　王庆瑞　编著

机械工业出版社

本书将 C 语言程序设计、算法和数据结构等内容融为一体，旨在向读者介绍软件技术中最基本的、也是最重要的知识，即程序设计技术。C 语言程序设计部分包括 C 语言基础、程序控制结构、数组、函数、指针、结构类型、共用体和枚举类型、文件等，算法和数据结构部分包括算法的概念和评价方法，表、树、图等主要结构和栈、队、矩阵、字符串和散列表等基本结构，以及各种排序算法。本书内容丰富，叙述简练，每章都配有练习题。

另外，与本书同步出版的配套教材《计算机软件技术基础实践教程》用于指导读者如何编程并上机调试。

本书可作为大学计算机软件技术基础课程的教材或教学参考书，也可作为广大电脑爱好者学习程序设计方法的自学书籍。

图书在版编目（CIP）数据

计算机软件技术基础 / 陆勤，王庆瑞编著. —北京：机械工业出版社，2005.4
（高等院校计算机基础教育改革推荐教材）
ISBN 7-111-16148-3

Ⅰ. 计… Ⅱ. ①陆…②王… Ⅲ. 程序设计－高等学校－教材 Ⅳ. TP311.1

中国版本图书馆 CIP 数据核字（2005）第 011722 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策　　划：胡毓坚
责任编辑：王　颖
责任印制：石　冉

北京中兴印刷有限公司印刷 • 新华书店北京发行所发行

2005 年 4 月第 1 版 • 第 1 次印刷
787mm×1092mm 1/16 • 21.5 印张 • 530 千字
0001－5000 册
定价：29.00 元

高等院校计算机基础教育改革推荐教材
编委会成员名单

编者的话

在当前的高等教育中，计算机基础教育受到了越来越多的重视，各院校也开始注意结合各专业教学的需求及人才培养的目标，不断地进行改革，使计算机基础教学的水平不断地得到提高。但是，多年来“认识跟不上发展，步伐赶不上变化”的现象仍较为严重。为此，国家教委 2003 年颁发了计算机基础教育白皮书：“关于进一步加强高等学校计算机基础教学的意见”。这对于计算机基础教育领域统一思想认识、加快改革步伐，有着深远的指导意义。

“高等院校计算机基础教育改革推荐教材”（以下简称“推荐教材”）正是在充分研究了这个重要文件后组织编写的。编委会和各教材的作者完全认同“白皮书”对非计算机专业本科毕业生在计算机知识与能力方面应达到的水平的定位，教材的选材完全覆盖了“白皮书”所提出的计算机知识与能力方面应该达到的基本要求。

正如“白皮书”所指出的那样：非计算机专业本科毕业生在计算机知识与能力方面应该达到以下基本要求：

（1）掌握计算机软硬件基础知识：具备使用计算机实用工具处理日常事务的基本能力；具备通过网络获取信息、分析信息、利用信息，以及与他人交流的能力；了解并能自觉遵守信息化社会中的相关法律与道德规范。

（2）具备使用典型的专用软件（包）和软件工具来解决本专业领域中应用问题的能力。

（3）具备利用数据库技术对信息进行管理、加工和利用的意识与能力。

对某些学校、某些专业或部分学生的一些更高的要求。

（1）具备通过建模编程、在本专业领域中进行科学计算的基本能力（偏理工科专业）。

（2）掌握计算机硬件的基本技术与分析方法，具备利用计算机硬件及接口技术解决本专业领域中问题的基本能力（偏工科类专业）。

（3）具备专业领域中计算机应用系统的集成与开发能力（较高要求，对部分学生）。

为了使接受公共计算机课程教学的学习更好地达到上述要求，“推荐教材”努力做到以下四个“加强”：

加强基础核心课程教材的基础性和系统性，强调基本概念、基本技术、方法和理论的准确阐述。

加强技能类教材在计算机技术新发展、新成果方面的介绍，让学生学习到一些先进的开发工具和开发方法。

加强教材的普遍性，使不同院校、不同专业选用方便，还能适应教师指导下学生自主学习的教学模式。

加强教材的实践性环节，“推荐教材”的主教材和上机实验教材配套，教材内容分工合理。

欢迎广大读者对“推荐教材”提出批评和指导。

高等院校计算机基础教育改革推荐教材编委会

前　言

与计算机科学技术日新月异的变化相比，计算机科学技术的专业教育仍显得滞后。因此，加强计算机科学技术教育，推出一批优秀教材，强化计算机软件及程序设计技术的训练，势在必行。

本书是按照非计算机专业计算机课程基本要求中所规定的软件技术基础课程（包括程序设计和数据结构两部分）的教学内容，并参考教育部制定的计算机基础教学主要课程教学大纲，结合目前软件技术发展趋势而编写的。

全书分为上篇 C 语言程序设计和下篇算法和数据结构两部分。上篇共分 9 章，第 1 章 C 语言基础，介绍基本概念、基本数据类型、运算符、表达式、输入输出以及编译预处理等。第 2 章介绍程序控制结构。第 3 章至第 7 章分别介绍数组、函数、结构体、共用体、枚举类型。第 8 章介绍文件用法。第 9 章给出综合示例。

下篇共分 6 章，第 10 章介绍算法和数据结构的意义，包括数据结构的种类、算法描述方法及效率评估方法。第 11 章介绍表结构，包括顺序表和链表以及查找、插入、删除的算法设计。第 12 章介绍栈、队、散列表、字符串、矩阵等表的相关结构。第 13 章介绍树结构，包括树的概念、二叉树的构造和遍历、检索树和哈夫曼树。第 14 章介绍图结构，包括图的定义和存储方法、图的先深搜索和先广搜索、最小生成树和最短路径等。第 15 章介绍几种排序算法。主要有插入排序、堆排序、冒泡排序和快速排序等。

另外，将常用关键字、运算符的优先级和结合性、字符与其 ASCII 码对照表、常用库函数表等作为附录。

上篇中所有程序均在 Turbo C（简称 TC）环境下运行通过，下篇中所有程序（需另配主函数和相关程序段）均在 Visual C++（简称 VC）环境下运行通过。TC 和 VC 的使用方法，见本书的参考文献 1《计算机软件技术基础实践教程》。

本书可用作高等学校非计算机专业本科学生相关课程的教材，旨在培养学生的程序设计能力和初级算法设计能力。一般情况下，教学时数应安排 70 到 90 学时（含上机实践时间），可根据具体条件适当增减教学内容和学时数。多上机练习是学好本书内容的捷径，希望读者通过学习本书的姊妹篇《计算机软件技术基础实践教程》，尽快动手编程及上机调试的技能。

本书上篇由陆勤编写，下篇由王庆瑞编写。在本书编写过程中，得到国防科技大学、解放军理工大学有关领导和人士的大力支持，作者在此表示感谢。

作者特别感谢王元元教授，感谢他对本书编写题纲提出的宝贵意见以及对本书的出版所给予的巨大帮助。还要感谢陈卫卫副教授对本书部分内容所提出的修改意见。

由于作者水平有限，书中难免存在错误或不当之处，敬请读者批评指正。

编　者

目　录

编者的话
前言

上篇　C 语言程序设计

第 1 章　C 语言基础 …… 1
1.1　C 程序的基本结构 …… 1
1.2　基本字符集与标识符 …… 4
1.2.1　基本字符集 …… 4
1.2.2　标识符 …… 4
1.2.3　关键字与标准标识符 …… 4
1.2.4　常量与变量 …… 5
1.2.5　常量标识符 …… 5
1.3　简单数据类型 …… 6
1.3.1　整数类型 …… 6
1.3.2　实数类型 …… 8
1.3.3　字符类型 …… 9
1.4　运算符与表达式 …… 10
1.4.1　算术运算符与算术表达式 …… 10
1.4.2　关系运算符与关系表达式 …… 11
1.4.3　逻辑运算符与逻辑表达式 …… 12
1.4.4　其他运算符 …… 13
1.4.5　类型转换 …… 18
1.5　赋值语句及简单输入、输出 …… 19
1.5.1　赋值语句 …… 19
1.5.2　赋值表达式 …… 20
1.5.3　标准输出函数 printf …… 20
1.5.4　标准输入函数 scanf …… 23
1.6　编译预处理 …… 24
1.6.1　预处理命令 …… 24
1.6.2　宏替换 …… 25
1.6.3　文件包含 …… 26
1.6.4　条件编译 …… 26
1.7　程序设计风格 …… 27
1.8　习题 …… 28
第 2 章　程序控制结构 …… 31
2.1　if 语句 …… 31

2.1.1 简单 if 语句 ……31
2.1.2 复合语句 ……32
2.1.3 扩展 if 语句与嵌套 if 语句 ……32
2.2 switch 语句 ……34
2.3 while 语句 ……35
2.4 do-while 语句 ……36
2.5 for 语句 ……37
2.6 多重循环结构 ……38
2.7 break 语句、continue 语句和 goto 语句 ……39
2.7.1 break 语句 ……39
2.7.2 continue 语句 ……40
2.7.3 goto 语句 ……41
2.8 习题 ……41
第 3 章 数组 ……46
3.1 一维数组 ……46
3.1.1 一维数组说明 ……46
3.1.2 一维数组的初始化 ……47
3.2 二维数组 ……47
3.2.1 二维数组说明 ……47
3.2.2 二维数组的初始化 ……48
3.3 数组应用示例 ……48
3.4 字符数组与字符串 ……51
3.5 习题 ……53
第 4 章 函数 ……61
4.1 函数定义 ……61
4.2 函数调用与函数说明 ……63
4.2.1 函数调用的一般形式 ……63
4.2.2 函数调用的两种方式 ……63
4.2.3 函数说明 ……64
4.3 通过函数参数传递数据 ……66
4.4 函数的递归调用 ……67
4.5 数组作为函数参数传递 ……70
4.6 变量的作用域和存储类别 ……72
4.6.1 变量的作用域 ……72
4.6.2 局部变量和全局变量 ……72
4.6.3 变量的生存期 ……73
4.6.4 变量的存储类别 ……73
4.6.5 变量的存储类别说明符 ……73
4.7 函数的存储类别 ……76

4.7.1 用 extern 说明函数 …… 76
4.7.2 用 static 说明函数 …… 77
4.8 习题 …… 77
第 5 章 指针 …… 85
5.1 指针变量的说明与赋值操作 …… 85
5.1.1 指针和地址的概念 …… 85
5.1.2 指针变量的说明 …… 85
5.1.3 指针变量的初始化 …… 86
5.1.4 指针变量的赋值操作 …… 87
5.2 指针运算 …… 89
5.2.1 指针的赋值运算 …… 90
5.2.2 指针的算术运算 …… 91
5.2.3 指针的关系运算 …… 92
5.3 指向数组的指针变量 …… 93
5.3.1 数组与指针的关系 …… 93
5.3.2 指向一维数组的指针 …… 94
5.3.3 指向二维数组的指针 …… 96
5.3.4 数组指针作为函数参数 …… 97
5.4 指向字符串的指针变量 …… 98
5.4.1 字符数组与字符串的区别 …… 98
5.4.2 指向字符串的指针变量——字符指针 …… 99
5.4.3 字符指针作为函数参数 …… 99
5.5 指向函数的指针变量 …… 101
5.6 返回指针值的函数 …… 102
5.7 指针数组 …… 103
5.8 指向指针的指针变量 …… 106
5.9 习题 …… 107
第 6 章 结构类型 …… 116
6.1 用 typedef 定义类型 …… 116
6.2 结构变量说明与赋值操作 …… 117
6.2.1 结构类型定义 …… 117
6.2.2 结构变量说明 …… 118
6.2.3 结构变量的初始化 …… 119
6.2.4 结构变量的赋值操作 …… 120
6.2.5 结构类型的嵌套 …… 121
6.3 结构数组和结构指针 …… 123
6.3.1 结构数组 …… 123
6.3.2 结构指针 …… 125
6.3.3 结构指针作为函数参数 …… 127

6.4 动态存储分配及释放 …… 129
6.4.1 动态数据结构 …… 129
6.4.2 动态存储分配函数 malloc …… 129
6.4.3 动态存储释放函数 free …… 130
6.4.4 动态存储分配函数 calloc …… 131
6.5 习题 …… 131
第 7 章 共用体和枚举类型 …… 133
7.1 共用体 …… 133
7.1.1 共用体的基本概念 …… 133
7.1.2 共用体类型定义和变量说明 …… 133
7.1.3 共用体变量的初始化 …… 136
7.1.4 共用体变量的赋值操作 …… 136
7.2 枚举类型 …… 140
7.2.1 枚举类型定义和枚举变量说明 …… 140
7.2.2 枚举变量的初始化 …… 142
7.2.3 枚举变量的运算 …… 143
7.3 习题 …… 146
第 8 章 文件 …… 149
8.1 文件的基本概念 …… 149
8.2 文件指针 …… 150
8.3 文件的打开与关闭 …… 151
8.3.1 打开文件函数 fopen …… 151
8.3.2 关闭文件函数 fclose …… 154
8.4 文本文件操作 …… 155
8.4.1 字符读写函数 fgetc 和 fputc …… 155
8.4.2 字符串读写函数 fgets 和 fputs …… 158
8.4.3 格式化读写函数 fscanf 和 fprintf …… 160
8.5 二进制文件操作 …… 161
8.5.1 二进制文件操作的特点 …… 161
8.5.2 数据块读写函数 fread 和 fwrite …… 161
8.6 文件的定位与随机读写 …… 163
8.6.1 文件位置指针当前位置函数 ftell …… 163
8.6.2 文件位置指针复位函数 rewind …… 164
8.6.3 文件位置指针定位函数 fseek …… 165
8.6.4 文件的随机读写 …… 166
8.7 文件检测函数 …… 169
8.7.1 文件结束检测函数 feof …… 170
8.7.2 文件操作出错检测函数 ferror …… 170
8.7.3 出错标志复位函数 clearerr …… 171

8.8 习题 …… 171
第 9 章 综合示例 …… 175
9.1 示例 …… 175
9.2 习题 …… 190

下篇 算法和数据结构

第 10 章 算法和数据结构的意义 …… 193
10.1 数据结构的概念和分类 …… 193
10.2 算法的描述和效率评估 …… 194
10.3 习题 …… 197
第 11 章 表结构 …… 198
11.1 顺序表 …… 198
11.1.1 表结构的定义和存储方法 …… 198
11.1.2 顺序表的基本插入和删除 …… 199
11.1.3 顺序表的查找 …… 200
11.2 简单的单向链表 …… 204
11.2.1 链表的概念 …… 204
11.2.2 插入删除结点时的链操作方法 …… 207
11.2.3 简单链表的构造、查找和输出 …… 210
11.3 其他形式的单向链表 …… 212
11.3.1 带监督元结点的链表 …… 212
11.3.2 循环链表 …… 214
11.4 有序链表的插入和删除 …… 214
11.4.1 有序链表的插入 …… 214
11.4.2 有序链表的删除 …… 216
11.5 双向链表 …… 216
11.5.1 双向链表的结构 …… 216
11.5.2 双向链表的查找插入和删除 …… 217
11.6 静态链表 …… 219
11.6.1 静态链表的含义 …… 219
11.6.2 静态链表的综合程序 …… 221
11.7 习题 …… 223
第 12 章 表的相关结构 …… 227
12.1 栈结构和队结构 …… 227
12.1.1 栈和队的概念 …… 227
12.1.2 栈的实现 …… 228
12.1.3 队的实现 …… 230
12.1.4 栈的应用 …… 233
12.2 散列表 …… 237

12.2.1 散列表的概念和散列函数设计方法 …… 237
12.2.2 散列表的构造和查找 …… 238
12.3 字符串 …… 242
12.3.1 字符串的运算和存储结构 …… 242
12.3.2 模式匹配的实现 …… 243
12.4 矩阵 …… 246
12.4.1 矩阵的存储 …… 246
12.4.2 矩阵运算示例 …… 249
12.5 习题 …… 252
第 13 章 树结构 …… 255
13.1 基本概念 …… 255
13.1.1 树的定义和有关术语 …… 255
13.1.2 二叉树 …… 257
13.1.3 普通树、森林和二叉树的相互转换 …… 260
13.2 二叉树的遍历和构造 …… 262
13.2.1 遍历算法 …… 262
13.2.2 遍历的应用 …… 265
13.2.3 遍历序列的前趋和后继 …… 267
13.2.4 用中序序列和先序序列构造二叉树 …… 268
13.2.5 用扩充先序序列构造二叉树 …… 270
13.3 检索树 …… 271
13.3.1 检索树的定义和查找算法 …… 271
13.3.2 检索树的插入和构造 …… 271
13.3.3 检索树的删除 …… 273
13.4 哈夫曼树 …… 277
13.4.1 哈夫曼算法 …… 277
13.4.2 哈夫曼树的应用 …… 280
13.5 习题 …… 280
第 14 章 图结构 …… 283
14.1 基本概念 …… 283
14.1.1 图的定义和种类 …… 283
14.1.2 有关术语 …… 284
14.2 图的存储方法 …… 286
14.2.1 邻接矩阵及其顺序存储 …… 286
14.2.2 邻接表 …… 288
14.3 图的遍历 …… 289
14.3.1 先深搜索 …… 290
*14.3.2 先广搜索 …… 292
14.4 最小生成树 …… 294

14.4.1 Kruskal 算法 …… 294
14.4.2 Prim 算法 …… 296
14.5 最短路径 …… 298
14.5.1 Dijkstra 算法的描述 …… 298
14.5.2 Dijkstra 算法的实现方法 …… 300
14.6 习题 …… 302
第 15 章 排序 …… 303
15.1 插入排序 …… 303
15.1.1 排序方法的种类 …… 303
15.1.2 直接插入排序 …… 304
15.1.3 二分插入排序 …… 305
15.2 选择排序 …… 307
15.2.1 直接选择排序 …… 307
15.2.2 堆排序 …… 307
15.3 交换排序 …… 312
15.3.1 冒泡排序 …… 312
15.3.2 快速排序 …… 314
15.4 习题 …… 317
附录 …… 319
附录 A C 常用关键字 …… 319
附录 B C 运算符的优先级和结合性 …… 319
附录 C C 常用字符与 ASCII 代码对照表 …… 320
附录 D C 常用库函数表 …… 321
参考文献 …… 330

上篇　C 语言程序设计

第 1 章　C 语言基础

1.1　C 程序的基本结构

C 语言具有强大的功能以及高度的灵活性，它提供了丰富的数据类型、完备的数据处理运算符以及广泛实用的软件资源。

C 语言具有下列特点：

（1）C 是中级语言

它把高级语言面向用户及低级语言的硬件操纵能力与实用性有效地结合起来。较其他高级语言更接近硬件，同时它又具有通常高级语言的一系列特性，灵活紧凑。

（2）C 是结构化语言

结构化语言显著的特点是代码与数据相分离，程序的各个部分除了必要的信息交流外，彼此独立。它主张程序模块化，各个模块具有独立性，体现抽象与信息隐蔽等原则。这种结构化程序设计方式可使程序层次清晰，便于使用、维护及调试，有利于大型软件的设计和开发。C 语言是以函数形式提供给用户的，这些函数可方便地被调用，从而使程序完全结构化。

（3）数据类型丰富，功能齐全

C 语言具有各种各样的数据类型，能处理与实现各类复杂的数据结构，特别是指针类型，其功能强大，高效灵活。

（4）运算符、运算类型特别丰富，灵活多样

C 包含 30 多种运算符，其灵活多样的表达形式，足以表达并实现各类复杂运算。

（5）可移植性高，适用范围广

C 适用于多种操作系统，其可移植性高、适用范围广。

（6）语法限制少，程序设计的自由度大

C 不是强类型语言，这一特点体现了 C 灵活、多功能的特色，但程序的正确性、可靠性和安全性需由程序员自身的能力来确保。

例 1-1　一个简单的 C 程序。

```
#include <stdio.h>
main( )
{
    printf("This is my first program!\n");
```

```
}
```

程序第 1 行是预处理命令。它的作用是将名为 stdio.h 的文件内容在编译前插入到源程序中，以便程序员能引用其中所包含的标准库函数（如本示例中的函数 printf）。这里，stdio 系 standard input & output（标准输入/输出）的缩写。

程序第 2 行是主函数的起始行。main 是 C 语言中标识主函数的专用名。C 语言规定必须用 main 作为主函数名。main 后面的一对圆括号不能省略，因为它表明 main 是一个函数。一个 C 程序总是从主函数 main 开始执行。main( )称为函数 main 的首部。

程序第 3 行至第 5 行是主函数 main 的函数体。函数体自左花括号{开始，至右花括号}结束。这一对花括号通常被称为语句括号。它们中可以包括一组用以描述数据的说明部分以及一组用以规定操作的语句执行部分。本示例的函数体内仅包含一个用于实现输出的语句，printf 是系统提供的一个标准输出函数，它属于 stdio.h 文件，其后跟有一对圆括号，表明这是一个函数调用，其功能是照原样输出括在两个双引号之内的字符串信息（换行符\n 除外）。换行符\n 是 C 中的一个转义符，它的作用是回车换行，故当输出上述字符串信息之后，光标将移至下一行，即新行的起始位置。

若在计算机上通过键盘正确地键入上述程序，经编译后运行该程序，则将在显示屏上输出如下一行文字。

```
This is my first program!
```

在程序第 4 行的末尾有一个分号。在 C 语言中，分号是语句终止符，它是每一个语句的必要组成成分。

例 1-2 分别给两个实型变量 a 和 b 赋值，计算出它们的比率，并赋予实型变量 ratio，最后输出计算结果。

```
#include <stdio.h>
main( )
/* for calculating the ratio */
{
    float a,b,ratio;
    a=-7.5;
    b=2.0;
    ratio=a/b;
    printf("The ratio is  %f.\n",ratio);
}
```

程序第 3 行是个注释行。它自/*开始，至*/结束。其间的内容由程序员书写，但以能有助于理解程序为宜。程序员在文档中插入注释的目的是为了提高程序的可读性。

程序第 5 行是个变量说明语句，它是主函数体中的说明部分，其中 float 是个系统预定义的专用名词，称为关键字，它表明其后的变量是实数（又称浮点数）类型，而 a，b，ratio 均是变量名，该说明语句表明变量 a，b，ratio 均被说明为实型变量。

程序第 6 行至第 8 行是 3 个同样类型的语句——赋值语句。前两个赋值语句使变量 a 和 b 被分别赋予实数值-7.5 和 2.0；最后一个赋值语句实现把 a 与 b 相除的结果赋予变量

ratio。

程序第 9 行中%f 是一个格式转换说明符，它用来指定对应输出项，即实型变量 ratio 的值在输出时所采用的数据类型和格式，此处系指将按带小数点的十进制形式输出，隐含小数 6 位。

程序运行后，其输出结果是：

```
The ratio is -3.750000.
```

例 1-3 运行下述程序将获得与例 1-1 中程序相同的输出结果。

```
#include <stdio.h>
void display_message( )
{
    printf("This is my first program!\n" );
}
main( )
{
    display_message( );
}
```

程序第 2 行是自定义函数 display_message 的首部，其函数体内仅包含一个输出语句。主函数体内则包含了对该函数调用的语句，从而执行该函数体内的输出语句，由前述可知，输出一行字符串信息。函数 display_message 并无返回值，故用 void 表示它是一个无返回值函数。在程序中，程序员可根据解决问题的需要，编写若干个自定义的函数。

以上 3 个简例初步勾划出一个 C 程序的基本结构。概括如下：

1）C 程序是一种函数结构，函数是 C 程序的基本构件。可执行的 C 程序系由一个或多个函数组成，每个函数有其自己的名字，实现指定的功能。任何 C 程序必须有一个名为 main 的主函数。运行 C 程序总是从 main 函数开始，结束于 main 函数最后一个花括号。

2）C 程序中的函数由函数首部与函数体组成。函数首部说明了该函数的名字和类型特征等。函数体是一个由左、右花括号括起来的复合结构，其中可包含若干变量说明及执行语句。变量必须先说明后使用。执行语句包括赋值语句、格式输出语句等。

3）为了输出指定的字符串信息或计算结果，可以使用格式输出语句。这实际上是一个函数调用语句，printf 是一个标准库函数，它在系统预定义头文件 stdio.h 中定义。在程序开始位置处所列出的包含命令

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

为该程序提供了一批有关输入、输出等方面的系统服务。

4）在 C 程序中，任一语句必须以分号作为其终止符。分号不仅仅起语句分隔符的作用，而且是语句的必要组成成分，故不能省略。

5）为了增加 C 程序的可读性，可在适当位置插写注释。注释以/*开头，*/结尾，其间允许有任意字符串，在编译时它被 C 编译器忽略。

6）为便于理解并修改程序，宜按一定的缩进格式来书写并键入程序。对于层次较多、结构较为复杂的程序而言，这样做更显示出其重要性。

例 1-4 最简单的、语法正确从而能编译通过的 C 程序编码是：

```
main(){}
```

1.2 基本字符集与标识符

1.2.1 基本字符集

C 程序是由 C 语言基本字符构成的一个序列。C 的基本字符集是 ASCII（American national Standard Code for Information Interchange——美国国家信息交换标准代码）的一个子集。ASCII 字符集由控制符和图示符组成。标准的 ASCII 字符集共有 128 个 ASCII 代码，其序号由 0 至 127。扩充的 ASCII 字符集共有 256 个 ASCII 代码，其序号由 0 至 255，或由−128 至 127。

C 中使用的基本字符集包括数字字符 0 至 9；英文大、小写字母字符 A 至 Z，a 至 z；一批专用符号，如+，−，*，/，=，<，>，(，)，[，]，{，}，\等等；若干字符对也是专用符号，如==，!=，<=，>=等等。在 C 程序中还用到一些表示特定功能的符号，如换行符，用转义符\n 来表示。

1.2.2 标识符

标识符用以标记常量、变量、语句标号以及用户自定义函数的名字。C 标识符的形成规则是：它必须由英文字母字符或下划线字符开头，后面可跟英文字母、数字和下划线的任意组合。C 对标识符中的英文字母大小写是严格区分的，英文大写字母和英文小写字母被认为是两个不同的字符，因此，data 和 Data 是两个不同的标识符。一个标识符的有效长度即组成该标识符的字符个数与系统实现有关。按照 ANSI C 标准规定，对于一个标识符，C 编译器须有效辨别其前 32 个字符。

例 1-5 下面是一些符合规则的标识符。

```
x1      class _num      FLOAT
```

例 1-6 下面是几个非法的标识符。

3s	以数字开头
s*T	出现非法字符*
−x2	以减号开头
first name	不能包含空格符
YorN?	问号符不允许

1.2.3 关键字与标准标识符

除了由程序员根据程序需要而选用的自定义标识符外，程序中有些名字在 C 中有特定含义，称为关键字（或保留字），如例 1-2 中的 float。它们在程序中具有固定的含义。不能把关键字用作程序员自定义的标识符，否则 C 编译器将指示出错信息。

表 1-1 给出 C 的基本关键字，它们全部都由英文小写字母字符组成。