

21世纪高等职业教育计算机系列规划教材

C语言 程序设计

徐受蓉 袁可可 主 编

武春岭 李怡平 副主编

- 突出实用性和专业性
- 培养应用能力和岗位工作能力

配备
教案、课件



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

21 世纪高等职业教育计算机系列规划教材

C 语言程序设计

徐受蓉 袁可可 主 编

武春岭 李怡平 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

C 语言是目前国内外广泛使用的程序设计语言之一。C 语言功能强、使用灵活、可移植性好、目标代码质量高, 它既具有高级语言的优点, 又具有低级语言的许多特点, 既适合编写应用程序, 又适合开发系统软件。C 语言程序设计已成为高等院校普遍开设的一门计算机程序设计课程。

全书共分 11 章。主要内容包括 C 语言概述, 数据类型、运算符及表达式, 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, 函数与编译预处理, 指针, 结构体与共用体, 文件和综合实训。我们精选教材内容, 运用案例教学、项目教学, 融入以应用技术为核心的能力培养。

本书可供高职高专学校、继续教育学院和民办高校使用, 也可作为 C 语言培训教材或自学参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

C 语言程序设计/徐受蓉, 袁可可主编. —北京: 电子工业出版社, 2009.6

(21 世纪高等职业教育计算机系列规划教材)

ISBN 978-7-121-08843-8

I. C… II. ①徐…②袁… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 075177 号

策划编辑: 徐建军

责任编辑: 徐 磊

印 刷: 北京市李史山胶印厂

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 15 字数: 384 千字

印 次: 2009 年 6 月第 1 次印刷

印 数: 4 000 册 定价: 26.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

《C 语言程序设计》读者意见反馈表

尊敬的读者：

感谢您购买本书。为了能为您提供更优秀的教材，请您抽出宝贵的时间，将您的意见以下表的方式（可从 <http://www.huaxin.edu.cn> 下载本调查表）及时告知我们，以改进我们的服务。对采用您的意见进行修订的教材，我们将在该书的前言中进行说明并赠送您样书。

姓名：_____ 电话：_____

职业：_____ E-mail：_____

邮编：_____ 通信地址：_____

1. 您对本书的总体看法是：

☐很满意 ☐比较满意 ☐尚可 ☐不太满意 ☐不满意

2. 您对本书的结构（章节）：☐满意 ☐不满意 改进意见_____

3. 您对本书的例题：☐满意 ☐不满意 改进意见_____

4. 您对本书的习题：☐满意 ☐不满意 改进意见_____

5. 您对本书的实训：☐满意 ☐不满意 改进意见_____

6. 您对本书其他的改进意见：

7. 您感兴趣或希望增加的教材选题是：

请寄：100036 北京市万寿路 173 信箱高等职业教育分社 收

电话：010-88254565 E-mail: gaozhi@phei.com.cn

前 言

C 语言是一种结构化的程序设计语言，是程序员的入门语言，也是许多高校计算机语言教学的首选程序设计课程之一。C 语言功能强、使用灵活、可移植性好，它兼有高级语言和低级语言的优点，既适合编写应用程序，又适合开发系统软件，是国内外广泛使用的计算机语言之一。

本书按 C 语言教学大纲要求，针对高职高专学生及入门语言这样的特点，总结多年教学经验，系统地讲述了 C 语言的基本内容。在实例的安排上循序渐进、深入浅出、精心设计，仔细选择了大量具有代表性的例题和习题，使读者既掌握了 C 语言的基本概念、基础知识和技能，又拓宽了编程的思路。积极探索任务驱动、项目导向等有利于增强学生能力的教学模式。

本书根据高职教育的办学宗旨及高素质技能型 IT 专门人才的特点，从程序设计的基本概念和基础知识出发，立足“理论适度，着眼技术，注重实用，培养能力”的原则，对 C 语言程序设计知识及应用技术进行了系统介绍。注重融“教、学、做”为一体的教学方法和手段，强化学生能力的培养。

本书突出 C 语言程序设计课程的应用性、实践性特点，始终贯穿能力培养主线。通过强化应用性教学内容，力求达到在应用中学习知识、培养能力的目的。

本书由徐受蓉、袁可可主编，武春岭、李怡平副主编，参加本书编写的还有蒋文豪、赵波、熊登峰、陈磊、林娟。全书由徐受蓉统稿定稿。

由于作者水平有限，书中的不足和错误在所难免，恳请广大读者批评指正。

为了方便教学，本书配有电子课件，相关教学资源请登录 www.huaxin.edu.cn 或 www.hxedu.com.cn 免费下载。

编 者

目 录

第 1 章 C 语言概述	(1)
1.1 C 语言简介	(1)
1.2 简单的 C 程序介绍	(3)
1.2.1 C 语言程序的总体结构	(3)
1.2.2 C 语言程序的书写格式	(5)
1.3 C 程序的上机步骤	(6)
1.3.1 从输入到运行一个 C 语言程序的一般过程	(6)
1.3.2 在 Turbo C 下运行 C 程序的步骤	(6)
1.4 小结	(10)
1.5 实训	(10)
1.5.1 实训目的	(10)
1.5.2 实训内容	(10)
1.5.3 实训过程	(11)
1.5.4 实训总结	(11)
思考与习题	(11)
第 2 章 数据类型、运算符及表达式	(13)
2.1 常量和变量	(13)
2.1.1 常量与符号常量	(13)
2.1.2 变量	(14)
2.2 C 语言的基本数据类型	(15)
2.2.1 整型数据	(15)
2.2.2 实型数据	(17)
2.2.3 字符型数据	(18)
2.3 运算符和表达式	(21)
2.3.1 C 运算符简介	(21)
2.3.2 算术运算符和算术表达式	(21)
2.3.3 自增自减运算符及表达式	(23)
2.3.4 逗号运算符和逗号表达式	(24)
2.3.5 赋值运算符与赋值表达式	(25)
2.3.6 关系与逻辑运算符及表达式	(25)
2.3.7 条件运算符与条件表达式	(27)
2.4 小结	(28)
2.5 实训	(28)
2.5.1 实训目的	(28)
2.5.2 实训内容	(28)
2.5.3 实训过程	(28)
2.5.4 实训总结	(29)

思考与习题	(29)
第3章 顺序结构程序设计	(31)
3.1 C 语句概述	(31)
3.1.1 表达式语句	(31)
3.1.2 函数调用语句	(31)
3.1.3 空语句	(32)
3.1.4 复合语句	(32)
3.1.5 流程控制语句	(33)
3.2 程序的三种基本结构	(33)
3.2.1 顺序结构	(33)
3.2.2 选择结构	(34)
3.2.3 循环结构	(34)
3.3 数据的输入输出	(35)
3.3.1 数据输出语句	(35)
3.3.2 数据输入语句	(38)
3.4 顺序结构程序设计举例	(42)
3.5 小结	(43)
3.6 实训	(43)
3.6.1 实训目的	(43)
3.6.2 实训内容	(44)
3.6.3 实训过程	(44)
3.6.4 实训总结	(44)
思考与习题	(44)
第4章 选择结构程序设计	(48)
4.1 if 语句	(48)
4.1.1 if 语句三种形式	(48)
4.1.2 if 语句的嵌套	(51)
4.2 switch 语句	(53)
4.2.1 switch 语句的一般形式	(54)
4.2.2 switch 语句的执行过程	(54)
4.2.3 switch 语句应用举例	(54)
4.3 选择结构程序设计举例	(56)
4.4 小结	(58)
4.5 实训	(59)
4.5.1 实训目的	(59)
4.5.2 实训内容	(59)
4.5.3 实训过程	(59)
4.5.4 实训总结	(62)
思考与习题	(62)
第5章 循环结构程序设计	(66)
5.1 while 语句	(66)

5.2	do while 语句	(67)
5.3	for 语句	(69)
5.4	多重循环	(72)
5.4.1	多重循环的概念	(72)
5.4.2	多重循环的结构	(74)
5.5	break、continue 和 goto 语句	(74)
5.5.1	break 语句	(74)
5.5.2	continue 语句	(76)
5.5.3	goto 语句	(77)
5.6	循环结构程序设计举例	(78)
5.7	小结	(82)
5.8	实训	(82)
5.8.1	实训目的	(82)
5.8.2	实训内容	(83)
5.8.3	实训过程	(83)
5.8.4	实训总结	(85)
	思考与习题	(85)
第 6 章	数组	(88)
6.1	数组的概念	(88)
6.2	一维数组	(88)
6.2.1	一维数组的定义	(88)
6.2.2	一维数组元素的引用	(89)
6.2.3	一维数组元素的初始化	(90)
6.3	排序和查找	(91)
6.3.1	排序	(92)
6.3.2	查找	(94)
6.4	二维数组	(95)
6.4.1	二维数组的定义	(95)
6.4.2	二维数组元素的引用	(96)
6.4.3	二维数组的初始化	(97)
6.5	字符数组和字符串	(99)
6.5.1	字符数组的定义	(99)
6.5.2	字符数组的初始化	(99)
6.5.3	字符串的输入和输出	(100)
6.5.4	字符串处理函数	(102)
6.6	小结	(107)
6.7	实训	(107)
6.7.1	实训目的	(107)
6.7.2	实训内容	(107)
6.7.3	实训过程	(107)
6.7.4	实训总结	(109)

思考与习题	(109)
第7章 函数与编译预处理	(112)
7.1 函数的定义	(112)
7.1.1 函数概述	(112)
7.1.2 函数定义	(113)
7.2 函数的参数和返回值	(114)
7.2.1 形式参数和实际参数	(114)
7.2.2 函数的返回值	(115)
7.3 函数的调用	(116)
7.3.1 函数调用的一般形式	(116)
7.3.2 函数调用的方式	(117)
7.3.3 库函数的调用	(117)
7.3.4 自定义函数的调用	(118)
7.3.5 函数的嵌套调用	(119)
7.3.6 函数的递归调用	(120)
7.4 变量	(124)
7.4.1 变量分类	(124)
7.4.2 局部变量和全局变量	(125)
7.4.3 变量的存储类型	(127)
7.4.4 内部函数和外部函数	(132)
7.5 编译预处理命令	(134)
7.5.1 文件包含	(135)
7.5.2 宏定义	(135)
7.5.3 条件编译	(137)
7.6 小结	(139)
7.7 实训	(139)
7.7.1 实训目的	(139)
7.7.2 实训内容	(139)
7.7.3 实训过程	(139)
7.7.4 实训总结	(140)
思考与习题	(140)
第8章 指针	(143)
8.1 指针的概念	(143)
8.1.1 变量存储的相关概念	(143)
8.1.2 变量的访问方式	(144)
8.2 指向变量的指针变量	(145)
8.2.1 指针变量的定义及初始化	(145)
8.2.2 指针变量的引用	(146)
8.2.3 指针变量做函数参数	(150)
8.3 指向一维数组的指针变量	(154)
8.3.1 一维数组指针的概念	(154)

8.3.2	一维数组的指针表示方法	(154)
8.3.3	一维数组元素的引用方法	(155)
8.3.4	通过指针引用数组元素	(156)
8.4	指向字符串的指针变量	(157)
8.5	小结	(160)
8.6	实训	(160)
8.6.1	实训目的	(160)
8.6.2	实训内容	(160)
8.6.3	实训过程	(161)
8.6.4	实训总结	(161)
	思考与习题	(161)
第9章	结构体与共用体	(165)
9.1	构造结构体类型	(165)
9.2	结构体变量的定义、初始化及引用	(166)
9.2.1	结构体变量的定义	(166)
9.2.2	结构体变量的初始化	(168)
9.2.3	结构体变量的引用	(168)
9.3	结构体数组	(169)
9.3.1	定义结构体数组及结构体数组的初始化	(169)
9.3.2	结构体数组应用举例	(171)
9.4	指向结构体数据的指针	(172)
9.4.1	指向结构体变量的指针	(172)
9.4.2	指向结构体数组的指针	(173)
9.4.3	用结构体变量和指向结构体的指针做函数参数	(174)
9.5	共用体	(176)
9.5.1	共用体的概念	(176)
9.5.2	共用体的引用方式	(177)
9.5.3	共用体类型数据的特点	(177)
9.6	小结	(178)
9.7	实训	(178)
9.7.1	实训目的	(178)
9.7.2	实训内容	(178)
9.7.3	实训过程	(179)
9.7.4	实训总结	(180)
	思考与习题	(180)
第10章	文件	(182)
10.1	文件概述	(182)
10.1.1	文件及文件名	(182)
10.1.2	文件分类	(182)
10.1.3	构成文件的基本单元与流式文件	(183)
10.1.4	ANSI C 的缓冲文件系统	(183)

10.2	文件的打开与关闭	(184)
10.2.1	缓冲型文件类型	(184)
10.2.2	非缓冲型文件类型	(187)
10.3	文件的读写函数	(189)
10.3.1	按字符读写的函数	(189)
10.3.2	按字符串读写的函数	(190)
10.3.3	按格式要求读写的函数	(191)
10.3.4	按数据块读写的函数	(193)
10.3.5	文件的其他常用函数	(194)
10.4	小结	(195)
10.5	实训	(195)
10.5.1	实训目的	(195)
10.5.2	实训内容	(195)
10.5.3	实训过程	(195)
10.5.4	实训总结	(197)
	思考与习题	(197)
第 11 章	综合实训	(200)
11.1	成绩管理系统	(200)
11.1.1	成绩管理系统问题描述	(200)
11.1.2	成绩管理系统设计	(200)
11.1.3	成绩管理系统实施阶段	(201)
11.2	电话簿管理系统	(205)
11.2.1	电话簿管理系统问题描述	(205)
11.2.2	电话簿管理系统设计	(206)
11.2.3	电话簿管理系统实施阶段	(207)
11.3	小结	(218)
	思考与习题	(219)
附录 A	常用字符与 ASCII 编码对照表	(220)
附录 B	C 语言中的关键字	(222)
附录 C	运算符和结合性	(223)
附录 D	C 语言常用库函数	(224)
	参考文献	(228)

第 1 章 C 语言概述

教学要求：通过本章的教学，要求读者了解 C 语言的特点，掌握 C 程序的基本结构和书写格式，熟悉 C 程序的编辑、编译、调试和运行过程。

教学内容：首先简单介绍 C 语言出现的历史背景和特点，然后通过 3 个简单的 C 程序实例分析，总结 C 程序的总体结构和书写格式，最后重点介绍 C 程序的开发环境和运行步骤。

掌握程序设计的前提是掌握程序设计语言，在众多的程序设计语言中，C 语言以其灵活性和实用性受到了广大计算机应用人员的喜爱。C 语言是既得到美国国家标准化协会（ANSI）认可，又得到工业界广泛支持的计算机语言之一，几乎任何一种机型、任何一种操作系统都支持 C 语言开发；C 语言在巩固其原有应用领域的同时，又在拓展新的应用领域，支持大型数据库开发和 Internet 应用。一旦掌握了 C 语言，就可以较为轻松地学习其他任何一种程序设计语言，为后续的面向对象程序设计、Windows 程序设计和 Java 程序设计等程序设计语言的学习打下基础。

1.1 C 语言简介

早期的操作系统等系统软件主要是用汇编语言编写的，如 UNIX 操作系统。由于汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好改用高级语言，但一般高级语言难以实现汇编语言的某些功能，而汇编语言可以直接对硬件进行操作，如对内存地址的操作、位（bit）操作等，于是人们设想能否找到一种既具有一般高级语言特性，又具有低级语言特性的语言，集它们的优点于一身。C 语言就在这种情况下应运而生，之后它便成为国际上广泛流行的计算机高级语言。它适合于作为系统描述语言，即用来写系统软件，也可用来写应用软件。

C 语言是由 D.M.Ritchie 于 1972 年在 B 语言的基础上设计的，主要用于编写 UNIX 操作系统。后来 C 语言进行了多次改进，1977 年出现了可移植的 C 语言编译程序，使得 C 语言编写的 UNIX 操作系统可以在各种计算机上使用。随着 UNIX 的广泛应用，C 语言得到了普及和推广，并最终独立于 UNIX 而成为世界上应用最广泛的计算机语言之一。C 语言发展过程如图 1.1 所示。

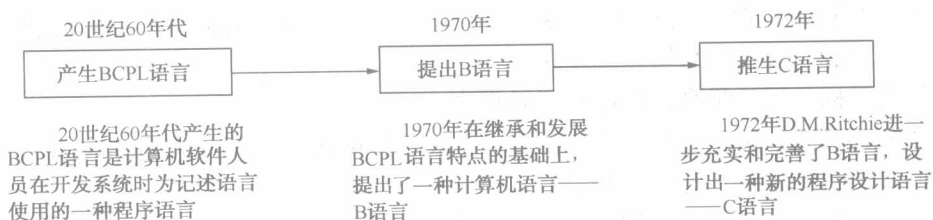


图 1.1 C 语言的发展过程

说明：在 C 语言的推广过程中，出现了多种版本。为了解决这些版本的兼容问题，1983 年美国国家标准协会（ANSI）制定了 C 语言标准，即 ANSI C。目前流行的 C 编译系统都是以它为基础的，如 Microsoft C、Turbo C、Quick C、Borland C 等。这些版本的 C 语言虽然基本部分

相同,但也有些差异,读者可参考相关手册。本书的叙述基本上以 ANSI C 为基础。

C 语言与面向对象的程序设计语言截然不同,在 Windows 如此普及的今天,却依然有大量专业程序员、计算机爱好者,以及非计算机专业的工程技术人员在使用它。C 语言具有如此强大的生命力,正是源于它不可替代的特点。

(1) 语言简洁、紧凑,使用方便、灵活。

C 语言一共有 32 个关键字,9 种控制语句,压缩了一切不必要的成分,程序书写形式自由,语句简练。

(2) 运算符丰富,适用的范围广泛。

C 语言的运算符包含的范围很广,共有 34 种运算符,它把括号、赋值符号、强制类型转换符号等都作为运算符处理,从而使 C 语言的运算符类型极为丰富,表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算和操作。

(3) 数据结构丰富,具有现代化语言的各种数据结构。

C 的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等。这些丰富的数据类型能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。尤其是 C 的指针类型,功能强大、灵活多样,恰当地使用不仅可以简化程序结构,而且可以节省存储空间、提高运算速度。

(4) C 语言是结构式语言

结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护及调试。C 语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(5) 编程限制少,程序设计自由度高。

例如对数组下标越界不做检查,由程序编写者自己保证程序的正确性。对变量的类型使用比较灵活,如整型数据与字符型数据及逻辑型数据可以通用,使得某些运算变得更加简单、直接。

一般的高级语言语法规则和检查比较严格,能检查出几乎所有的语法错误。而 C 语言允许程序的编写有较大的自由度,因此放宽了语法检查。编写者应当仔细检查程序,保证其正确性,而不要过分依赖于编译软件去查错。“限制”和“灵活”是一对矛盾体。限制严格,就失去了灵活性;而强调灵活,就必须放松限制。这一点使得 C 语言较其他语言对程序编写者的要求更高。

(6) 可直接对硬件操作。

C 语言允许直接访问物理地址,直接对硬件进行操作,提供对字节、位、内存和寄存器的操作,能实现汇编语言的大部分功能。这个特点使得 C 语言既具有高级语言的功能,又兼备低级语言的许多功能,可以用来写系统程序。

(7) 生成目标代码质量高,程序执行效率高。

C 语言生成的目标代码一般只比汇编语言生成的目标代码的效率低 10%~20%。

(8) 用 C 语言编写的程序可移植性好。

与汇编语言相比,用 C 语言编写的程序基本上不做修改就能用于各种型号的计算机和操作系统,使程序具备了很好的移植性。

以上所述的是 C 语言的一般特点,至于其内部的其他特点将结合以后各章节内容逐一进行介绍。正是 C 语言的这些特点,使得它的应用非常广泛。许多大的软件都用 C 编写,这主要是由于 C 的可移植性好和对硬件的控制能力高,表达和运算能力强。许多以前只能用汇编语言处理的问题现在可以改用 C 语言来处理了。

总之，C语言对编程者要求较高。由于使用C语言编写程序会感到限制很少，灵活性大，功能强，可以编写出任何类型的程序，因此学习和使用C语言的人越来越多。

1.2 简单的C程序介绍

1.2.1 C语言程序的总体结构

一个完整的C语言程序，是由一个main()函数（又称主函数）和若干个其他函数组合而成的，或者仅由一个main()函数构成。下面我们通过几个例子，从中分析C程序的组成和形式。

例 1.1 仅由main()函数构成的C语言程序。

```
/*程序 L1_1.C 功能：在屏幕上显示一句话：This is the first C program! */
#include<stdio.h>    /*包含输入输出头文件*/
main()              /*定义主函数*/
{
    printf("This is the first C program!\n");/*在屏幕上输出字符串且光标移到下一行*/
}
运行结果为：
This is the first C program!
```

(1) C程序的注释信息可以出现在程序的任何地方，它总是以“/*”符号开始，以“*/”符号结束。注释是给人看的，目的是方便程序的阅读，对编译和运行不起作用，不会出现在目标程序中。

(2) 由于程序中用到输出功能函数printf()，因此在程序开始时需要将包含该函数的标准输入输出头文件stdio.h导入，即#include<stdio.h>。

(3) main表示“主函数”，一个能够运行的C程序必须有且仅有一个main()函数，这样程序运行时才可以找到入口。

(4) 大括号{}括起来的部分称为函数体，在其中写入该函数要执行的功能。本例中主函数内使用的printf()是C语言的输出函数，将双引号中的字符串原样输出。“\n”是换行符，即在“This is the first C program!”输出之后回车换行。

(5) 每一个语句后用“;”(分号)表示结束。

例 1.2 仅由main()函数构成的C语言程序（函数体中使用到变量）。

```
/*程序 L1_2.C 功能：求两个整数之和 */
#include<stdio.h>
main()
{
    int a,b,sum; /* 声明部分，定义整型变量 a、b、sum */
    a=12;b=34; /* 给变量 a、b 赋值 */
    sum=a+b; /* 将变量 a 和 b 之和赋给变量 sum */
    printf("sum is %d\n",sum); /* 输出变量 sum 的值 */
}
运行结果为：
sum is 46
```

(1) 主函数内容由大括号括起来，包括定义所有要使用的变量（用int说明a、b、sum都是

整数), 为变量赋值 (将常数赋值给 a、b), 运算 (sum 等于 a、b 之和), 以及输出 (%d 表示执行输出时此位置将由变量 sum 的十进制整数值代替)。

(2) 每一个语句结束时加 “;”。

例 1.3 由 main() 函数和 1 个其他函数 min() 构成的 C 语言程序。

```
/*程序 L1_3.C 功能: 通过键盘输入两个实数, 输出较小的一个*/
#include<stdio.h>
float min(float x,float y) /*定义求实数 x,y 的最小数的函数 min */
{
    float z; /*min 函数中的声明部分, 定义本函数中用到的实型变量 z */
    if (x<y) z=x; /* 如果 x 小于 y, 则将 x 的值赋给 z */
    else z=y; /* 否则将 y 的值赋给 z */
    return(z); /* 返回 z, min 函数值为 z 的值 */
}
main() /* 定义主函数 */
{
    float a,b,c; /*声明部分, 定义 a、b、c 三个变量为实型变量*/
    printf("Please input two float numbers (a,b):"); /* 输出屏幕提示信息 */
    scanf("%f,%f",&a,&b); /* 从键盘输入变量 a 和 b 的值 */
    c=min(a,b); /*调用 min 函数, 将得到的较小的数值返回值赋给变量 c */
    printf("min=%f",c); /* 输出 c 的值 */
}
运行结果为:
Please input two float numbers (a,b): 3.4, 5.6↵
min=3.4900000
```

(1) 程序由主函数 main() 和用户自定义函数 min(float x, float y) 组成。用户自定义函数 min(float x, float y) 用于求出给定两个实数 x、y 中较小的一个。

(2) 用户自定义函数 min(float x, float y) 内容在大括号内, 包括定义变量 z, 进行运算 (用 if-else 语句实现将 x 和 y 中较小的一个数值赋给 z), 最后 return (z) 使得 min(float x, float y) 的结果等于 z。

(3) 主函数内容同样在大括号内, 包括定义所有要使用的变量, 在执行时屏幕先给出提示信息, 然后 scanf() 输入函数等待用户通过键盘输入两个数给 a、b (注意两个数之间用逗号间隔), 接着 c 等于 min(a, b) (即 a、b 的最小数), 最后输出实数 c 的值。

(4) 每一个语句结束时加 “;”。

 **重点:** C 语言程序结构特点。

通过对以上几个例子的分析, 我们可以总结出 C 语言程序结构具有如下特点。

(1) 函数是 C 程序的基本组成单位。

main() 函数的作用相当于其他高级语言的主程序, 其他函数的作用相当于子程序。

其中, 被调用的函数可以是系统提供的库函数 (printf() 和 scanf() 函数), 也可以是用户根据需要自己编制设计的函数 (如 min() 函数)。C 的函数相当于其他语言中的子程序, 它用函数来实现特定的功能。可以说 C 是函数式的语言, 程序中的全部工作都是由各个函数分别完成的。编写 C 程序就是编写一个个函数。C 的函数库十分丰富, 此特点使之容易实现程序的模块化。

(2) C 语言程序总是从 main 函数开始执行的, 而不论 main 函数在整个程序中的位置如何。

(3) C 语言本身没有输入和输出语句。输入和输出的操作是由库函数, 如 scanf() 和 printf()

等函数来完成的，即 C 对输入输出实行“函数化”。

(4) 一个函数由两部分组成。

① 函数的首部（函数的第一行）。

例: float	min	(float	x,	float	y)
↓	↓	↓	↓	↓	↓
函数类型	函数名	函数参数类型	函数参数名	函数参数类型	函数参数名

注意：一个函数名后面必须跟一对圆括号，函数参数可以没有，如 main()。

② 函数体，即函数首部下面的大括号内的部分。

如果一个函数内有多个大括号，则最外面的一对为函数体的范围。

函数体一般包括以下两部分。

声明部分：在这部分中定义所用到的变量。

执行部分：由若干个语句组成。

注意：

(1) 函数体中的变量定义语句，必须在所有可执行语句之前。

例如，下面程序中变量定义语句“int sum;”的位置是非法的：

```
main()
{
    int a,b; /* 变量定义语句：定义两个整型变量 a、b */
    a=2; /* 可执行的赋值语句：将 2 赋值给变量 a */
    b=3; /* 可执行的赋值语句：将 3 赋值给变量 b */
    int sum; /* 变量定义语句：出现在可执行的赋值语句"a=2;"和"b=3;"之后，非法！ */
    sum=a+b;
    printf("sum=%d\n",sum);
}
```

思考题：如何解决此问题？

(2) 如果不需要，也可以缺省变量定义语句（如例 1.1）。甚至可以既无声明部分，又无执行部分，如：

```
dump(){ }
```

1.2.2 C 语言程序的书写格式

(1) C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上。各语句之间用分号分隔，每个语句和数据定义的最后必须有一个分号。

(2) 可以用/*.....*/对 C 程序中的任何部分做注释，以提高程序的可读性。

注意：“/”和“*”，以及“*”和“/”之间不能有空格，否则会出错。

 提示：为了避免遗漏必须配对使用的符号，如注释符号、函数体的起止标识符（大括号）和圆括号等。在输入时，可连续输入这些起止标识符，然后再在其中进行插入来完成内容的编辑。这在起止标识符嵌套时，以及相距较远时，更有必要。

(3) 注释的位置可以单独占一行，也可以跟在语句的后面。另外，如果一行写不下，可另起一行继续写。

(4) 注释中允许使用汉字。在非中文操作系统下，看到的会是一串乱码，但不影响程序运行。