



21世纪全国高等院校计算机教育“十一五”规划教材
丛书主编 全国高等学校计算机教育研究会课程与教材建设委员会主任 李大友

C语言程序设计教程

主 编	姜庆娜	姜玉波
副主编	杜忠友	刘 浩
	王新春	刘秀婷



中国计划出版社



中国计算机学会推荐教材 高等学校计算机专业核心课程教材
教育部高等学校计算机专业教学指导委员会推荐教材

C语言程序设计教程

林 燕 编
林 燕 编



21世纪全国高等院校计算机教育“十一五”规划教材

C 语言程序设计教程

本书编委会 编著

中国计划出版社

图书在版编目 (CIP) 数据

C语言程序设计教程 / 《C语言程序设计教程》编委会
编著. —北京: 中国计划出版社, 2007. 12
21世纪全国高等院校计算机教育“十一五”规划教材
ISBN 978-7-80242-043-4

I. C… II. C… 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 (2007)·第201256号

内 容 简 介

本书浓缩了作者多年的教学经验, 按照认识规律, 对章节顺序做了合理安排, 做到先易后难, 循序渐进。本书主要包括: C语言的基础知识, 数据类型、运算符与表达式, 顺序结构、选择结构和循环结构程序设计, 数组, 函数, 指针, 对函数的进一步讨论与预处理命令, 结构体、共用体与枚举类型, 文件等。

本书可作为高等院校学生C语言程序设计课程的教材, 也可作为高职高专、成人教育、社会培训的教材, 还可作为C语言自学者的教材或参考书。

21世纪全国高等院校计算机教育“十一五”规划教材

C语言程序设计教程

本书编委会 编著

☆

中国计划出版社出版

(地址: 北京市西城区木樨地北里甲11号国宏大厦C座4层)

(邮政编码: 100038 电话: 63906433 63906381)

新华书店北京发行所发行

河北省高碑店市鑫宏源印刷厂印刷

787×1092毫米 1/16 20.5印张 499千字

2007年12月第一版 2007年12月第一次印刷

印数1—3000册

☆

ISBN 978-7-80242-043-4

定价: 29.00元

丛书编委会

主 任：李大友

副主任：王行言 郑 莉

委 员：（按音序排列）

樊金生 冯春辉 高延武 韩金仓 刘凤田

刘 云 刘建臣 刘三满 罗德茹 彭宣戈

齐玉斌 孙晨霞 王潜平 王书海 姚 华

杨晓斌 张广斌 赵建明 赵连胜 邹修明

本书编委会

主 编：姜庆娜 姜玉波

副主编：杜忠友 刘 浩 王新春 刘秀婷

参 编：孙红敏 解艳艳 张海林 李 锋

孙晓燕 徐一红 姜 亮

丛 书 序

随着我国高等教育发展与改革的逐步深化,越来越多的高等院校将其自身定位于工程型或应用型,立足于培养能够满足各行各业需求的,素质高、能力强的应用型人才。与此同时,由于信息化是当今社会与经济发展的必然趋势,因而应用信息技术的能力亦将会成为衡量人才水平的重要标尺。由此可见,培养既具有专业知识,又拥有良好信息技术应用能力的人才,是现今高等教育发展与改革的目标之一。

目前,教育部有关计算机教育的教学指导委员会、全国高等学校计算机教育研究会等学术团体、各高等院校的专家学者已经在计算机教学与教材改革方面做了大量的工作,许多一线教师已经在计算机教学和科研方面积累了许多宝贵经验。在这些条件下,通过将其教研成果汇总并转化为教材的形式向全国各高等院校推广,对于促进高等院校计算机教育的发展与改革,培养应用型人才,是一件十分有意义的事情。

鉴于以上情况,中国计划出版社与全国高等学校计算机教育研究会决定联合策划组织、编写出版了本套“21 世纪全国普通高等院校计算机教育‘十一五’规划教材”。为实施精品战略,出版社与全国高等学校计算机教育研究会在全国范围内进行了系统、详细的调查,对各类教学指导性文件进行了认真、深入的研究,对国内外已出版同类教材进行了客观、理性的分析,组织专家学者、一线教师及企业人员展开研讨,以期打造切实符合实际教学需求的精品教材。

为配合各学校的精品课程建设工程,本套教材以国家级精品课程指标为指引方向,借鉴其他兄弟出版社的先进经验和成功案例,提出了建设“立体化教学资源平台”的概念,其内容包括教材、教学辅导资料、教学资源包、网络平台等内容,并将在后续培训、论文发表等多方面满足教师与精品课程建设的需求。

本套教材具有以下特点。

1. 定位明确,应用为本

本套教材定位于高等院校学生计算机应用能力的培养,不仅要使学生理解计算机相关的基本理论与基本知识,还要使学生掌握利用计算机解决实际问题的能力。要使学生面对一个实际问题时,不仅要知其然,还要知其所以然,更要会其如何然,最终,要具备实际操作应用能力。

2. 案例驱动,能力培养

本套教材通过从实际应用中提炼出的案例来辅助知识的讲授与能力的培养,在案例设计时从其科学性、实用性及开放性出发,尽量营造贴近实际应用的环境,激发学生的学习兴趣,从而提高教学效率,提高学生的实际应用能力。

3. 资源丰富,便于教学

我社免费为选用本套教材中图书的教师提供如下资源服务:

- 多媒体电子课件(PowerPoint 格式)
- 所有案例的相关素材(图片、声音与源程序等)与最终结果

- 所有习题的素材与答案
- 两套模拟测试题及答案
- 不定期组织教师培训

高等院校计算机教育的发展与改革不会停止，各院校的实际情况又有所不同，我们恳请各位老师在使用过程中提出批评与建议，以便及时改进教材欠妥与不足之处，使本套教材日趋完善。

我们相信在各位专家学者与一线教师的支持与帮助下，本套教材一定能成为特点鲜明、质量上乘的精品教材，同时，我们也希望通过本套教材的出版为高等院校计算机教育的发展与改革做出自己的一份贡献。

丛书编委会

前 言

在我国大多数高校,无论是计算机专业还是非计算机专业,往往将“C 语言程序设计”作为第一门程序设计课程。笔者从事 C 语言程序设计教学近 20 年,在不断追踪课程发展趋势和从事教学改革实践的过程中,积累了不少教学经验,根据教育部高等学校非计算机专业计算机基础课程教学指导分委员会“关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求”(简称“白皮书”)的精神和中国高等院校计算机基础教育改革课题教研组发布的“中国高等院校计算机基础教育课程体系 2006”(简称 CFC2006)中对 C 语言程序设计的要求,编写了本书。

“C 语言程序设计”作为大学计算机基础教学系列中的核心课程“计算机程序设计基础”中的一门课程,它主要讲授程序设计语言的基本知识和程序设计的方法与技术,其内容以程序设计语言的语法规则和程序设计技术的基本方法为主,同时包括程序设计方法学、数据结构与算法基础等方面的初步内容。

本书的主要思想是通过对 C 语言语法规则的学习,通过对各领域程序设计应用案例的思考和理解,使学生掌握计算机程序设计的思想和方法,初步具备在各领域应用计算机的能力,并为后续课程的学习创造条件,从而贯彻和达到“白皮书”和 CFC2006 的要求。

本书的主要特点如下:

一、按照认识规律,对章节顺序作了合理安排,做到先易后难,循序渐进,语言叙述注重图文并茂,理论讲解注重结合实际应用,这些有助于读者很快进入角色,进而对本书、本课程产生兴趣,易于教学,便于自学。

二、例题贴近实际生活,章末配有大量的练习题,这便于读者领会、理解、掌握和自我检查所学的语法规则及程序设计方法。

三、配有学习指导与实验教程、教学课件,便于教、学双方参考使用。

四、做到提升学生的知识——能力——素质,把握教学的难度——深度——强度,体现基础——技术——应用,提供教材——学习指导与实验——课件支持。

全书共分 11 章,内容包括: C 语言的基础知识,数据类型、运算符与表达式,顺序结构、选择结构和循环结构程序设计,数组,函数,指针,对函数的进一步讨论与预处理命令,结构体、共用体与枚举类型,文件等。

本书可作为高等院校学生 C 语言程序设计课程的教材,也可作为高职高专、成人教育、社会培训的教材,还可作为 C 语言自学者的教材或参考书。

建议教学总学时为 72 学时(其中,理论课 42 学时、上机实验课 30 学时):第 1~3 章各 4 学时,第 4~6 章各 8 学时,第 7~8 章各 10 学时,第 9 章 4 学时,第 10 章 8 学时,第 11 章 4 学时。

本书由山东建筑大学姜庆娜副教授、姜玉波副教授、杜忠友教授、刘浩教授、刘秀婷

副教授以及济南职业学院的王新春老师主编。为了集思广益，吸收了山东建筑大学的多位教师参与编写：解艳艳老师参与了第 3 章的编写，张海林老师和孙晓燕老师参与了第 7 章和第 9 章的编写，李锋老师参与了第 10 章的编写。此外，山东省工会管理干部学院的徐一红老师参与了第 6 章的编写，山东工艺美术学院的姜亮老师参与了第 11 章的编写，孙红敏老师参与了第 8 章的编写。

在本书的编写过程中，参阅了大量的图书资料，其中的部分参考资料已列在参考文献中，编者在此向这些图书的作者表示衷心的感谢。

由于编者水平和时间所限，书中难免有不足之处，恳请有关专家和广大读者不吝赐教（E-mail:jiangqn@sdjzu.edu.cn）。

编者

2007年10月

目 录

第1章 程序设计与C语言概述	1
1.1 程序设计语言	1
1.2 C语言概述	3
1.2.1 C语言的发展历史	3
1.2.2 C程序的基本结构	3
1.2.3 C语言的特点	6
1.3 程序设计与算法	7
1.3.1 程序设计	7
1.3.2 算法	8
1.4 C程序的开发过程	10
1.5 Turbo C集成开发环境简介	11
1.6 小结与提高	14
1.7 思考与练习	14
第2章 数据类型、运算符与表达式	16
2.1 基本数据类型	16
2.1.1 整型	16
2.1.2 实型	18
2.1.3 字符型	19
2.2 常量与变量	20
2.2.1 常量	20
2.2.2 变量	24
2.2.3 标识符的命名原则	25
2.3 运算符与表达式	25
2.3.1 C语言的运算符	25
2.3.2 算术运算符与算术表达式	26
2.3.3 算术表达式中的数据类型转换问题	27
2.3.4 赋值运算符与赋值表达式	28
2.3.5 自加与自减运算符	30
2.3.6 负号运算符、逗号运算符与sizeof运算符	31
2.3.7 位运算	32
2.4 小结与提高	36
2.5 思考与练习	37

第3章 顺序结构程序设计	40
3.1 简单示例及顺序结构特点	40
3.2 数据输出	41
3.2.1 printf——格式化输出函数	41
3.2.2 putchar——字符输出函数	49
3.3 数据输入	50
3.3.1 scanf——格式化输入函数	51
3.3.2 getchar——字符输入函数	56
3.4 复合语句与空语句	57
3.4.1 复合语句	57
3.4.2 空语句	58
3.5 程序举例	58
3.6 小结与提高	60
3.7 思考与练习	60
第4章 选择结构程序设计	66
4.1 用于描述条件的运算符与表达式	66
4.1.1 关系运算符与关系表达式	66
4.1.2 逻辑运算符与逻辑表达式	67
4.2 if语句的三种格式	69
4.2.1 if...else格式与两分支结构	69
4.2.2 if格式与单分支结构	71
4.2.3 else if格式与多分支结构	72
4.2.4 if语句与分支的嵌套结构	74
4.2.5 条件运算符与二分支结构	75
4.3 break语句与switch语句	76
4.3.1 break语句	76
4.3.2 switch语句	76
4.4 语句标号和goto语句	78
4.5 应用举例	79
4.6 小结与提高	80
4.7 思考与练习	81
第5章 循环结构程序设计	85
5.1 while语句	85
5.2 do...while语句	87
5.3 for语句	90
5.4 三种循环语句的比较	93
5.5 循环的嵌套结构	93
5.6 break语句和continue语句	95

5.6.1 break语句	95
5.6.2 continue语句	97
5.7 程序举例	98
5.8 小结与提高	103
5.9 思考与练习	104
第6章 数组	110
6.1 一维数组	110
6.1.1 一维数组的定义	110
6.1.2 一维数组的初始化	112
6.1.3 一维数组应用举例	114
6.2 二维数组	118
6.2.1 二维数组的定义	118
6.2.2 二维数组的初始化	119
6.2.3 二维数组应用举例	120
6.3 字符数组与字符串	122
6.3.1 一维字符数组与一个字符串	123
6.3.2 二维字符数组与一组字符串	129
6.3.3 字符数组应用举例	129
6.4 数组综合应用举例	131
6.5 小结与提高	133
6.6 思考与练习	133
第7章 函数	144
7.1 概述	144
7.2 函数定义	146
7.2.1 函数名与形参	147
7.2.2 函数体	149
7.2.3 函数值与函数值的数据类型	150
7.3 函数调用	151
7.3.1 实参	151
7.3.2 函数调用在main中出现的位置	157
7.4 函数声明	157
7.5 变量的存储属性	159
7.5.1 变量的存储类型	160
7.5.2 局部变量	160
7.5.3 外部变量	165
7.6 数组的存储属性	169
7.7 动态存储方式与静态存储方式	170
7.8 小结与提高	171
7.9 思考与练习	172

第8章 指针	181
8.1 指针与基本类型的变量	181
8.1.1 与指针有关的基本概念	181
8.1.2 指针变量的定义与使用	182
8.1.3 指针变量作为被调函数的形参	186
8.2 指针与一维数组	189
8.2.1 一维数组元素的间接访问	189
8.2.2 用指针变量存放一维数组元素的地址	190
8.2.3 一维数组作为函数参数	193
8.3 指针与二维数组	196
8.3.1 二维数组行与元素的表示	196
8.3.2 用指针变量存放二维数组元素的地址与行地址	199
8.3.3 二维数组作为函数参数	201
8.4 指针与字符串	202
8.4.1 一个字符串的表示	202
8.4.2 一组字符串的表示	207
8.5 指向指针的指针	210
8.6 小结与提高	212
8.7 思考与练习	212
第9章 对函数的进一步讨论与预处理命令	220
9.1 函数的嵌套调用	220
9.2 递归函数与递归调用	221
9.3 指针与函数	223
9.3.1 返回值为指针的函数	223
9.3.2 指向函数的指针变量	225
9.4 函数值的存储属性	228
9.5 main函数的参数	228
9.6 预处理命令	230
9.6.1 宏定义	230
9.6.2 文件包含	236
9.6.3 条件编译	238
9.7 小结与提高	241
9.8 思考与练习	242
第10章 结构体、共用体与枚举类型	245
10.1 结构体	245
10.1.1 结构体类型	246
10.1.2 结构体变量	247
10.1.3 结构体数组	252
10.1.4 指向结构体类型的指针	256

10.1.5	结构体与函数	258
10.2	链表	262
10.2.1	用指针和结构体构成链表	262
10.2.2	处理动态链表所需的函数	264
10.2.3	链表的基本操作	265
10.3	共用体	268
10.3.1	共用体类型	268
10.3.2	共用体变量	269
10.3.3	共用体应用举例	271
10.4	枚举	273
10.4.1	枚举类型	273
10.4.2	枚举变量	273
10.5	类型定义语句typedef	276
10.6	小结与提高	277
10.7	思考与练习	278
第11章	文件	285
11.1	与文件有关的概念	285
11.1.1	ASCII文件与二进制文件	285
11.1.2	缓冲文件系统	286
11.1.3	文件型指针	287
11.2	文件的打开与关闭	288
11.2.1	文件的打开	288
11.2.2	文件的关闭	290
11.3	文件的读/写操作	290
11.3.1	fputc函数与fgetc函数的使用	290
11.3.2	fgets函数与fputs函数的使用	293
11.3.3	fscanf函数与fprintf函数的使用	295
11.3.4	按数据块方式读写——fwrite函数与fread函数	296
11.4	文件的定位	298
11.5	小结与提高	300
11.6	思考与练习	301
附录A	常用字符与ASCII码对照表	303
附录B	运算符的优先级和结合性	304
附录C	C语言中的关键字	306
附录D	C语言中的库函数	307
	主要参考文献	313

第 1 章

程序设计与 C 语言概述

为什么要学习这门课程？学习这门课程有什么目的和意义呢？本课程是向学生介绍结构化程序设计的基本思想和方法，通过学习结构化程序设计语言，了解用计算机解决问题的一般方法，掌握程序设计的基本方法，掌握编写和调试简单应用程序的方法，使学生养成利用计算机解决工作、生活中的实际问题的习惯，提高学生计算机方面的能力和素质。

C 语言是一种介于高级语言与汇编语言之间的程序设计语言。本章在简述程序设计语言与程序设计的基础上，将介绍 C 语言的诞生与发展，C 程序的基本组成及其特点，C 语言的特点，算法的描述，C 程序的上机开发过程及开发环境。

1.1 程序设计语言

当今社会，计算机作为一种现代化工具，已经广泛应用于人们生活的各个领域。凡是使用过计算机的人都知道，计算机的神奇之处在于能够运行安装在计算机上的各种程序。程序是用计算机能够理解和处理的语言（即计算机语言）作为描述工具，对通过计算机进行处理的数据以及处理步骤进行描述。这里的计算机语言，我们称之为程序设计语言。所以，可以简单地理解为：程序设计语言就是用来编写程序的语言，是人和计算机进行交流的工具。

程序设计语言是科学家们研制开发出来的。人们如果想让计算机为自己服务，按照自己的意图工作，就必须掌握一门程序设计语言，把自己的意图编写成程序语言让计算机执行。

从第一种程序设计语言诞生到如今，程序设计语言已经被研发出了成百上千种，被广泛使用的也达几十种。综合其特点，可以把程序设计语言分为四类：机器语言、汇编语言、高级语言和面向对象的语言。

1. 机器语言

机器语言是用计算机能够直接识别的二进制代码来表示计算机的硬件指令，用这种方法表示的硬件指令称为机器指令，即机器语言。机器语言的唯一优点是用机器指令编写的程序可以直接送给计算机去执行。与其优点相比，机器语言存在许多缺点：用一串 0 或 1 表示机器指令容易出错，可读性差，不利于人们学习和掌握，编程的效率低，不易修改和维护等。为了克服这些缺点，科学家研发出了汇编语言。

2. 汇编语言

汇编语言是用助记符来表示计算机的硬件指令。助记符一般使用的是英文单词，对于比较长的单词则采用其缩写，例如，用英文 `add` 表示加法指令，用 `subtract` 的缩写 `sub` 表示减法指令。与机器语言相比，汇编语言有助于人们学习和理解。但是，汇编语言的助记符计算机不能直接识别，还需要翻译成用二进制代码表示的机器指令才能送给计算机去执行。

机器语言和汇编语言都依赖于计算机的硬件指令，故称两者为面向机器的语言，也称为低级语言。汇编语言与计算机硬件密切相关，适合用它开发系统软件，许多大型计算机系统的操作系统都是用汇编语言编写的。不同的计算机系统提供的指令系统不同，因此用汇编语言编写的程序通用性差，不便于移植在不同的计算机系统上。针对这些问题，在 20 世纪 50 年代中期，计算机专家们推出了高级语言。

3. 高级语言

与低级语言相比，高级语言的“高级”体现在：它与计算机硬件有较少的直接依赖关系；它比较接近自然语言和数学语言。用高级语言编写程序类似于用英语写文章。文章由段落构成，段落又是由句子组成的，而句子是把一些英语词汇按照语法要求排列在一起来表达具体意思的。程序由函数或过程或称子程序构成，函数由语句构成，语句是把标识符、数据、运算符按照格式规定组织在一起，告诉计算机该做什么的。标识符通常由英文单词或其缩写构成。因此，高级语言适合大多数人学习和掌握。

用高级语言编写的程序同样需要经过“翻译”后计算机才能识别，就像两个母语不同的人交谈时必须借助于翻译一样。高级语言的翻译系统有两种：解释系统与编译系统。解释系统是指：对高级语言程序翻译一句提交给计算机执行一句，整个程序翻译完计算机也就执行完，类似于“口译”。编译系统是指：把高级语言程序经过分步处理和整体优化翻译成一个完整的机器代码程序（也就是可执行程序），再送给计算机去执行，类似于“笔译”。

已经推出的高级语言有几百种之多，`ALGOL`、`FORTRAN`、`COBOL`、`BASIC`、`PASCAL` 等都是曾经被广泛使用的高级语言。高级语言适合编写应用软件。`C` 语言是一种高级语言，同时也具备对二进制代码的位进行运算的功能，所以它既适合编写应用软件，也适合编写系统软件。

以上高级语言又称为面向过程的程序设计语言，也就是面向具体的解题过程，即算法语言。

4. 面向对象的语言

面向对象的语言起源于 20 世纪 60 年代挪威开发的 `Simula` 编程语言，它是一种以对象作为基本程序结构单位的程序设计语言，即用于描述的设计以对象为核心，而对象是程序运行的基本成分。语言中提供了类、继承等成分。

面向对象语言的发展有两个方向：一种是纯面向对象语言，如 `Smalltalk`、`Eiffel` 等；另一种是混合型面向对象语言，即在面向过程语言中加入类、继承等成分，如 `C++`、`Java`、

Delphi 等。

1.2 C 语言概述

1.2.1 C 语言的发展历史

C 语言的产生与 UNIX 操作系统有密切关系。在 20 世纪 60 年代末,美国贝尔实验室开发出了一种 UNIX 操作系统,早期的 UNIX 是用汇编语言编写的。针对汇编语言移植性差等缺点,贝尔实验室的 Ken Thompson 决定开发一种程序设计语言,使它在具有汇编语言特点(适合编写系统软件)的基础上,又具有高级语言的特点(易于理解,移植性强等),目的是使用它来编写 UNIX 操作系统。

当时有一种编程语言叫 BCPL (Basic Combined Programming Language), BCPL 的前身是 CPL 语言,而 CPL 语言的前身是 ALGOL 语言。ALGOL 语言是 1958 年由国际委员会推出的一种高级语言。1963 年剑桥大学将 ALGOL 60 语言发展成为 CPL (Combined Programming Language) 语言,它是在 ALGOL 语言中增加了汇编语言的一些功能。这样,从理论上讲,CPL 语言是一种既具有高级语言特点又具有汇编语言功能的编程语言,但因其规模大,实际上难以实现。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL 语言。

1970 年, Ken Thompson 从当时的众多语言中选中了 BCPL 语言,对它做了进一步的简化和完善,并为其起了一个名字“B 语言”。B 语言在功能上过于简单,而且缺乏丰富的数据类型。为了更好地适应编写系统软件的要求,美国贝尔实验室的 Dennis M.Ritchie 在 B 语言的基础上设计出了一种新的语言,他取了 BCPL 的第二个字母作为这种语言的名字,这就是“C 语言”。1973 年, Ken Thompson 和 Dennis M.Ritchie 用 C 语言重写了 UNIX 操作系统代码,即 UNIX 第 5 版,这一版本奠定了 UNIX 的基础,使其逐渐成为最重要的操作系统之一。

1978 年, Brian W.Kernighian 和 Dennis M.Ritchie 合作出版了名著《The C Programming Language》,这本书所介绍的 C 语言被人们称为非官方的标准 C。随着 UNIX 操作系统的成功和广泛流行, C 语言也引起人们的注意,并成为软件开发中深受人们欢迎的一种程序设计语言。随着 C 语言的广泛应用,出现了几十种不同版本的 C 语言编译系统,它们之间存在着某些差异。为了改变这种情况,1983 年,美国国家标准化协会(ANSI)制定了 ANSI 的标准 C; 1987 年, ANSI 又公布了 87 标准 ANSI C; 1990 年, C 语言成为国际标准化组织(ISO)通过的标准语言。

20 世纪 80 年代后期,在 C 语言的基础上又推出了 C++ 语言(也称为带类的 C),它在保留 C 语言全部功能的基础上,增加了面向对象语言的特点。

1.2.2 C 程序的基本结构

下面通过几个用 C 语言编写的程序,来介绍 C 语言程序的组成,并总结 C 程序基本结构的特点,使读者对 C 语言有个感性认识。