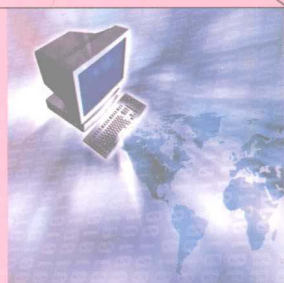




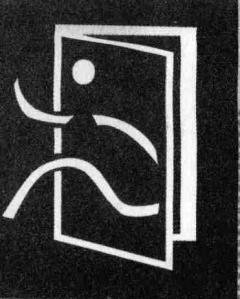
普通高等学校非计算机专业计算机基础系列教材



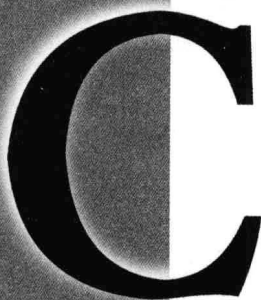
C 语言程序设计

◆ 主 编 龙昭华
◆ 副主编 纪 钢

重庆大学出版社



普通高等学校非计算机专业计算机基础系列教材



语言程序设计

主 编 龙昭华
副主编 纪 钢
编 者 丁明勇 龙昭华
纪 钢 杨芳明

重庆大学出版社

内容简介

C 语言是在各种计算机上运行较为流行的一种高级语言,应用十分广泛。本教材在内容安排及编写上,力求做到概念清楚、内容完整、难易适中,一方面讲授 C 语言的基础知识,另一方面在例题的讲解、习题的选择上尽量接近 C 语言二级等级考试的难度。

本书在内容安排上将指针的概念及使用融合在数组和函数中;在程序设计的章节中,为读者展示了 C 语言的编程技巧,这是本书具有的特点之一。本书适合作为高等院校非计算机专业计算机程序设计类课程的教材。

图书在版编目(CIP)数据

C 语言程序设计/龙昭华主编. —重庆:重庆大学出版社,2004.2

(普通高等学校非计算机专业计算机基础教材系列教材)

ISBN 7-5624-2993-6

I. C... II. 龙... III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 003734 号

普通高等学校非计算机专业计算机基础教材系列教材

C 语言程序设计

主 编 龙昭华

副主编 纪 钢

责任编辑:王海琼 王 勇 版式设计:吴庆渝

责任校对:廖应碧 责任印制:张立全

*

重庆大学出版社出版发行

出版人:张鸽盛

社址:重庆市沙坪坝正街 174 号重庆大学(A 区)内

邮编:400030

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn (市场营销部)

全国新华书店经销

重庆华林天美彩色报刊印务有限公司印装

*

开本:787×1092 1/16 印张:20.75 字数:466 千

2004 年 2 月第 1 版 2004 年 2 月第 1 次印刷

印数:1—5 000

ISBN 7-5624-2993-6/TP·431 定价:27.00 元

本书如有印刷、装订等质量问题,本社负责调换

版权所有 翻印必究

编委会

顾 问 吴中福 邱玉辉

主 任 陈流汀

副主任 杨天怡 严欣平 张鸽盛

委 员 (以姓氏笔画为序)

王世迪 邓亚平 程小平

杨国才 范幸义 洪汝渝

郭松涛 黄 勤 曾 一

谭世语 熊 壮 莫 堃



序言

计算机技术的飞速发展,加快了人类进入信息社会的步伐,改变了世界,改变了人们的工作、学习和生活,对社会发展产生了广泛而深远的影响。计算机技术在其他各学科中的应用,极大地促进了各学科的发展。不掌握计算机技术,就无法掌握最先进、最有效的研究开发手段,将影响到其所从事学科的发展。因此,计算机技术基础是 21 世纪高校非计算机专业大学生必须掌握的、最重要的基础之一。

经过多年的探索和实践,按“计算机文化基础”、“计算机技术基础”、“计算机应用基础”三个层次组织教学已被公认为高校非计算机专业计算机基础教学的基本模式。第一层次开设“计算机文化基础”课程,教学的主要任务是使学生掌握计算机基础知识和基本操作能力;第二层次开设“计算机软件技术基础”和“计算机硬件技术基础”课程,教学的主要任务是使学生掌握计算机软、硬件技术的基本知识和基本开发技术;第三层次按专业群开设“计算机信息管理基础”、“计算机辅助设计基础”、“计算机网络技术基础”、“计算机控制技术基础”等不同课程,教学的主要任务是培养学生应用计算机技术分析解决本学科及相关领域问题的能力。

为了适应计算机技术的飞速发展和广泛应用对高校非计算机专业人才培养提出的新要求,我们组织一批

长期从事计算机技术教学和科研的教师,编写了这套计算机基础教学系列教材。本系列教材有如下特点:

1. 适合于层次教学模式。系列教材内容覆盖了高校各类非计算机专业三层次计算机基础教学要求,既有适合理工类专业使用的,也有适合文经类专业使用的,各类专业都可从中选择到相应的教材。

2. 内容新。系列教材较好地反映了计算机技术的新发展,如《计算机文化基础》介绍了图形窗口界面和网络、多媒体基础;《计算机软件技术基础》介绍了软件基本概念和基本工具、结构化及面向对象程序设计的概念与方法、软件工程的基本思想和最先进的开发环境及平台;《计算机信息管理基础》以大型关系数据库管理系统为背景,介绍了关系数据库的基本知识和数据的构造方法以及网络技术在系统中的应用等等。

3. 强调应用和实用。非计算机专业的计算机基础教学以应用为目的,因此,本系列教材在编写上特别注意应用需要,强调实用性。主要课程教材都配有实验教程,基本知识理论讲深讲透,使用技术主要通过学生上机实验来掌握。

4. 便于自学。为了充分调动学生的学习主动性和能动性,本系列教材在写法上,既注意概念的严谨与清晰,又特别注意用易读易懂的方法阐述问题,应用举例丰富,便于自学。

总而言之,本系列教材的编写指导思想是:内容要新,要体现计算机技术的新发展和适应教学改革的要求;概念要清晰、通俗易懂,便于学生自学;应用性、实用性要强,切实在培养学生应用能力上下功夫;层次配套,可选择性强,适用面宽,既是普通高校非计算机专业本专科学生教材,亦可作为高等教育自学教材和工程技术人员的参考书。

限于编者水平,系列教材的内容及体系难免有缺点错误,诚恳希望读者和专家给予指正。

编委会
2002年1月



前言

不管是计算机专业或非计算机专业人员,学习计算机都应该学习程序设计。学习程序设计,必然要结合一门计算机语言来进行。C 语言是当今最有生命力的高级程序设计语言之一,它具有简洁、表达能力强、可移植性强、产生的目标代码效率高等优点。近年来,高等院校有关专业均陆续开设了这门程序设计基础课程。由于 C 语言具有较强的低级语言的特点,可代替部分汇编程序设计语言进行系统软件的开发;同时,它作为高级语言,也是向支持面向对象程序设计范型语言——C++ 过渡的桥梁,因而受到程序设计人员的青睐。

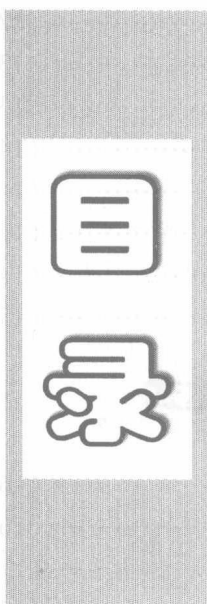
本书定位为学习 C 语言的基础知识,其广度和深度大体相当于计算机等级考试二级水平,具有较强的针对性和实用性。学习完本书后,读者应能掌握使用 C 语言的基本知识,通过该语言进行相应的程序设计并为参加计算机等级考试做好准备。学习 C 语言有一定难度,鉴于此,本书力图将抽象的 C 语言知识点简单化、规范化。程序设计是细致的活动,主要目的是借助于程序语言,按一定规则来描述问题。对于初学者,不宜一味追求多学几门程序设计语言,只要精通学好一门程序设计语言,在这基础上进行扩展,就能逐步掌握计算机软件设计技术。

本书由龙昭华主编,熊壮老师主审。第1章、第2章、第9章由龙昭华编写;第3章由杨芳明编写;第4章、第5章由纪钢编写;第6章、第7章、第8章由丁明勇编写。周建丽、徐子珊、曾立梅、聂永萍、蒋贵全等参加了本书部分策划、编录、整理等工作。

本书第9章“编程技巧”是在学习了C的基础知识基础上为激发学生用C语言开发软件的兴趣而编写,带提高性质;本章可由教师根据课程进度选择讲解,也可供学生自学。

本书编者都是长期从事大学计算机教学的教师,相信比较切合教学实际,但难免有疏漏和错误,恳请读者批评指正。

编 者
2004 年 1 月



1 程序设计与软件开发基础概述

1.1 计算机语言	(1)
1.1.1 计算机语言	(1)
1.1.2 编译程序和解释程序	(2)
1.2 算法与数据结构	(3)
1.2.1 数据结构的组织方法和操作方法	(3)
1.2.2 算法及其描述	(4)
1.2.3 软件的概念与分类	(4)
1.3 结构化程序设计的3种基本结构及其设计方法	(5)
1.3.1 3种基本结构	(5)
1.3.2 算法的表示	(6)
1.4 软件工程	(8)
1.4.1 软件危机产生原因及其表现	(8)
1.4.2 软件工程概念	(8)
1.4.3 软件工具及其软件开发环境概念	(9)
1.4.4 软件生命周期及各阶段的主要任务	(9)
1.4.5 几种常见软件开发模型	(10)
1.5 C程序的特点	(11)
1.5.1 函数	(11)
1.5.2 函数简单介绍	(11)
1.5.3 C语句分类	(11)

1.5.4 C 语言的输出/输入	(12)
1.5.5 C 语言程序书写格式	(13)
1.5.6 C 语言特点	(13)
1.6 常用 C 语言开发工具简介	(14)
1.7 C 语言的关键字、标识符	(14)
1.7.1 关键字	(14)
1.7.2 标识符	(14)
习题 1	(15)

2 C 语言的数据类型、运算符及其表达式

2.1 C 语言的数据类型	(16)
2.1.1 基本类型	(16)
2.1.2 构造类型	(17)
2.1.3 空类型	(17)
2.1.4 指针类型	(17)
2.2 常量及其类型	(17)
2.2.1 常量的概念	(17)
2.2.2 常量分类	(18)
2.3 变量及其类型	(20)
2.3.1 变量的含义	(20)
2.3.2 3 种基本数据类型变量	(21)
2.3.3 指向基本数据类型变量的指针变量	(25)
2.4 运算符和表达式	(25)
2.4.1 C 运算符简介	(25)
2.4.2 C 表达式概念	(26)
2.4.3 算术运算符和算术表达式	(26)
2.4.4 自动类型转换和强制类型转换	(27)
2.4.5 自增、自减运算符	(29)
2.4.6 赋值运算符与赋值表达式	(30)
2.4.7 逗号运算符和逗号表达式	(31)
2.4.8 关系运算符和关系表达式	(32)
2.4.9 逻辑运算符和逻辑表达式	(32)
2.4.10 取地址运算符和指针运算符	(33)
2.4.11 类型长度运算符	(34)
2.4.12 位运算符	(35)
习题 2	(37)



3 结构化程序设计

3.1 预定义输出/输入函数	(40)
3.1.1 格式化输出/输入函数	(40)
3.1.2 字符数据的输出/输入函数	(47)
3.2 顺序结构程序设计举例	(48)
3.3 选择结构程序设计	(50)
3.3.1 if 语句	(51)
3.3.2 条件运算符	(60)
3.3.3 switch 语句	(61)
3.3.4 选择结构程序设计举例	(65)
3.4 循环结构程序设计	(68)
3.4.1 4 种循环结构	(69)
3.4.2 break 与 continue 语句	(84)
3.4.3 循环结构程序设计举例	(86)
习题 3	(93)

4 指针与数组

4.1 一维数组	(98)
4.1.1 一维数组的定义	(99)
4.1.2 数组元素的引用	(99)
4.1.3 一维数组初始化	(100)
4.1.4 指向一维数组元素的指针变量	(101)
4.1.5 程序举例	(108)
4.2 二维数组	(110)
4.2.1 二维数组的定义	(110)
4.2.2 二维数组的存储方式	(110)
4.2.3 数组元素的引用	(111)
4.2.4 二维数组初始化	(112)
4.2.5 指向二维数组元素的指针变量	(113)
4.2.6 指向由 m 个元素构成的一维数组的指针变量	(115)
4.2.7 程序举例	(116)
4.3 字符数组与字符串	(118)
4.3.1 字符数组	(118)
4.3.2 字符串与字符数组的联系	(120)
4.3.3 常用字符串处理预定义函数	(123)

4.4 指针数组	(127)
4.4.1 指针数组的定义	(127)
4.4.2 指针数组的使用	(127)
4.4.3 指针数组与多级指针	(129)
习题4	(130)

5 函 数

5.1 函数的概念	(135)
5.1.1 函数的分类	(135)
5.1.2 库函数的特点	(136)
5.1.3 用户自定义函数的特点	(136)
5.1.4 函数返回值及参数传递问题	(137)
5.1.5 函数的定义形式	(137)
5.2 函数参数与函数值	(138)
5.2.1 函数参数	(138)
5.2.2 函数值	(141)
5.3 函数的调用	(143)
5.3.1 函数调用的形式	(143)
5.3.2 函数说明	(143)
5.3.3 函数调用方式	(144)
5.4 指向函数的指针及常用的几种参数传递方式	(145)
5.4.1 指向函数的指针	(145)
5.4.2 常用的几种参数传递方式	(148)
5.5 函数的嵌套调用与函数的递归调用	(153)
5.5.1 函数的嵌套调用	(153)
5.5.2 函数的递归调用	(154)
5.6 主函数带参数	(157)
5.7 局部变量和全局变量	(158)
5.7.1 局部变量	(158)
5.7.2 全局变量	(159)
5.8 变量的存储类别	(162)
5.8.1 自动变量	(162)
5.8.2 静态变量	(162)
5.8.3 寄存器变量	(163)
5.8.4 外部变量	(164)
习题5	(164)



6 编译预处理

6.1 编译预处理的特念和特点	(170)
6.1.1 编译预处理的特念	(170)
6.1.2 编译预处理的特念	(170)
6.2 宏定义	(171)
6.2.1 不带参数的宏定义	(171)
6.2.2 带参数的宏定义	(174)
6.3 文件包含及其应用	(179)
6.4 条件编译	(180)
习题 6	(182)

7 结构体、共用体、枚举类型

7.1 结构体	(186)
7.1.1 结构体类型定义	(186)
7.1.2 结构体变量定义	(187)
7.1.3 结构体变量的初始化	(188)
7.1.4 结构体变量的引用(结构体成员运算符)	(189)
7.1.5 指向结构体变量的指针变量	(190)
7.1.6 位段	(192)
7.1.7 结构体数组的定义	(193)
7.1.8 结构体数组的初始化	(194)
7.1.9 结构体数组的引用	(194)
7.1.10 指向结构体数组的指针变量(-> 运算符)	(199)
7.1.11 结构体与函数	(200)
7.1.12 单链表	(200)
7.2 共用体	(209)
7.2.1 共用体类型定义	(210)
7.2.2 共用体变量定义	(211)
7.2.3 共用体的引用	(211)
7.2.4 共用体的特念	(212)
7.3 枚举类型	(214)
7.3.1 枚举类型定义	(214)
7.3.2 枚举变量定义	(215)
7.3.3 枚举变量的值	(215)
7.4 复合数据类型(typedef)	(216)

习题 7	(216)
------------	-------

8 文 件

8.1 文件的基本概念	(223)
8.1.1 文件的概念	(223)
8.1.2 文件系统	(224)
8.1.3 文件分类	(225)
8.1.4 文件指针	(226)
8.2 文件操作的相关函数	(227)
8.2.1 文件的打开与关闭	(227)
8.2.2 文件的读/写(输入/输出)	(228)
8.2.3 文件的定位	(239)
习题 8	(241)

*9 编程技巧

9.1 图形程序设计基础	(248)
9.1.1 文本窗口	(248)
9.1.2 图形程序设计	(252)
9.2 屏幕界面设计	(256)
9.2.1 利用 BIOS 中断显示彩色汉字	(256)
9.2.2 利用 C 语言的库函数显示汉字	(258)
9.2.3 EGA/VGA 屏幕存储与恢复	(260)
9.2.4 中文状态下利用 Turbo C 的图形功能	(263)
9.2.5 设计立体投影窗口	(264)
9.3 其他应用技巧	(265)
9.3.1 用 C 语言放大汉字	(265)
9.3.2 用 C 语言开发音乐程序	(270)
9.3.3 利用键盘作图	(275)
9.3.4 动画设计	(281)
9.3.5 鼠标在程序设计中的应用	(284)
习题 9	(295)

附 录

附录一 ASCII 表	(296)
附录二 运算符表	(297)



附录三 常用函数表	(299)
附录四 C 语言笔试模拟试题	(304)

参考文献

1

程序设计与软件开发基础概述

1.1 计算机语言

1.1.1 计算机语言

存储程序原理是计算机工作的基本原理,程序是它的核心,而编写程序的工具是语言,因此了解程序设计语言具有重要的意义。计算机的历史有多久,计算机语言的发展就有多久。总的来说,按其历史时期来分,可分为5代语言;每进一代,其描述问题的方法越来越贴近人类的思维习惯,功能越来越完善,但表达形式却越来越简单。

第1代语言 又称“机器语言”,是计算机诞生和发展初期使用的语言,表现为二进制编码形式,由CPU可以识别的0、1序列构成的指令码。用机器语言编写程序极为复杂,要求使用者必须熟悉计算机的所有细节,程序的质量完全取决于个人的编程水平。随着计算机的硬件结构复杂化,指令系统变得非常庞大,一般的工程技术人员已经很难掌握。

第2代语言 即“汇编语言”,它开始于20世纪50年代初,用助记符来表示每一条机器指令,比机器语言进了一步,但其大部分指令还是和机器语言一一对应,用汇编语言编写的程序要翻译成机器语言后才能执行,呈现在用户面前的是装有汇编程序的计算机。

第3代语言 即“高级语言”,它起源于20世纪50年代中期,与人们的自然语言和数学语言更接近,可读性强,编程方便。用一种高级语言编辑的源程序,可以在具有该种语言编译系统的不同计算机上使用,但它也必须编译或解释成机器语言才能

执行。BASIC、FORTRAN、COBOL、PASCAL、C 等都属于第 3 代语言。

第 4 代语言 称为“非过程化语言”，顾名思义，用户使用这种语言，不必关心问题的解法和处理过程的细节描述，只要说明所要完成的加工和条件，指明输入数据和输出形式，就能得到需要的结果，其他工作由系统来完成。第 3 代语言告诉计算机怎么做，第 4 代语言告诉计算机做什么，因此，第 4 代语言又称为“面向对象的语言”。

第 5 代语言 堪称智能性语言，除具有第 4 代语言的特点外，还具备许多新的功能，尤其是具有一定的智能性。PROLOG 语言是第 5 代语言的代表，它广泛运用于抽象问题求解、数据逻辑、公式处理、自然语言理解、专家系统和人工智能的许多领域。

在 20 世纪 60 年代这 10 年里，出现了 200 多种高级语言，20 世纪 70 年代以来，随着结构化程序设计思想的日益深入，这段时间里问世的几种程序设计语言（如 PASCAL、Ada、C）的控制结构大大简化，20 世纪 80 年代提出“面向对象”的概念是相对于“面向过程”的一次革命。C++ 便是面向对象的程序设计语言，面向对象技术在系统程序设计、数据库及多媒体运用等诸多领域得到广泛运用。这些语言与人们的自然语言比较接近，大大提高了程序设计的效率，便于人们用这类语言进行交流。

1.1.2 编译程序和解释程序

今天的计算机仍然只能够理解和执行机器语言。程序语言的引入意味着：必须有一个程序，其任务是使机器能够理解用某一种语法规则书写的用户语句，担负这一任务的程序称为“语言处理程序”，它可以分为两大类：解释程序和翻译程序。

- 解释程序 解释程序是把编译和运行过程合为一体。这种方式并不形成目标程序，而是按目标程序的动态顺序逐句解释并立即执行程序。在此过程中，输入的是源程序和原始数据，输出为目标程序的计算结果而非目标程序。

解释程序的实际工作很复杂，通常解释程序对源程序进行词法、语法检查，源程序换码变成便于处理的内部形式，建立一些表格，如符号表、标号表等。进行错误检查后，下一阶段的任务是用前一阶段提供的表格，对内部形式的源程序解释并执行。

- 编译程序 用高级语言编写的源程序可通过编译器翻译为语意上等价的目標程序，由机器语言、汇编语言和某些中间语言表示。如目标程序是用汇编语言编成的，要先用汇编语言程序把它编成机器语言程序再执行；但如果是用某些中间语言编程，则应先由解释器解释执行或再经编译后执行。

高级语言程序运行需要执行 3 个步骤：编译阶段、连接装配阶段和运行阶段。编译阶段由编译程序扫描源程序并翻译为目标程序。编译程序输出是待装配的目标程序模块，需要先经过连接装配，把目标程序及必须的运行子程序连接形成可执行的机器代码。

编译阶段一般分为 5 个子阶段：词法分析、语法分析、语义分析和中间代码生成、代码优化、目标代码生成。除此以外还有两部分工作：建表和查表工作，进行出错处理。

在编译和解释这两种方式中，解释器比编译器灵活，解释器执行源程序时有对机