



国家级职业教育培训规划教材

21世纪全国**高职高专**计算机应用专业规划教材

劳动保障部教材办公室组织编写

C 程序设计教程

林小茶 编著

劳动保障部推荐教材



清华大学出版社



中国劳动社会保障出版社



国家级职业教育培训规划教材

21世纪全国**高职高专**计算机应用专业规划教材

劳动保障部教材办公室组织编写

劳动保障部推荐教材

C 程序设计教程

林小茶 编著

清华大学出版社
北京

中国劳动社会保障出版社
北京

内 容 简 介

C 语言是最适合作为程序设计的入门语言之一。本书在内容的编排上,更多地考虑了初学者的要求。

本书主要内容包括:C 语言的基础知识、结构化程序设计、模块化程序设计、数组、指针、结构和文件。

全书的内容从易到难,循序渐进,列举了大量能够解决实际问题的实例,并有一个贯穿始终的例子,将一个小程序逐渐扩充成一个比较大的程序。

本书主要是为那些从来没有学过程序设计语言的高职高专的学生量身订做的,当然也可以作为 C 语言自学者的教材或参考书。

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 程序设计教程/林小茶编著. — 北京:清华大学出版社,中国劳动社会保障出版社,2005.10
(21 世纪全国高职高专计算机应用专业规划教材)
ISBN 7-302-11605-9

I. C… II. 林… 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(2005)第 092536 号

出 版 者:清华大学出版社 中国劳动社会保障出版社

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

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:张 民

印 刷 者:北京四季青印刷厂

装 订 者:三河市新茂装订有限公司

发 行 者:新华书店总店北京发行所

开 本:185×230 印张:23 字数:481 千字

版 次:2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷

书 号:ISBN 7-302-11605-9/TP·7583

印 数:1~5000

定 价:29.00 元

编 审 委 员 会

主 任：张尧学 教育部高等教育司司长

副主任：刘 康 劳动和社会保障部培训就业司副司长

陈 淮 中国劳动社会保障出版社副总编

焦金生 清华大学出版社副总编

委 员：（按姓氏笔画为序）

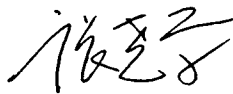
石连栓 安淑芝 沈沧海 张 民 张兴会

李 环 林 海 周 超 韩 伟 傅正泰

2002年全国职业教育工作会议指出：“推进职业教育的改革与发展是实施科教兴国战略、促进经济和社会可持续发展、提高国际竞争力的重要途径，是调整经济结构、提高劳动者素质、加快人力资源开发的必然要求，是拓宽就业渠道、促进劳动就业和再就业的重要举措。”为进一步落实全国职业教育工作会议的精神，在教育部高等教育司与劳动保障部培训就业司的共同指导与支持下，中国劳动社会保障出版社与清华大学出版社组织有关部门研究了高等职业教育（高等职业技术学院、高等专科学校、成人高等院校、高级技工学校）“计算机应用”专业的课程设置，并在此基础上启动了“21世纪全国高职高专计算机应用专业规划教材”的编写与出版工作，该套教材具有如下特点：

1. 针对性强。本套教材是为高职高专计算机应用专业的学生编写的，遵循“提出问题—解决问题”的思路，以培养计算机应用能力为主线，构造该专业的课程设置体系和教学内容体系，强调理论教学与实验实训密切结合，尤其突出实训环节的教学。
2. 配套出版辅助教材。编写出版主教材的同时，本套教材还配套出版相应的《实训》，旨在指导学生通过大量的实际训练，更好地掌握教程的内容，从而进一步提高学生在计算机各个方面的应用能力，突出职业教育的特色。
3. 版本更新及时。将紧跟科学技术的新发展和高职高专教育的新形势，不断推出新教材，及时修订更新教材内容。
4. 与考试认证、岗位培训等实际应用紧密结合。在体现自身特色的同时，尽量兼容目前的计算机考试辅导和岗位准入培训的要求。目前可以考虑兼容的有“全国计算机等级考试”、“高技能人才培训”、“高职院校毕业生资格职业培训”等，同时除了部分理论性较强的科目以外，该丛书的部分教材还可以用于非学历教育（含社会培训、职工岗前培训等）。

相信这套教材的编写和出版对进一步推动学校教育 with 职前培训的结合，促进高职高专的教学和教材改革，以及探索高等职业教育的新的思路等会有很好的促进作用。



教育部高等教育司司长

出版说明

我国高等职业技术教育是社会经济发展对职业教育提出的更高层次的要求,是中等职业教育的继续和发展。为了进一步适应经济发展对高等技术应用型和技能操作型人才的需求,国家正在理顺高等职业教育、高等专科教育和成人高等教育三者的关系,统称为高职高专教育,力求形成合力,将目标统一到培养高等技术应用型和技能操作型人才上来。

为了贯彻落实党中央、国务院关于大力发展高等职业教育、培养高等技术应用型和技能操作型人才的指导精神,解决高等职业教育缺乏通用教材的问题,劳动和社会保障部教材办公室从1999年下半年开始,组织部分高校编写了“21世纪全国高职高专专业教材”。这套教材具有三大特点:①为高等职业教育、高等专科教育和成人高等教育“三教”的整合与升级服务;②体现高职高专教育以培养高等技术应用型和技能操作型人才为宗旨,使学生获得相应职业领域的职业能力;③以专业教材为主,突出以应用技术、创造性技能和专业理论相结合为特色。目前我们已出版的高职高专专业教材有机械类、电工类和医学美容、汽车检测与维修、国际贸易、建筑装饰、物业管理等专业的教材,与教育部高教司合作开发、即将出版的计算机应用专业规划教材,以及正在陆续开发的电子商务、机电一体化、数控技术等几十个专业的教材。力争逐步建立起涵盖高职高专各主要专业,符合市场要求,满足经济建设需要的高职高专院校专业教材体系。

在本套教材的编写工作中,我们注意了以下两点:一是目标明确。立足于高等技术应用类型的专业,以培养生产建设、三产服务、经营管理第一线的高等职业技术应用型和技能操作型人才为根本任务,以适应经济建设的需求。二是突出特色。教材以国家职业标准为依据,以培养技术应用能力为主线,全面设计学生的知识、职业能力和培养方案,以“适用、管用、够用”为原则,从职业分析入手,根据职业岗位群所需的知识结构来确定教材的具体内容,在基础理论适度的前提下,突出其职业教育的功能,力争达到理论与实践的完美结合,知识与应用的有机统一,以保证高职高专教育目标的顺利实现。

编写这套适用于全国高职高专教育有关专业的教材既是一项开创性工作,又是一项系

统工程，参与编写这套系列专业教材的各有关院校的专家和教师为此付出了艰辛的努力，谨向他们表示衷心的感谢。同时由于缺乏经验，这套教材难免存在某些缺点和不足，在此，我们恳切希望广大读者提出宝贵意见和建议，以便今后修订并逐步完善。

劳动和社会保障部教材办公室

前言

C 语言作为程序设计的入门语言已经有相当长的历史了,之所以选其为入门语言,除了 C 语言的众多优点外,最主要的还是 C 语言的实用性。众所周知的操作系统 UNIX, MS-DOS, Microsoft Windows 及 Linux 等都是用 C 语言编写的。C 语言具有高效、灵活、功能丰富、表达力强和移植性好等特点。

C 语言是程序设计的工具,因此学会使用 C 语言并不是学习的惟一目的,掌握计算机处理问题的思维方式和程序设计的基本方法,用以解决实际问题其实更为重要。因此,本书将会花一些篇幅讨论程序设计的基本原理、概念和方法,并通过实例来巩固所学的知识。

本书主要包括:C 语言的基础知识、结构化程序设计、模块化程序设计、数组、指针、结构和文件。在内容的编排上主要有下面几点考虑:

第一,不要求在内容上面面俱到。例如,在 C 语言中,使用文本方式输入输出数据的主要目的是为了调试程序,而不是为了编写一个像 Windows 那样的可视环境,因此,有关这些函数的使用细节不在讨论的范围,但是为了方便读者,增加了一个附录 C,即“printf 函数的转换说明模式”;又例如链表是“数据结构”课程的教学内容,所以本书中将不大量地介绍与链表相关的程序,并且这部分内容对初学者有一定的难度;使用位操作进行程序设计对初学者不适合,因此只对位运算符做了简单介绍,并没有像很多教材一样将位操作作为一章的内容来讨论。

第二,强调如何编写好的程序。在本书的很多地方强调要努力编写一个好的程序,而不要花心思在一些小的程序设计技巧上。例如,本书在分析增 1/减 1 运算符的副作用时,告诫学习者应避免使用像 $i++++i$ 这样的表达式,而不是花大量的篇幅去分析这个表达式到底等于多少,因为真正编写程序的人是不会这样用的,使用增 1/减 1 运算符的最好方法是使用执行语句“ $i++;$ ”。类似地,本书在介绍运算符的优先级和结合性时也提出,使用括号表示运算顺序是最好的方法,而不要求学习者去背诵每个运算符的优先级。

第三,本书编写了大量的程序实例,试图通过这些实例帮助读者更好地掌握所学知识,相当多的实例都是经过精心设计和考虑的,并对那些稍微复杂的程序都尽量做了比较详尽的注释。本书中有一个例子随着学习内容的深入,不断地被扩充功能,最后的程序几乎用

到了所有学过的基本数据类型和构造数据类型。

本书还配有习题解答和实训教程，给出本教材中所有习题的参考答案，供读者学习时借鉴和参考。

编者水平有限，错误在所难免，敬请广大读者批评指正。

作 者

目 录

第 1 章 C 语言概述	1
1.1 程序设计语言	1
1.1.1 低级语言	1
1.1.2 高级语言	2
1.2 通过实例认识 C 程序的结构	3
1.2.1 问候界面	3
1.2.2 计算两个整数的差	5
1.2.3 使用自定义函数计算两个整数的差	7
1.3 C 语言的标准和编译器	8
1.3.1 C 语言的标准	8
1.3.2 常用的 C 语言编译器	9
1.4 C 程序的调试	9
1.4.1 调试步骤	9
1.4.2 Turbo C 集成开发环境	10
习题	14
第 2 章 C 语言的基础知识	15
2.1 标识符、变量与常量	15
2.1.1 标识符	15
2.1.2 变量	17
2.1.3 常量	19
2.2 C 语言的数据类型	19
2.2.1 C 语言的数据类型种类	19
2.2.2 问题的提出	21

2.2.3	整型数据	21
2.2.4	字符型数据	30
2.2.5	浮点数据	36
2.3	运算符和表达式	39
2.3.1	表达式与简单语句	39
2.3.2	算术运算符	40
2.3.3	赋值运算符	42
2.3.4	增 1/减 1 运算符	43
2.3.5	位逻辑运算符	44
2.3.6	逗号运算符	47
2.3.7	求字节数运算符	47
2.3.8	不同数据类型数据间的混合运算	49
2.3.9	赋值表达式的类型转换	51
2.4	指针类型与指针运算符	57
2.4.1	指针概念和指针变量的定义	57
2.4.2	指针运算符 & 和 * 的使用	58
习题		62
第 3 章	结构化程序设计	67
3.1	结构化程序设计思想	67
3.1.1	结构化程序设计思想的产生	67
3.1.2	结构化程序设计的 3 种基本结构	69
3.2	语句与分程序	73
3.3	顺序结构程序设计	75
3.4	关系运算符与逻辑运算符	76
3.4.1	关系运算符	77
3.4.2	逻辑运算符	78
3.5	选择结构程序设计	80
3.5.1	问题的提出	80
3.5.2	if 语句的 3 种形式	81
3.5.3	嵌套的 if 语句	94
3.5.4	switch 语句	98
3.5.5	条件运算符	105
3.5.6	选择结构程序举例	107

3.6 循环结构程序设计	111
3.6.1 问题的提出	111
3.6.2 while 语句	112
3.6.3 do while 语句	117
3.6.4 for 语句	120
3.6.5 多重循环	123
3.6.6 break 语句在循环语句中的用法	126
3.6.7 continue 语句	128
3.6.8 循环结构程序举例	131
习题	140
第4章 模块化程序设计	151
4.1 模块化程序设计思想	151
4.2 函数的定义、说明与调用	152
4.2.1 实例	152
4.2.2 函数的定义形式	154
4.2.3 函数的返回值	156
4.2.4 函数说明	157
4.2.5 函数调用	159
4.3 函数的参数传递	162
4.3.1 形参和实参的关系	162
4.3.2 普通变量作为函数的形式参数	163
4.3.3 指针变量作为函数的形式参数	164
4.4 函数程序举例	167
4.5 函数的递归调用	170
4.6 变量的存储类别	173
4.6.1 自动变量与外部变量	174
4.6.2 静态变量	183
4.6.3 寄存器变量	186
4.7 预处理命令	186
4.7.1 宏定义	187
4.7.2 文件包含	193
4.7.3 条件编译	195
习题	197

第 5 章 数组和指针	205
5.1 一维数组	205
5.1.1 问题的提出	205
5.1.2 一维数组的定义	205
5.1.3 一维数组的引用	207
5.1.4 一维数组的初始化	209
5.1.5 程序举例	210
5.1.6 数组名作为函数的参数	213
5.2 指针与一维数组	222
5.2.1 指针值的算术运算	222
5.2.2 指针方式和数组方式对数组元素的操作	227
5.2.3 指向一组空间首地址的指针作为函数参数	228
5.3 动态的一维数组	230
5.3.1 空指针	230
5.3.2 存储器申请与释放	230
5.3.3 动态数组的使用	233
5.4 字符串与字符串函数	234
5.4.1 字符数组	235
5.4.2 字符串变量	235
5.4.3 字符串变量的输入与输出	236
5.4.4 指针与字符串	239
5.4.5 程序举例	241
5.4.6 字符串函数	244
5.5 二级指针	246
5.6 指针数组与命令行参数	249
5.6.1 指针数组	249
5.6.2 命令行参数	252
5.7 二维数组	254
5.7.1 二维数组的定义	255
5.7.2 二维数组的引用	255
5.7.3 二维数组的初始化	257
5.7.4 程序举例	258
5.7.5 用指针方法操作二维数组	263
习题	265



第 6 章 结构体等构造数据类型	275
6.1 结构体	275
6.1.1 问题的提出	275
6.1.2 结构体的说明	276
6.1.3 结构体变量的定义	278
6.1.4 结构体成员的引用	279
6.1.5 结构体的初始化	281
6.2 结构体与数组	282
6.2.1 结构体包含数组	282
6.2.2 结构体数组	283
6.3 结构体与指针	285
6.3.1 指向结构体的指针	285
6.3.2 用结构体类型指针建立链表	288
6.4 结构体与函数	291
6.4.1 结构体数据作为函数的参数	291
6.4.2 返回指向结构体的指针的函数	292
6.5 联合体与枚举	294
6.5.1 使用联合体与枚举的目的	294
6.5.2 联合体与枚举的说明	294
6.5.3 联合体变量与枚举变量的定义	295
6.5.4 联合体变量成员的引用	297
6.5.5 枚举变量的使用	298
6.5.6 指向联合体变量的指针	299
6.5.7 联合体变量与函数	300
6.5.8 使用联合体与枚举的程序举例	300
6.6 类型定义	303
6.7 程序举例	304
习题	307
第 7 章 文件	314
7.1 文件概述	314
7.1.1 问题的提出	314
7.1.2 文件“流”	314
7.1.3 文件操作的特点	315



7.1.4	缓冲文件系统	316
7.2	文件的打开与关闭	316
7.2.1	文件类型指针	316
7.2.2	文件的打开	317
7.2.3	文件的关闭	319
7.3	文件的读写操作	319
7.3.1	fputc 函数与 fgetc 函数	320
7.3.2	fprintf 函数与 fscanf 函数	325
7.3.3	fread 函数与 fwrite 函数	329
7.3.4	fgets 函数与 fputs 函数	334
7.4	文件的定位操作	334
7.4.1	文件的顺序存取和随机存取	334
7.4.2	rewind 函数	335
7.4.3	fseek 函数	336
7.4.4	ftell 函数和 feof 函数	337
	习题	338
附录 A	ASCII 代码与字符对照表	341
附录 B	运算符的优先级和结合性	344
附录 C	printf 函数的转换说明模式	346

第 1 章

C 语言概述

C 语言是一种通用的程序设计语言,由于其功能非常强大,因此可以用来完成一些非常复杂的工作。很多操作系统都是用 C 编写的,例如 UNIX,MS-DOS,Microsoft Windows 及 Linux 等。C 语言具有高效、灵活、功能丰富、表达力强和移植性好等特点。

1.1 程序设计语言

语言是人际交流的重要工具之一,人们用语言可以准确地表达自己的思想,达到互相沟通的目的。同样,人与计算机的交流也需要使用一种语言,这就是程序设计语言。程序设计语言是一组用来定义计算机程序的语法规则。

计算机程序是由计算机指令构成的序列。计算机按照程序中的指令逐条执行,就可以完成相应的操作。程序员使用程序设计语言定义计算机完成特定任务所需要使用的数据和计算机的动作。计算机程序的操作对象是“数据”。

计算机自己不会做任何工作,它所做的工作都是由人们事先编好的程序来控制的。程序需要人来编写,使用的工具就是程序设计语言。目前,通用的计算机还没有识别自然语言的能力,而只能识别特定的计算机程序设计语言。

一般情况下,计算机程序设计语言分为两类,一类是低级语言,另一类是高级语言。

低级语言的主要特点是直接依赖计算机的硬件系统,一种机型能识别的低级语言,另一种机型可能完全不能识别。而高级语言与低级语言不同,它不再依赖计算机的硬件系统,用高级语言编写的程序几乎可以不加修改地运行在不同机型的计算机上。

需要强调的是,无论采用何种语言来编写程序,程序在计算机上的执行都是由 CPU 所提供的机器指令来完成的。机器指令是用二进制表示的指令集。每种类型的 CPU 都有与之对应的指令集。

1.1.1 低级语言

低级语言包括机器语言和汇编语言。

直接使用二进制 0 和 1 表示的指令序列来编程的语言就是机器语言。使用机器语言

编写的程序,都是用 0 和 1 表示的,例如,“101110001110100000000011”的功能是将 1000 送入寄存器 AX 中。使用机器语言编写程序,对程序员来说,不仅需要高超的技巧,还要特别心细,哪一位错了都不行,同时,程序员必须准确无误地牢记每一条指令的二进制编码。

机器语言的优点是执行速度快,并且可以直接对硬件进行操作,例如主板上的 BIOS,还可以编写一些设备的驱动程序等。

机器语言的缺点也是显而易见的。首先是可读性差,就是编写程序语句“101110001110100000000011”的人也未必马上就能看懂该句表示的是什么命令;其次是可维护性差,用机器语言编写的程序很难看懂,又如何谈维护呢?最后就是可移植性差,因为不同的机型有自己的一套机器指令,与其他机型的机器指令不兼容。另外,用机器语言编写程序的生产效率低,而且不能保证程序有好的质量。

为了方便记忆和编程,人们用一些符号和简单的语法来表示机器指令,这就是汇编语言。汇编语言是特定机器的机器指令的助记符,它依赖于特定的 CPU 体系结构。例如,“101110001110100000000011”用汇编语言表示就是“mov ax,1000”。用“mov ax,1000”表示“将 1000 送入寄存器 AX 中”,从中可看出可读性要好于机器指令。但是 CPU 并不能识别汇编语言,因此,需要一个“翻译”程序将汇编语言翻译成机器语言,我们把这种将汇编语言翻译成机器语言的程序叫做“汇编器”。汇编语言与机器语言的指令是一一对应的,所以,除了提高了一些可读性,汇编语言从根本上并没有改变机器语言的特点。因此可以说,汇编语言是面向机器语言的。当然,汇编语言仍然具备机器语言的优点。许多大型系统的核心部分都是用汇编语言编写的,因为这部分工作需要很高的效率,直接和硬件打交道。

由于用低级语言编写的程序不具备良好的可读性、可维护性和可移植性,因此人们发明了高级语言。

1.1.2 高级语言

高级语言是一种比较接近自然语言和数学语言的程序设计语言。高级语言非常符合人类的逻辑思维,抽象程度大大提高,高级语言的出现大大提高了程序员的工作效率,降低了程序设计的难度,并改善了程序的质量。用高级语言编写的程序看起来更像是英语,很容易读懂,不但使程序具备良好的可读性和可维护性,而且使更多的人掌握了程序设计方法,从而使计算机技术得到迅速的应用和普及。

例如,语句段

```
if (u >= v)
    max = u;
```