



高等院校新课程体系计算机基础教育规划教材

C语言程序设计 (第二版)

王宏志 韩志明 主编 张亚辉 郑建明 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等院校新课程体系计算机基础教育规划教材

C 语言程序设计

（第二版）

王宏志 韩志明 主编 张亚辉 郑建明 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是学习 C 语言程序设计的实用教材, 主要内容包括 C 语言概述、数据类型及其表达式、3 种基本结构的程序设计、数组、函数、构造数据类型、指针、编译预处理命令、文件、位运算以及字符屏幕和图形函数等。本书根据教育部计算机课程教学指导委员会颁布的大纲要求, 安排了具有很强的实用性和可操作性的教学内容。

本书适合作为高等学校相关课程或计算机等级考试(二级)的教材, 也可作为高职高专、高等院校成人教育的教材或教学参考书。

图书在版编目(CIP)数据

C 语言程序设计/王宏志, 韩志明主编. —2 版. —北京:
中国铁道出版社, 2009. 2

(高等院校课程体系计算机基础教育规划教材)

ISBN 978-7-113-08804-0

I. C… II. ①王…②韩… III. C 语言—程序设计—高等
学校—技术 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 015244 号

书 名: C 语言程序设计(第二版)

作 者: 王宏志 韩志明 主编

策划编辑: 秦绪好 杨 勇

责任编辑: 秦绪好

编辑助理: 高 爽

责任印制: 李 佳

编辑部电话: (010) 63583215

封面设计: 付 巍

封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 中国铁道出版社印刷厂

版 次: 2009 年 2 月第 2 版 2009 年 2 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 20.25 字数: 473 千

印 数: 5 000 册

书 号: ISBN 978-7-113-08804-0/TP·2835

定 价: 30.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

* 凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

高等院校新课程体系计算机基础教育规划教材

主 任： 何 桥

副主任： 吕英华 邢满堂 苏长龄

委 员： （以姓氏字母先后为序）

卞 清	崔 成	崔荣一	郭成辉
何 桥	李 平	李 严	吕英华
宋绍成	苏长龄	滕国文	王北星
王宏志	邢满堂	于繁华	于江涛
张凌雯	张 强		

C 语言是国内外广泛推广使用的结构化程序设计语言,它功能丰富、表达能力强、使用方便灵活、目标程序效率高、可移植性好,既具有高级语言的优点,又具有低级语言的许多特点。因此,C 语言既可用于开发系统软件,也可用于开发应用软件,应用面很广,许多大型的软件都是采用 C 语言开发的。目前,多数高等院校不仅计算机专业开设 C 语言这门课程,而且,非计算机专业也开设了这门课程。同时,许多学生都选择 C 语言作为参加全国计算机等级考试(二级)的考试科目。学习 C 语言已成为广大学生的迫切需要。

由于国家级计算机等级考试中,C 语言上机考试环境已经由 Turbo C 环境改为 Visual C++ 环境,根据形势的变化,作者对第一版教材的内容做了相应调整和补充,分别介绍了 Turbo C 2.0 和 Visual C++6.0 集成开发环境。

本书根据作者多年的教学实践经验编写而成,同时借鉴和吸取了已有 C 语言教材的优点,对第一版教材进行了修订,第二版教材保持了第一版教材的写作风格,并具有以下特点:

- (1) 教材内容经过精心组织,体系合理、结构严谨,增强了实用性。
- (2) 教材内容组织形式由浅入深、循序渐进,以便于学生学习并有利于提高学生的程序设计能力。
- (3) 增加字符屏幕、图形函数内容,使本书的深度和广度增强,可作为学生学习 C 语言的参考内容。

本书共分 13 章,全面介绍了在 Turbo C 环境下 C 语言的主要内容,包括基本概念、数据类型、表达式、控制语句、数组、函数、结构体、共用体、指针、编译预处理命令、文件、位运算以及字符屏幕和图形函数等内容,并精选了一部分全国计算机等级考试题(二级 C 语言程序设计)充实到教材中。程序设计是一门实践性很强的课程,在掌握基本概念的基础上,要学会编程并进行上机调试。为了满足教学和实验要求,作者还编写了与本书配套的《C 语言程序设计习题解答与上机指导(第二版)》供读者学习时使用。

本书在吉林省计算机共同课教学专业委员会的指导下编写完成,王宏志、韩志明任主编,张亚辉、郑建明任副主编。具体编写分工:第 1、5、8 章由王宏志编写,第 7、9、10、12 章由韩志明编写,第 6 章由王宏志、韩志明共同编写,第 3、4、11 章由张亚辉编写,第 2、13 章由郑建明编写。全书由王宏志统稿。

随着计算机技术的发展和应用的普及,在高等院校对计算机的教育也在不断发展,新的教育教学体系和思想也在探索中,加之编者水平有限,编写时间仓促,书中难免有疏漏和不足之处,恳请读者和专家批评指正,以便下次修订时更正。

编 者

2008 年 12 月

第一版前言

FOREWORD

为进一步推动高等学校的计算机基础教学改革,提高教学质量,适应新世纪对高级人才知识的需求,掌握一门计算机语言已经成为应用计算机必备条件之一。目前,不论是计算机专业还是非计算机专业的学生,都将C语言作为学习程序设计语言的入门语言。因为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,既具有高级语言的优点,又具有低级语言的许多特点。学习和掌握C语言已成为许多学生的迫切需要。

现组织长期从事计算机基础教学工作的教师编写了《C语言程序设计》一书,编写内容的选择上充分考虑计算机学科发展快、更新快的特点,力图反映新内容,使之具有先进性,同时又兼顾了高等学校计算机语言教学的实际情况使之具有现实可行性,尽量做到少而精,力求通俗易懂。本教材的主要特点是:由浅入深、循序渐进地讲解C语言程序设计的思想和方法。全书在编写过程中,力求做到概念准确、内容正确、由浅入深、循序渐进、繁简适当。每章都有小结和习题,通过实例和习题加深基本概念的理解和掌握,提高计算机操作的水平。为进一步满足本书的教学和实验要求,作者还编写了与本书配套的《C语言程序设计习题解答与上机指导》供读者学习时选用。

本书适合作为高等学校及高职高专各专业学生相关课程或参加计算机等级考试的教材,也可作为高等院校成人教育的培训教材或教学参考书。

本书由苏长龄、刘威任主编,由王北星、于秀霞任副主编。刘威编写第1章~第5章,于秀霞编写第6章和第7章,苏长龄编写第8章和第9章,王北星编写第10章和第11章,全书由苏长龄统稿。感谢在本书的编写和出版过程中,中国铁道出版社给予的大力帮助和支持。

由于编者水平和经验有限,书中难免有不足之处,恳请读者提出宝贵意见和建议。

编者

2006年2月

第 1 章 C 语言概述.....	1
1.1 概述.....	1
1.1.1 C 语言及其特点	1
1.1.2 C 语言程序的组成和结构特点	3
1.2 C 语言的基本符号	5
1.2.1 基本符号集.....	6
1.2.2 标识符	6
1.3 算法及其描述方法	7
1.3.1 算法的概念.....	7
1.3.2 算法的描述方法	8
1.4 程序设计方法简介	10
1.5 C 程序的开发环境	12
1.5.1 Turbo C 2.0 集成开发环境.....	13
1.5.2 Microsoft Visual C++ 6.0 的集成开发环境.....	15
本章小结	18
习题一	20
第 2 章 数据类型及其表达式.....	22
2.1 数据类型	22
2.2 常量	23
2.2.1 整型常量	23
2.2.2 实型常量	24
2.2.3 字符常量	24
2.2.4 字符串	25
2.2.5 符号常量	26
2.2.6 转义字符	27
2.3 变量	28
2.3.1 变量的概念.....	29
2.3.2 变量说明	30
2.3.3 变量地址	30
2.3.4 变量的初始化.....	30
2.3.5 数据类型	31
2.4 运算符和表达式.....	33
2.4.1 运算符和表达式	33
2.4.2 赋值运算	34

2.4.3 算术运算	36
2.4.4 逗号运算	38
2.4.5 关系运算	38
2.4.6 逻辑运算	40
2.4.7 条件运算	41
2.4.8 运算符的结合律和优先级	42
本章小结	43
习题二	45
第 3 章 顺序结构的程序设计	49
3.1 顺序结构的语句	49
3.1.1 表达式语句、空语句和赋值语句	49
3.1.2 复合语句	51
3.2 字符数据的输入和输出	51
3.2.1 字符输入函数 getchar()	52
3.2.2 字符输出函数 putchar()	53
3.3 格式输入函数 scanf()	54
3.4 格式输出函数 printf()	56
本章小结	59
习题三	59
第 4 章 选择结构的程序设计	63
4.1 条件选择结构	63
4.2 条件分支的嵌套	65
4.3 开关选择结构	68
4.4 无条件选择结构	74
本章小结	75
习题四	76
第 5 章 循环结构的程序设计	80
5.1 用 goto 语句构成循环结构	80
5.2 while 语句	81
5.3 do...while 语句	83
5.4 for 语句	84
5.5 循环嵌套结构	88
5.6 循环结构中使用 break 和 continue 语句	89
5.7 循环语句之间的差异	91
5.8 程序举例	92
本章小结	94
习题五	94
第 6 章 数组	99
6.1 数组和数组元素	99

6.2 一维数组	100
6.3 多维数组	106
6.4 字符数组	111
本章小结	115
习题六	117
第 7 章 函数	121
7.1 概述	121
7.2 函数定义的一般形式	122
7.2.1 无参函数定义的一般形式	123
7.2.2 带参函数定义的一般形式	123
7.3 函数的参数和函数的返回值	124
7.3.1 形式参数和实际参数	124
7.3.2 函数的返回值	125
7.3.3 数组作为函数的参数	126
7.4 函数的调用	130
7.4.1 函数调用的一般形式	130
7.4.2 调用函数与被调用函数的相对位置关系与函数说明	131
7.5 函数的嵌套调用和递归调用	134
7.5.1 函数的嵌套调用	134
7.5.2 函数的递归调用	135
7.6 局部变量和全局变量	140
7.6.1 局部变量	140
7.6.2 全局变量	141
7.7 存储类型和变量的作用域	142
7.7.1 数据在内存中的存储	143
7.7.2 变量的存储类型	143
7.7.3 变量的作用域和生存期	146
本章小结	146
习题七	148
第 8 章 构造数据类型	152
8.1 结构体	152
8.1.1 结构体类型定义及结构体类型变量的说明	152
8.1.2 结构体类型变量的使用	156
8.1.3 结构体变量的初始化	158
8.1.4 结构体数组	159
8.1.5 结构体和函数	160
8.2 共用体	162
8.2.1 共用体类型的定义和共用体变量的说明	162
8.2.2 共用体成员的使用	163

8.3 枚举类型	164
8.3.1 枚举类型的定义和枚举变量的说明	164
8.3.2 枚举类型数据的使用	165
8.4 用 typedef 定义类型	167
本章小结	168
习题八	169
第 9 章 指针	174
9.1 指针的概念	174
9.1.1 变量和地址	174
9.1.2 指针变量和指针的类型	175
9.2 变量的指针与指针变量	175
9.2.1 指针变量的定义及使用	175
9.2.2 指针变量的初始化	178
9.2.3 指针运算	178
9.3 指针与数组	181
9.3.1 指向数组的指针	181
9.3.2 字符指针与字符数组	185
9.3.3 多级指针及指针数组	188
9.3.4 指针与多维数组	191
9.4 指针与函数	195
9.4.1 函数参数为指针	196
9.4.2 函数的返回值为指针	198
9.4.3 指向函数的指针	200
9.4.4 命令行参数	203
9.5 指针与结构体	205
9.5.1 结构体指针与指向结构体数组的指针	205
9.5.2 结构体指针与函数	207
9.6 链表	209
9.6.1 链表和动态存储分配	210
9.6.2 用包含指针项的结构体变量构成结点	211
9.6.3 用于动态存储分配的函数	214
9.6.4 链表应用举例	215
本章小结	220
习题九	222
第 10 章 编译预处理命令	228
10.1 宏定义	228
10.1.1 不带参数的宏定义	228
10.1.2 带参数的宏定义	229
10.2 文件包含	232

10.3 条件编译	232
本章小结	234
习题十	235
第 11 章 文件	239
11.1 C 语言文件概述	239
11.1.1 文本文件和二进制文件	239
11.1.2 缓冲文件系统和非缓冲文件系统	240
11.1.3 标准输入/输出库函数	240
11.1.4 标准设备文件及 I/O 改向	242
11.2 缓冲型文件输入/输出系统	244
11.2.1 文件 (file) 类型结构及文件指针	244
11.2.2 文件的打开与关闭	245
11.2.3 文件的读写	246
11.2.4 文件的定位	252
11.2.5 出错的检测	254
11.3 非缓冲型文件输入/输出系统	254
本章小结	258
习题十一	259
第 12 章 位运算	263
12.1 基本位运算符与位运算	263
12.1.1 按位与运算符 (&)	263
12.1.2 按位或运算符 ()	264
12.1.3 按位异或运算符 (^)	265
12.1.4 按位取反运算符 (~)	266
12.2 位移运算符与位移运算	266
12.2.1 左移运算符 (<<)	267
12.2.2 右移运算符 (>>)	267
12.3 位运算的复合赋值运算符	268
12.4 位段	268
12.4.1 位段的定义	268
12.4.2 位段的引用	270
12.5 位运算应用举例	271
本章小结	273
习题十二	274
第 13 章 字符屏幕和图形函数	276
13.1 显示器及其工作模式	276
13.2 字符屏幕函数	277
13.2.1 窗口	277
13.2.2 基本输入/输出函数	277

13.2.3	屏幕操作函数	278
13.2.4	字符属性控制函数	280
13.2.5	字符屏幕状态函数	282
13.2.6	程序举例	284
13.3	Turbo C 的图形函数	285
13.3.1	图形模式的初始化	285
13.3.2	屏幕颜色的设置和清屏函数	287
13.3.3	基本图形函数	289
13.3.4	封闭图形的填充	292
13.3.5	有关图形视口和图形操作函数	296
13.3.6	图形模式下的文本输出	298
13.3.7	独立图形运行程序的建立	302
本章小结		302
习题十三		303
附录 A	C 常用库函数	304
附录 B	ASCII 码表	310
参考文献		311

第 1 章 C 语言概述

本章的 1.1 节~1.3 节介绍了 C 语言及其特点, C 语言程序的组成和结构特点, C 语言的基本符号、算法及其描述方法; 1.4 节对结构化程序设计做了简明阐释; 1.5 节介绍了 C 语言在 Turbo C 和 VC++ 6.0 环境下的上机步骤。

1.1 概 述

C 语言是国际上广泛流行的计算机高级语言, 既可以用来编写系统软件, 也可以用来编写应用软件。因为 C 语言功能强大而且又具有低级语言的特性, 所以尽管面向对象的语言应用已十分广泛, 但 C 语言在计算机程序设计领域中仍发挥着重要作用。

1.1.1 C 语言及其特点

1. 程序设计语言的发展

随着计算机的诞生, 用于计算机的程序设计语言随即产生, 人们利用程序设计语言编制程序以便更好地使用计算机。程序设计语言经历了机器语言、汇编语言、高级语言的不同发展阶段。机器语言是计算机能够识别和直接执行的二进制语言, 用机器语言编写的程序不直观, 而且难懂、易出错。下面这两行就是某计算机的机器指令, 功能是计算十进制数 63 与 56 相加之和:

0111010000011111

数 63 送入寄存器 A

0010010000111000

寄存器 A 的内容 63 与 56 相加, 其和 119 送回 A

为了克服机器语言的不足, 人们又发明了汇编语言, 汇编语言是用助记符来表示机器语言, 但计算机不能直接执行, 需通过汇编系统将汇编语言翻译成机器语言。机器语言与汇编语言都是面向机器的语言, 一般称为低级语言。下面这两行就是计算十进制数 63 与 56 相加之和的汇编指令:

MOV A, #63

; 数 63 送入寄存器 A

ADD A, #56

; 寄存器 A 的内容 63 与 56 相加, 其和 119 送回 A

随着计算机的发展, 人们发明了接近自然语言的程序设计语言, 如 FORTRAN、Pascal、BASIC、COBOL、C 以及 C++ 等。这些语言能直接表达计算式和逻辑式, 被称为高级语言或算法语言。例如, $A=63+56$, 就是用 C 语言计算 63 与 56 相加之和的语句。

2. C 语言简介

C 语言是一种得到广泛重视并普遍应用的计算机程序设计语言, 也是国际公认的最重要的几种通用程序设计语言之一, 它既可用于编写系统软件也可用于编写应用软件。

1972 年, C 语言由贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 根据 Thompson 的 B 语言设计而成。B 语言是由一种早期的编程语言 BCPL (basic combined programming language) 发展演变而来的。BCPL 的根源可以追溯到 1960 年的 ALGOL 60 (algol programming language), ALGOL 60 是一种面向问题的高级语言, 离硬件较远。1963 年, 英国剑桥大学推出 CPL (combined programming language) 语言, CPL 改进了 ALGOL 60, 使其能够直接完成较低层次的操作。1967 年, 英国剑桥大学的 Martin Richards 对 CPL 做了改进, 推出了 BCPL 语言。

最初的 C 语言是为描述和实现 UNIX 操作系统提供的一种工具语言, 但 C 语言并没有被束缚在任何特定的硬件或操作系统上, 它具有良好的可移植性。1977 年, 出现了不依赖于具体机器的 C 语言编译文本——可移植 C 语言编译程序, 用该程序编写的 UNIX 系统迅速在各种计算机上运行, 而 UNIX 系统支持的 C 语言也被移植到相应的计算机上。C 语言和 UNIX 系统在发展过程中相辅相成, 得到了广泛应用, 使 C 语言先后被移植到各种大型、中型、小型、微型计算机上。

以 1978 年发表的第 7 版 UNIX 系统中的 C 语言编译程序为基础, B.W.Kernighan 和 D.M.Ritchie 合著了 *The C Programming Language*。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础, 被称为标准 C 语言。1983 年, 美国国家标准化协会 (ANSI) 根据 C 语言问世以来的各种版本对 C 语言的发展和扩充制定了新的标准, 称为 ANSI C。1990 年, C 语言成为国际标准化组织 (ISO) 通过的标准语言。本书介绍的 C 语言部分以 ANSI C 为基础, 书中的程序全部在 Turbo C 2.0 系统中调试通过。

3. C 语言的特点

一种语言之所以能存在和发展, 是因为它有不同于其他语言的特点。C 语言也是如此, 它的特点是多方面的, 人们从不同的角度可总结出众多特点, 但从全面考虑可归纳为以下几点。

(1) C 语言是比较低级的高级语言。有人把 C 语言称为高级语言中的低级语言, 也有人称它是中级语言。它具有许多汇编语言才具备的功能, 如位操作、直接访问物理地址等, 这使 C 语言在进行系统程序设计时显得非常有效, 而过去系统软件通常只能用汇编语言编写。事实上, C 语言的许多应用场合是汇编语言的传统“领地”, 现在用 C 语言代替汇编语言, 减轻了程序员的负担, 提高了效率, 而且写出的程序具有更好的可移植性。

C 语言具有很多接近硬件操作的功能, 但它不能直接处理复合对象。例如, 作为整体看待的字符串、数组等操作, 这些较高级的功能必须通过调用函数来完成。这看起来是个缺陷, 但这种语言规模小, 更容易说明, 学习起来也快。例如, C 语言只有 32 个关键字, 而一些 BASIC 语言, 关键字多达 100 个以上。

(2) C 语言是结构化的程序设计语言。C 语言主要结构成分是函数, 函数允许一个程序中的各任务分别定义和编码, 使程序模块化。C 语言还提供了多种结构化的控制语句, 如用于循环的 for、while、do...while 语句, 用于判定的 if...else、switch 语句等, 十分便于采用自顶向下、逐步细化的结构化程序设计技术。因此, 用 C 语言编制的程序容易理解、便于维护。

(3) C 语言具有丰富的运算能力。在 C 语言中除了一般高级语言使用的算术运算及逻辑运算功能外, 还具有独特的以二进制位 (bit) 为单位的位与、位或、位非以及移位操作等运算, 并且 C 语言具有如 a++、b--等单项运算和 +=、-=等复合运算功能。

(4) C 语言数据类型丰富, 具有现代语言的各种数据类型。C 语言的基本数据类型有整型 (int)、浮点型 (float)、字符型 (char)。在此基础上按层次可产生各种构造类型, 如数组、指针、

结构体、共用体等。同时，C语言还提供了用户自定义数据类型，用这些数据类型可以实现复杂的数据结构，如栈、链表、树等。因此，C语言具有较强的数据处理能力。

(5) C语言具有预处理能力。在C语言中提供了#include和#define两个预处理命令，实现对外部文件的包含以及对字符串的宏定义，同时还具有#if...#else等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率并为程序的组织和编译提供了便利。

(6) C语言可移植性好。目前，在许多计算机上使用的C语言程序，大部分是由C语言编译移植得到的。C编译程序的可移植性使C语言程序便于移植。

C语言的优点很多，但也有一些不足。例如，语法限制不太严格、类型检验太弱、不同类型数据转换比较随便，这就要求程序员对程序设计的方法和技巧更熟练，以保证程序的正确性。总之，C语言已成为国内外广泛使用一种编程语言，并且非常适合用于程序设计语言课程的教学。

1.1.2 C语言程序的组成和结构特点

C语言是函数型语言，函数是构成C语言程序的基本单位。下面通过一个例子来分析C语言程序的组成和结构。

【例 1.1】C语言程序的组成和结构。

```
main()                                /*主函数*/
{
    int a,b,sum;                      /*定义 a、b 和 sum 共 3 个变量*/
    a=3; b=4;                        /*为 a、b 赋值*/
    sum=add(a,b);                    /*调用函数 add，将得到的值赋给变量 sum*/
    printf("sum=a+b=%d\n",sum);      /*屏幕输出 sum 变量的值*/
}

int add(int x,int y)                 /*定义 add 函数和形式参数 x, y*/
{
    int z;                           /*定义 z 变量*/
    z=x+y;                           /*变量 x 与 y 相加的和送给 z*/
    return(z);                       /*返回 z 的值送给 add*/
}
```

运行结果：

```
sum=a+b=7
```

从本例中可以看到，C语言程序有如下特点：

1. C程序由函数组成

C语言源程序由若干个函数组成，函数是C程序的基本单位。组成程序的若干函数中必须有且仅有一个名为main的函数。例1.1中包含两个函数：main()和add()。因为在main()函数中调用add()函数，所以main()为主函数，add()为被调用的函数。被调用函数可以是系统提供的库函数，如例1.1中的printf()函数；也可以是用户根据需要自己定义的函数，如例1.1中的add()函数。一个C程序可以包含零个到多个用户自定义函数。

2. C语言函数由函数首部和函数体两部分组成

(1) 函数的首部：这部分包括函数名、函数类型、参数名和参数类型。

例如, 例 1.1 中 `add()` 函数的说明部分如下:

<code>int</code>	<code>add</code>	<code>(int</code>	<code>x ,</code>	<code>int</code>	<code>y)</code>
↓	↓	↓	↓	↓	↓
函数类型	函数名	参数类型	参数名	参数类型	参数名

函数名后必须有一对圆括号 “()”, 这是函数的标志, 函数类型是函数返回值的类型。参数类型就是形参类型。形参可以有也可以省略, 形参省略时函数名后的一对圆括号不能省略, 如 `main()` 函数就没有参数。如果有参数, 放在圆括号中, 如 `int add(int x,int y)`。

参数类型的说明也可以放在圆括号外, 是传统的函数说明形式, 如:

```
add(x,y)
int x,y;
```

这种参数类型的说明形式和放在圆括号中的参数说明形式 `int add(int x,int y)` 作用一样。

(2) 函数体: 由函数首部下面最外层的一对大括号中的内容组成。一个函数如果有多对大括号, 则最外层的一对大括号中的内容为函数体的范围。

函数体一般包含变量定义 (变量说明) 和执行语句两部分。在例 1.1 中函数 `main()` 中的 `int a,b,sum;` 是变量定义部分, 其余是语句执行部分。

3. 函数 `main()`

C 程序必须有函数 `main()`, 习惯上称其为主函数。C 语言程序总是从函数 `main()` 开始执行, 并且在函数 `main()` 中结束, 这与函数 `main()` 在程序中的位置无关, 函数 `main()` 可以在整个程序的任意位置, 通常把函数 `main()` 放在程序中其他函数的前面。

4. C 程序书写格式自由

C 程序没有行号, 书写格式自由, 一行内可写多条语句, 且语句中的空格和回车符均可忽略不计。一个语句也可以写在多行上, 用 “\” 做续行符。

5. 程序中的每个语句后必须有一个分号

分号 “;” 是 C 语句的一部分。例如, `sum=a+b;` 分号不可少, 即便是程序的最后一个语句也应包含分号。

6. C 语言本身没有输入、输出语句

输入和输出的操作是由库函数 `scanf()` 和 `printf()` 等来完成的。C 语言对输入、输出实行 “函数化”。

7. 可以在 C 程序的任何部分加注释, 以提高程序的可读性

注释使程序变得清晰, 能帮助使用者阅读和理解程序。给程序加注释是一个良好的编程习惯。C 语言注释部分由 “/*” 开始, 至 “*/” 结束, 注释应括在 “/*” 和 “*/” 之间, “/” 和 “*” 之间不允许留有空格。注释部分允许出现在程序中的任何位置。注释可为若干行, 但不允许在 “/*” “*/” 中间又出现 “/*” 和 “*/” 注释。

下面介绍几个简单的 C 语言程序, 以便对 C 程序结构有进一步的了解。

【例 1.2】最小的 C 程序例。

```
main()
{ }
```

这是一个最小的 C 程序, 什么也不做, 但这是符合 C 语言函数规定的程序, 有 `main()` 组成的函数的首部和一对大括号中无任何内容的函数体。这个程序在计算机上运行时没有错误, 因为函

数体是空的, 这个C程序什么也不做。

【例 1.3】函数C程序例。

```
main() /*主函数*/
```

```
{
    Printf("This is a C program.");
}
```

运行结果:

```
This is a C program.
```

这个程序的函数 main()的函数体中有一个输出函数 printf(), 在计算机上运行时 printf()按照原样向显示屏输出双引号中的字符串, 本程序的字符串是英文句子 This is a C program。

【例 1.4】编写C语言程序求键盘输入的两个数中较小的数, 并且将其输出到屏幕上。

```
main() /*主函数*/
```

```
{
    int a,b,c; /*说明3个变量a、b和c*/
```

```
    printf("Input two integers:"); /*输出一行提示信息*/
```

```
    scanf("%d,%d",&a,&b); /*键盘输入两个数值送给a和b变量*/
```

```
    c=min(a,b); /*调用min()函数, 运行结果送给c变量*/
```

```
    printf("min=%d",c) /*输出c变量的值*/
```

```
}
int min(int x,int y) /*定义min()函数, x、y为形参*/
```

```
{
    int z;
    if(x<y) z=x; /*条件语句: 如果x小于y成立z=x, 否则z=y*/
```

```
    else z=y;
    return(z); /*将z的值返回, 通过min()函数带回调用处*/
```

```
}
```

当运行上面这个程序时, 首先, 屏幕上显示一条提示信息:

```
Input two integers:
```

要求用户从键盘输入两个整数。如果用户输入3和5, 即:

```
Input two integers: 3,5 ✓
```

这里, 符号✓表示按【Enter】键, 显示输入的结束。此时屏幕显示运行结果:

```
min=3
```

本程序包括主函数 main()和被调用的函数 min()。程序执行 scanf()函数时, 操作员由键盘输入两个整数值送给 a 和 b 变量。程序执行 c=min(a,b) 时, 调用函数 min(), 将 a 的值送给 x, b 的值送给 y。程序转到 min()函数执行, min()函数中的 if 语句的作用是将 x 变量和 y 变量中的较小值赋给 z 变量。return 语句的作用是将 z 变量的值返回给函数 min(), 同时程序返回主函数执行, 再将 min()函数值送给 c 变量。最后再 printf()函数输出 c 变量的值到屏幕。

1.2 C语言的基本符号

本节主要介绍C语言中的基本符号、用户标识符、保留字以及预定义标识符, 以便读者在编写C语言程序时, 能够正确使用基本符号和标识符, 避免使用非法字符。