

新世纪高职高专课程与实训系列教材

C语言程序设计 基础与实训教程

姜 丹 主 编
匡桂阳 刘迎春 杨秀芳 副主编

课 程

与

实 训



清华大学出版社

新世纪高职高专课程与实训系列教材

C 语言程序设计基础与实训教程

姜 丹 主编

匡桂阳 刘迎春 杨秀芳 副主编

清华大学出版社

北 京

内 容 简 介

C 语言用于编写 DOS 或 Windows 平台上使用的应用程序,是一种结构化的程序设计语言,近年来在国内外得到迅速的推广使用。

作者在多年教学经验的基础上,根据学生的认知规律精心组织了本教材内容,并通过大量有现实意义的例题,循序渐进地介绍了 C 语言程序设计的有关概念和编程技巧。

书中例题都经过了仔细的调试,且配有大量的上机实训题和课后习题,并为教师配有上机实训参考答案和教学详解,读者可从 <http://www.wenyuan.com.cn> 下载。

本书概念清晰、例题丰富、深入浅出、知识结构及深度合理,可作为高等职业学校和高等专科学校的教材,也可作为计算机培训班的教材及自学者的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C 语言程序设计基础与实训教程/姜丹主编;匡桂阳,刘迎春,杨秀芳副主编. —北京:清华大学出版社,2006.9

(新世纪高职高专课程与实训系列教材)

ISBN 7-302-13763-3

I. C… II. ①姜…②匡…③刘…④杨… 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(2006)第 103297 号

出 版 者:清华大学出版社

地 址:北京清华大学学研大厦

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

邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:凌宇欣

文稿编辑:宋延清

封面设计:陈刘源

印 刷 者:北京密云胶印厂

装 订 者:北京国马印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:17.5 字数:375 千字

版 次:2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

书 号:ISBN 7-302-13763-3/TP·8281

印 数:1~5000

定 价:25.00 元

丛 书 序

21 世纪人类已迈入“信息社会”的新时代,科学技术正发生着深刻的变革,整个社会对德才兼备高素质应用型人才的需求更加迫切。如何培养出符合时代要求的优秀人才,是全社会尤其是高等院校面临的一项急迫而现实的任务。

社会对学生的职业能力要求催化出新型的课程结构和教学模式。新型教学模式必须是以工作为基础的模仿式学习。教材作为知识的载体,是人才培养过程中传授知识、训练技能和发展智力的重要工具,同时也是学校教学和科研水平的重要反映。教材在教学中起到稳定教学秩序、保证教学质量、创新教学内容以及主导教学方向的作用。同时,教材内容的革新也是课程建设的重要组成部分。而新型教材模式必须以新型的教材内容为依托,是原有教材的有益补充。

为了适应高职高专院校应用型人才培养迅速发展的需要,本着厚基础、重能力、求创新的总体思想,培养以就业市场为导向的具备“职业化”特征的高级应用型人才,着眼于国家发展和培养复合型人才的需要,着力提高学生的学习能力、实践能力和创新能力。我们联合全国著名的职业院校计算机专业的有关专家组成了《高职高专课程与实训系列教材》编审委员会,全面研讨了新形势下计算机和信息技术专业的课程建设及人才培养方案,组织了本系列面向应用的、切合新一轮教学改革和高校教材建设目标的《高职高专课程与实训系列教材》——计算机系列。

本丛书以“理论与应用并重,基础与实践兼顾”为原则,理论知识做到三用一新,即“实用、适用、够用和创新”,并在讲解理论知识的同时充分融合了丰富的案例与上机实训,真正做到了理论知识与实训内容合二为一。

本丛书是教材改革的创新之作,它的出版定将真正切合当前教育改革的需要。

本套教材的主要特色

1. 双师型的教材编写模式

本丛书针对高职院校以及部分应用型本科院校计算机相关专业学生编写,以实用性为基础,以问题驱动为导向,以培养高级专门人才为目标,突出实践教学环节。为保证教学案例的实践性,每本教材均能做到如下几点:

- 至少有一名本学科的知名专家或学科带头人提出指导意见。
- 至少有一名高等院校教学一线的资深教师参与组织编写。
- 至少有一名计算机行业专家负责整理教学案例及配套资源。

2. 就业导向型的教材定位

面向高职院校人才培养模式的新需求,面向教育部颁布的新的学科专业调整方案和高校教材建设目标。根据行业需求,构建以能力为本的课程创新体系,把以能力为本的课程设置与我国劳动和社会保障部推行职业资格证书制度的培训互相接轨。

努力使系列教材的理论背景充分体现“以行业为导向、以能力为本位、以学生为中

心”的发展趋势，培养“学术型”与“应用型”相结合的人才。使教材建设具有实用性和前瞻性，与就业市场结合得更加紧密。

3. 基础理论与应用实践合二为一的内容体系

本套教材打破了一本主教材配套一本实训教材的传统课程分配方式，使得学生在学习完基础理论知识后能通过案例课及实训课尽快加深对知识点的理解与掌握，不需要再单独开设实训课程，也消除了因不同的教师教授实训课程而存在的与讲课内容相脱节的问题，使得知识点的讲解与实训课程能充分融合。

本套教材在内容安排上遵循适用性原则，使教材在结构纵横的布局、内容重点的选取、示例习题的设计等方面均符合教改目标和教学大纲的要求，把教师的备课、试讲、授课、辅导答疑等教学环节有机地结合起来。

4. 立体化的教材服务

为了在内容、体例上更适应教学需求，本套丛书跳出以往单一的纸介质的图书模式，推出了一套多元化的教学服务产品，每本书均包括以下几个部分：

- 图书正本。
- 每章课后的上机实训题(含题目、指导步骤及参考源文件)。
- 图书中的示例源文件。
- 图书配套的电子教案。

5. 读者定位

本系列教材定位于职业教育，主要面向高职高专院校，同时也适用于同等学历的职业教育和继续教育。本丛书以三年制高职为主，也适用于两年制高职学生使用。

6. 网上资源的下载及服务

为方便教师教学和学生学习，本系列教材配有电子课件与上机实训操作答案，需要时可以直接免费上网索取。读者可以直接访问我们的网站 <http://www.wenyuan.com.cn>，下载相关的信息资源。如有其他的问题，也可以在网上留言或发邮件，提出您的宝贵意见和建议，以便于我们随时与您沟通。

丛 书 编 委 会

主 任：吴文虎

委 员(排名不分先后)：

白延丽 占跃华 王熔熔 吴建平 曹建春

曹然彬 吴文庆 万朝阳 陈承欢 陈培植

杨清学 程远东 周朋红 范国渠 冯静哲

徐洪祥 王泽生 何慧荣 胡美香 黄玉春

姜 丹 吕凤顺 李胜军 李 越 杨小劲

刘志成 穆红涛 史宝慧 陶树平 武传宝

张邦文 赵雪林 姜 锐 赵克林 王宇川

前 言

计算机技术的飞速发展,促进了计算机基础教育的发展。根据我国当前教学改革和建设的需要,教育部提出了“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的教学体系。计算机程序设计语言是高等院校各专业的一门基础课程,属于计算机技术基础教育,是当代大学生必须掌握的一种应用技能。

C 语言是近年来国内外得到迅速推广使用的一种现代计算机语言。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势,又具有低级语言的许多特点,特别适合编写系统软件,已经成为计算机类本科生和高职高专生及职专生的必修课程。

为适应培养技能型紧缺人才的需要,进一步补充和完善 C 语言程序设计系列教材,为此编写了《C 语言程序设计基础与实训教程》。此书旨在通过对适量习题的练习和上机实践,提高学生的 C 语言程序设计能力和实际应用 C 语言的能力。

本书以《计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》为依据,在内容上突出以就业为导向,以实践技能为核心,倡导以学生为本位的教育理念,注重全面提高学生的职业实践能力和职业素养。本书遵循大多数初学者的认识规律和学习思路,在内容上力求准确、层次清晰、通俗易懂、实用性强,使学生在少走弯路的前提下对 C 语言产生浓厚的学习兴趣。

全书共分 12 章:

第 1 章介绍 C 语言的特点, C 语言程序的结构和书写规则。

第 2 章介绍 C 语言的数据类型、运算符和表达式,是程序设计最基础的部分。

第 3 章、第 4 章、第 5 章介绍结构化程序设计语言的特点及应用。

第 6 章介绍 C 语言的用户自定义函数的定义、调用、应用等内容。

第 7 章、第 8 章、第 9 章介绍 C 语言中构造数据类型的定义及应用。

第 10 章介绍文件的基本操作和应用。

第 11 章介绍 C 语言提供的多种预处理功能。

第 12 章介绍 C 语言的位运算。

本书不仅可以作为高职学习的指导教材,也可以作为全国计算机等级考试 C 语言二级考试的复习资料,或自学 C 语言的参考用书。

本书的第 1 章由王慧珍编写;第 2 章由肖建芳编写;第 3 章由尹志东编写;第 4 章由杨浩编写;第 5 章由刘迎春编写;第 6 章由杨秀芳编写;第 7 章由匡桂阳编写;第 8 章、第 9 章的结构体指针部分由刘建华编写;第 9 章、第 11 章、附录由姜丹编写;第 10 章由王晓勇编写;第 12 章由陆俊松编写。全书由姜丹统稿。

由于计算机技术发展迅速,加上作者水平有限,书中难免存在缺点和错误,请读者不吝指正。

编者

目 录

第 1 章 基础知识1

- 1.1 程序设计基础1
- 1.2 C 语言简介2
- 1.3 C 语言的程序结构及书写规则3
 - 1.3.1 C 语言的程序结构3
 - 1.3.2 函数的一般结构4
 - 1.3.3 源程序书写格式6
- 1.4 C 语言的基本符号6
- 1.5 习题7

第 2 章 数据类型、运算符和表达式9

- 2.1 C 语言的数据类型9
- 2.2 常量10
 - 2.2.1 整型常量10
 - 2.2.2 实型常量10
 - 2.2.3 字符型常量和字符串常量11
 - 2.2.4 符号常量13
- 2.3 变量13
 - 2.3.1 整型变量14
 - 2.3.2 实型变量17
 - 2.3.3 字符变量18
 - 2.3.4 变量赋初值20
 - 2.3.5 不同类型数据间的相互转换20
 - 2.3.6 数值的定点表示形式与浮点表示形式22
- 2.4 运算符和表达式23
 - 2.4.1 C 语言的运算符简介23
 - 2.4.2 算术运算符和算术表达式24
 - 2.4.3 赋值运算符和赋值表达式26
 - 2.4.4 逗号运算符和逗号表达式28
- 2.5 上机实训29
- 2.6 习题31

第 3 章 最简单的 C 程序设计 34

- 3.1 C 语句概述 34
- 3.2 赋值语句 35
- 3.3 数据输入与输出 37
 - 3.3.1 格式化输出 37
 - 3.3.2 格式化输入 40
 - 3.3.3 字符输入与输出 44
- 3.4 顺序结构程序设计 45
- 3.5 上机实训 47
- 3.6 习题 48

第 4 章 选择结构程序设计 52

- 4.1 关系运算符和关系表达式 52
 - 4.1.1 关系运算符 52
 - 4.1.2 关系表达式 53
- 4.2 逻辑运算符和逻辑表达式 54
 - 4.2.1 逻辑运算符 54
 - 4.2.2 逻辑表达式 55
- 4.3 if 语句 56
 - 4.3.1 if 语句的 3 种形式 56
 - 4.3.2 if 语句的嵌套 59
 - 4.3.3 条件运算符和条件表达式 61
- 4.4 switch 语句 62
- 4.5 程序举例 64
- 4.6 上机实训 66
- 4.7 习题 67

第 5 章 循环控制 74

- 5.1 while 语句 74
- 5.2 do-while 语句 76
- 5.3 for 语句 78
- 5.4 循环的嵌套 81

5.5 循环结构的特殊控制 break 和 continue 的使用	82	7.3.3 字符串数组	139
5.6 goto 语句以及用 goto 语句构成循环	85	7.3.4 字符串处理函数	140
5.7 上机实训	86	7.3.5 字符串数组的应用举例	142
5.8 习题	88	7.4 上机实训	144
第 6 章 函数	92	7.5 习题	147
6.1 概述	92	第 8 章 结构体及其他复合类型	157
6.2 函数定义的一般形式	93	8.1 结构体的定义和使用	157
6.2.1 函数的参数	94	8.1.1 结构体类型的定义	157
6.2.2 函数的返回值	95	8.1.2 结构体变量的使用	161
6.3 C 语言中函数的类型	96	8.1.3 结构体数组	164
6.4 函数的调用	97	8.2 共用体的定义和使用	166
6.5 函数的声明	98	8.2.1 共用体类型的定义	166
6.6 函数的嵌套调用	99	8.2.2 共用体变量的使用	168
6.7 函数的递归调用	100	8.3 枚举类型和用户自定义类型	169
6.8 C 语言中的变量	102	8.3.1 枚举类型的定义和使用	169
6.8.1 全局变量和局部变量	102	8.3.2 用户自定义类型	170
6.8.2 动态存储变量和 静态存储变量	106	8.4 上机实训	171
6.9 内部函数和外部函数	110	8.5 习题	172
6.10 上机实训	111	第 9 章 指针	179
6.11 习题	113	9.1 指针与指针变量的基本概念	179
第 7 章 数组	121	9.2 指针变量的定义与应用	181
7.1 一维数组	121	9.2.1 指针运算符	182
7.1.1 一维数组的定义	121	9.2.2 指针变量的运算	183
7.1.2 一维数组的初始化	123	9.3 数组指针变量的声明和应用	185
7.1.3 一维数组元素的引用	123	9.3.1 数组指针变量	186
7.1.4 一维数组的应用举例	126	9.3.2 数组名和数组指针变量 作函数参数	186
7.2 二维数组	129	9.3.3 指向多维数组的指针变量	187
7.2.1 二维数组的定义	129	9.3.4 动态数组	188
7.2.2 二维数组的初始化	130	9.4 字符串指针变量的声明和应用	190
7.2.3 二维数组元素的引用	131	9.5 函数的指针与函数指针 变量的应用	193
7.2.4 二维数组的应用举例	133	9.6 指针型函数	194
7.3 字符数组	134	9.7 指针数组与主函数 main() 的形参	196
7.3.1 一维字符数组与字符串	134	9.7.1 指针数组	196
7.3.2 字符串的输入与输出	137	9.7.2 主函数 main() 的形参	197
		9.7.3 指向指针的指针变量简介	199

9.8 指向结构体类型的指针.....	200	第 11 章 编译预处理.....	229
9.9 上机实训.....	203	11.1 宏定义.....	229
9.10 习题.....	206	11.1.1 不带参数的宏定义.....	229
第 10 章 文件.....	210	11.1.2 带参数宏定义.....	232
10.1 C 文件概述.....	210	11.2 文件包含.....	236
10.2 文件的打开和关闭.....	211	11.3 条件编译.....	236
10.2.1 文件指针.....	211	11.4 习题.....	238
10.2.2 文件的一般操作过程.....	212	第 12 章 位运算.....	240
10.2.3 文件的打开.....	212	12.1 位运算.....	240
10.2.4 文件的关闭.....	214	12.1.1 “按位与”运算符(&).....	240
10.3 文件的读写.....	214	12.1.2 “按位或”运算符().....	242
10.3.1 文本文件的读写函数.....	214	12.1.3 “异或”运算符(^).....	243
10.3.2 二进制文件的读写函数.....	220	12.1.4 “取反”运算符(~).....	244
10.4 文件的顺序存取和随机存取.....	222	12.1.5 左移运算符(<<).....	245
10.4.1 文件定位操作.....	222	12.1.6 右移运算符(>>).....	245
10.4.2 文件的顺序存取 和随机存取.....	224	12.1.7 位运算赋值运算符.....	246
10.5 文件操作的状态和出错检测.....	225	12.2 位运算举例.....	247
10.6 上机实训.....	226	12.3 位段.....	249
10.7 习题.....	227	12.4 习题.....	251
		附录 A Turbo C 的使用.....	253
		附录 B Turbo C 常用函数中文说明.....	259

第1章 基础知识

学习目的与要求:

C语言由于其功能强、使用灵活、可移植性好、目标程序质量好而受到广泛的欢迎。C语言既具有高级语言的优点,又具有低级语言的许多特点,既可以用来编写系统软件,又可以用来编写应用软件。C语言已成为软件工作者必须掌握的一个工具。

本章主要介绍C语言的特点,通过简单的C语言程序,了解并掌握C语言程序的特点、C语言程序的结构和书写规则。

学习C语言时有一点应当注意:应该把精力放在最基本、最常用的内容上。开始时不要钻牛角尖,有一些细节,随着对C语言的了解逐步深入,和实践经验的逐步丰富,会自然地掌握的,而有些细节则要通过长期的实践才能真正熟练地掌握。

1.1 程序设计基础

今天,人们所讲的计算机是一个计算机系统,而不是一台“裸机”。一个完整的计算机系统由计算机硬件系统和计算机软件系统(俗称程序系统)组成。前者一般为组成计算机的物资设备,后者为管理和使用计算机的各种程序。

计算机每做一次动作、一个步骤,都是按照已经用计算机语言编好的程序来执行的。程序是计算机要执行的指令集合,而任何程序都是用人们所掌握的计算机语言来编写的。所以人们要控制计算机,就一定要通过计算机语言向计算机发出命令。

因此可以说,程序设计就是开发人员根据实际任务提出问题,然后确定计算方法,构造数学模型,画出流程图,选择合适的语言,编写程序并上机反复调试,直到满足实际需要的过程。而进行程序设计的基础条件是掌握计算机语言。

计算机语言的种类非常多,总的来说可以分成机器语言、汇编语言、高级语言3大类。计算机所能识别的语言只有机器语言,即由0和1构成的代码。但通常人们编程时不采用机器语言,因为它非常难以记忆和识别。

目前通用的编程语言有两种形式:汇编语言和高级语言。

汇编语言的实质与机器语言是相同的,都是直接对硬件进行操作,只不过汇编指令采用了英文缩写的标识符,这样更容易识别和记忆。它同样需要编程者将每一步具体的操作以命令的形式写出来。汇编程序的每一条指令只能对应实际操作过程中的一个很细微的动作,因此汇编程序一般比较冗长、复杂、容易出错,而且使用汇编语言编程需要有更多的计算机硬件知识。但汇编语言的优点也是显而易见的,用汇编语言所能完成的操作不是一般高级语言所能实现的,而且源程序经汇编生成的可执行文件比较小,执行速度很快。

高级语言是目前绝大多数编程者的选择。与汇编语言相比,它不但能将许多相关的机

器指令合成为单条指令，而且去掉了与具体操作有关但与完成工作无关的细节，例如如何使用堆栈、寄存器等，这样就大大地简化了程序中的指令。同时，由于省略了很多的细节，因此编程者也就不需要有很多的硬件知识。

高级语言主要是针对汇编语言而言，它并不是特指某一具体的语言，而是包括了很多的编程语言，如目前流行的 Java、Visual Basic、Visual C++、FoxPro、C 等，这些语言的语法、命令格式都各不相同。

用高级语言所编写的程序不能直接被计算机识别，必须经过转换才能被执行。

按照转换的方式，可将它们分为以下两类。

(1) 解释类：执行方式类似于“同声翻译”，应用程序源代码一边用相应语言的解释器“翻译”成目标代码(机器语言)，一边执行，因此效率比较低。而且不能生成可独立执行的可执行文件，应用程序不能脱离其解释器。但这种方式比较灵活，可以动态地调整、修改应用程序。

(2) 编译类：编译是指在应用源程序被执行之前，就将程序源代码“翻译”成目标代码(机器语言)，因此其目标程序可以脱离其语言环境独立执行，使用比较方便、效率较高。但应用程序一旦需要修改就必须先修改源代码，再重新编译生成新的目标文件(*.OBJ)才能执行。现在大多数编程语言都是编译型的，例如 C、Visual C++、FoxPro 等。

1.2 C 语言简介

C 语言的出现与 UNIX 操作系统紧密相关，C 语言本身也有一个发展的过程，表 1.1 给出了 C 语言的发展过程。

表 1.1 C 语言的发展过程

语 言 名	设 计 者	年 份
CPL	C.Strachey 等	1968
BCPL	M.Richards	1969
B	K.Thompson	1970
C	D.M.Ritchie	1972

C 语言起源于 1968 年发表的 CPL(Combined Programming Language)语言，它的许多重要思想来自于 Martin Richards 在 1969 年研制的 BCPL 语言，以及以 BCPL 语言为基础的而由 Ken Thompson 在 1970 年研制成的 B 语言，用在 PDP-7 计算机。D.M.Richie 于 1972 年在 B 的基础上研制出 C 语言，并用 C 语言写成了第一个在 PDP-11 计算机上实现的 UNIX 操作系统。1977 年出现了独立于机器的 C 语言编译文本《可移植 C 语言编译程序》，从而大大简化了把 C 语言编译程序移植到新环境所需做的工作，这本身也使 UNIX 操作系统迅速地在各种机器上实现。例如 VAX、AT&T 等计算机系统都相继开发了 UNIX。

1978 年以后，C 语言已先后移植到大、中、小、微型机上，已独立于 UNIX 和 PDP 了。现在 C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

以 1978 年发表的 UNIX 第 7 版本中的 C 编译程序为基础，Brian W.Kernighan 和 Dennis

M.Ritchie(合称 K&R)合著了影响深远的名著《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，它被称为标准 C。

1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充,制定了新的标准,称为 ANSI C。1987 年,ANSI 又公布了新标准——87 ANSI C。1990 年,国际标准化组织 ISO 接受 87 ANSI C 为 ISO C 的标准。目前广泛流行的各种版本 C 语言编译系统都是以它为基础的。本书的叙述基本上是以 ANSI C 为基础。

C 语言是以与 PASCAL 及 ALGOL 等语言相类似的思想为基础而设计出来的面向结构的程序设计语言。其主要特征如下:

- 以英文小写字母为基础。
- 程序是以函数集合构成的,函数的参数是按“值调用”方式进行。
- 程序表示简洁、紧凑,使用方便、灵活。
- 指针可以作为数据来处理。
- 运算符非常丰富,有利于编写程序。
- 没有处理字符串的特别功能,但能区分字符及字符串。
- 没有作为语言组成部分的输入输出功能,但能通过函数程序来实现。
- C 语言能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件操作。
- 生成目标代码质量高,程序执行效率高。
- C 语言写的程序可移植性好(与汇编语言比)。基本上不做修改就能用于各种型号的计算机和各种操作系统。

当然, C 语言也有缺点,至今还缺少一个国际公认的完整的语法定义,语法关系的严格性不如 FORTRAN 及 PASCAL 语言,不太适合于数值计算。

总之, C 语言对程序员要求较高。程序员使用 C 语言编写程序会感到限制少,灵活性强,功能强,可以编写出任何类型的程序。现在, C 语言已经不仅用来编写系统软件,也用来编写应用软件。学习和使用 C 的人越来越多。

1.3 C 语言的程序结构及书写规则

上一节中谈到 C 语言的特征时说到:“程序是以函数集合构成的”,也就是说,任何 C 语言程序都是由一个或几个函数所组成。

1.3.1 C 语言的程序结构

一个完整的 C 语言程序,是由一个 main()函数(又称主函数)和若干个其他函数结合而成的,或仅由一个 main()函数构成。

【例 1.1】仅由 main()函数构成的 C 语言程序。

```
main( )
{
    printf("This is a C program.\n");
}
```

程序运行结果:

This is a C program.

【例 1.2】由 main() 函数和 1 个其他函数 max() 构成的 C 语言程序。

```
int max(int m, int n)
{ return(m>n ? m : n); }
main( )
{
    int num1,num2;
    printf("Input the first integer number: ");
    scanf("%d", &num1);
    printf("Input the second integer number: ");
    scanf("%d", &num2);
    printf("max=%d\n", max(num1, num2));
}
```

程序运行情况:

```
Input the first integer number:9←↵
Input the second integer number:4←↵
max=9
```

【例 1.3】改写例 1.2, 交换 main() 函数和 max() 函数的前后位置。

(源程序略) 程序运行情况:

```
Input the first integer number:9←↵
Input the second integer number:4←↵
max=9
```

(1) 函数是 C 语言程序的基本单位。

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

(2) C 语言程序总是从 main() 函数开始执行。

一个 C 语言程序, 总是从 main() 函数开始执行, 而不论其在程序中的位置如何。当主函数执行完毕时, 亦即程序执行完毕。

习惯上, 将主函数 main() 放在最前头。

1.3.2 函数的一般结构

任何函数(包括主函数 main())都是由函数说明和函数体两部分组成。其一般结构如下:

[函数类型] 函数名(函数参数表)	← 函数说明部分
{	
声明语句部分;	
执行语句部分;	
}	

} 函数体部分

1. 使用的语法符号约定

[...] : 方括号表示可选(即可以指定, 也可以缺省)

... : 省略号表示前面的项可以重复

| : 多(含2)中选1

2. 函数说明

由函数类型(可缺省)、函数名和函数参数表三部分组成, 其中函数参数表的格式为:

数据类型 形参1 [, 数据类型 形参2...]

例如, 例 1.2 中的函数 `max()`, 其函数说明各部分如下所示。

函数类型 函数名 函数参数表

↓ ↓ ↓
int max(int x, int y)

3. 函数体

在函数说明部分的下面、大括号(必须配对使用)内的部分。

函数体一般由声明语句和可执行语句两部分构成。

(1) 声明语句部分

声明语句部分由变量定义、自定义类型定义、自定义函数声明、外部变量声明等组成。

(2) 可执行语句

一般由若干条可执行语句构成。例如下面是例 1.2 的 `main()` 函数体的示意图。

```
main( )
{
    int num1,num2;  ← 变量定义部分
    printf("Input the first integer number: ");
    scanf("%d", &num1);
    printf("Input the second integer number: ");
    scanf("%d", &num2);
    printf("max=%d\n", max(num1, num2));
}
```

} 函数体

} 执行语句部分

4. 规则

(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, 下面程序中的变量定义语句 “`int max;`” 的位置是非法的:

```
main( )
{
    int m,n;          /* 变量定义语句: 定义两个整型变量 x,y. */
    m=3,n=6;          /* 可执行的赋值语句: 将 3 赋值给变量 x, 将 6 赋值给变量 y */
    int max;          /* 变量定义出现在可执行的赋值语句 "m=3,n=6;" 之后, 为非法! */
    max=x>y?x:y;
    printf("max=%d\n",max);
}
```

解决办法很简单,请读者自己思考。

(2) 如果不需要,也可以缺省变量定义语句。

1.3.3 源程序书写格式

(1) 所有语句都必须以分号“;”结束,函数的最后一个语句也不例外。

(2) 程序行的书写格式自由,既允许一行内写几条语句,也允许一条语句分写在若干行上。例如,例 1.2 的主函数 `main()`,也可改写成如下所示的格式:

```
main( )
{
    int num1,num2;
    printf("Input the first integer number: ");scanf("%d", &num1);
    printf("Input the second integer number: ");scanf("%d", &num2);
    printf("max=%d\n", max(num1, num2));
}
```

如果某条语句很长,一般需要将其分写在几行上。

(3) 允许使用注释。C 语言的注释格式为:

```
/* ... .. */
```

例如,在例 1.1 和例 1.2 中,以及本节其他部分给出的源程序中,凡是用“/*”和“*/”括起来的文字,都是注释。注释可以出现在程序的任何部分,在简单的程序中可以没有注释。“/*”和“*/”必须成对使用,且“/”和“*”、以及“*”和“/”之间不能有空格,否则都会出错。

④ 技巧: 为避免遗漏,必须配对使用的符号,例如注释符号、函数体的起止标识符(花括号)、圆括号等等,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。在起止标识符嵌套时,以及相距较远时,这样做更有必要。

注释的位置,可以单占一行,也可以跟在语句的后面。如果一行写不下,可另起一行继续写。注释中允许使用汉字,在非中文操作系统下,看到的是一串乱码,但不影响程序的运行。

1.4 C 语言的基本符号

任何一个计算机系统所使用的字符都是固定的、有限的。它要受硬件设备的限制。C 语言与其他语言一样。它的基本符号有英文字母、阿拉伯数字以及其他特殊符号。其中:

- 英文字母有 26 个(包括大写和小写)。
- 阿拉伯数字有 10 个,即从 0 到 9。
- 其他特殊符号主要指运算符,它通常由一至两个特殊符号组成。如 +, -, *, /, %, >, <, !, |, ^, ~, -, ==, !=, >=, <=, &&, ||, ++, --, >>, <<等。

标识符用来表示函数、类型或变量的名称，它是字母和数字的排列，但必须是字母开头。“_”也称为字母，它在书写名称时很方便。在C语言中大小写字母含义不同。按照C语言的习惯，变量名用小写字母，符号常数全部用大写字母。

根据程序设计的经验，标识符的选择采用“常用从简，专用从繁”的原则。一般不必太长，通常在6个字符之内能适应各种系统。

应尽量做到“见名知意”，即选有含义的英文单词(或其缩写)或其汉语拼音作标识符，如sum, name, year, total, zongshu等。

在操作系统中，C语言源程序的文件扩展名为“.c”。因此，完整的文件名称为*.c，这样有利于编译程序对C语言源程序进行编译。

C语言与其他语言一样，有一些具有特定含义的关键字，它们不允许做为用户标识符使用，一旦使用了就会出错。

C语言的关键字共有32个，根据关键字的作用，可分其为数据类型关键字、控制语句关键字、存储类型关键字和其他关键字4类。

- 数据类型关键字(12个): char, double, enum, float, int, long, short, signed, struct, union, unsigned, void
- 控制语句关键字(12个): break, case, continue, default, do, else, for, goto, if, return, switch, while
- 存储类型关键字(4个): auto, extern, register, static
- 其他关键字(4个): const, sizeof, typedef, volatile

1.5 习 题

1. 选择题

- (1) 下列选项中，合法的C语言关键字是()。
A. VAR B. char C. integer D. default
- (2) 以下正确的叙述是()。
A. 在C语言中，main函数必须位于文件的开头
B. C语言的每行中只能写一条语句
C. C语言本身没有输入、输出语句
D. 对一个C语言程序进行编译预处理时，可检查宏定义的语法错误
- (3) 以下能定义为用户标识符的是()。
A. scanf B. void C. _3com_ D. int

2. 填空题

- (1) C语言规定，标识符只能由字母、数字和_____字符组成。
- (2) 在C语言中，一个语句_____写在多行上。
- (3) _____是C程序的基本单位。
- (4) 一个C语言的语句至少应包含一个_____。