

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

C 程序设计教程

常东超 吕宝志 冯锡炜 郭来德 编著

清华大学出版社



21 世纪普通高校计算机公共课程规划教材

C 程序设计教程

常东超 吕宝志 冯锡炜 郭来德 编著

清华大学出版社
北京

内 容 简 介

本书参照最新计算机等级考试(二级C)教学大纲,并根据高校最新“C语言程序设计”课程教学大纲要求编写而成。全书共分11章,主要内容有:C程序的基本组成以及程序开发过程;C语言的基本数据类型、运算符、表达式、数据类型转换及标准的输入输出函数;C语言的基本语句和流程控制语句;数组、函数、指针的概念及用法;C语言的编译预处理功能;C语言结构体与共用体、位运算与位段的相关知识及使用;C语言中文件的相关概念以及文件的各种操作方法。

本书既可以作为高等学校本专科计算机“C语言程序设计”课程的教材,也可以作为培养读者计算机编程能力和参加全国计算机等级考试(C语言)的自学参考书。

本书另配有电子教案(PPT格式)与课后习题解答(Word格式),联系邮箱:changdc885@126.com。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C程序设计教程/常东超等编著. —北京:清华大学出版社,2010.1

(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-21410-6

I. C… II. 常… 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(2009)第201427号

责任编辑:梁颖

责任校对:李建庄

责任印制:何芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市清华园胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:20.25 字 数:491千字

版 次:2010年1月第1版 印 次:2010年1月第1次印刷

印 数:1~3000

定 价:28.50元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:035691-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对性不同

层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人: 梁颖 liangying@tup.tsinghua.edu.cn

前 言

C 语言是国内外长期并广泛使用的一种计算机语言,是计算机应用人员,特别是硬件产品开发和底层程序开发人员应该掌握的程序设计工具之一。C 语言以其语法结构简洁精妙、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、可移植性好而深受广大程序开发人员青睐。

由于 C 语言具有上述优点,所以至今仍得到广泛应用,无论是 Windows 还是 Linux 操作系统,其底层程序都是用 C 语言编写的,而且大部分网络协议、画面漂亮的游戏、工业控制程序等也是用 C 语言来实现的;而且很多新型的语言都是 C 语言衍生的,如 C++、Java、C#、J#、Perl 等。掌握 C 语言是掌握许多其他优秀语言的基础,有了这个基础,再学其他语言就容易多了。

随着国家信息化步伐的加快,对高校理工科学生的计算机基础教育提出了新的要求,全国计算机等级考试的三级和四级考试重新将 C 语言作为必考内容,因此绝大多数理工科高校、高职院校都把“C 语言程序设计”作为一门重要的计算机基础课程。

本书根据最新 C 语言程序设计教学大纲,在广泛汲取多种 C 语言教材优点的基础上,充分考虑计算机等级考试要求,由多年从事 C 语言教学的数位教师联合编写而成;与以往的 C 语言教材相比,更加贴近了全国计算机等级考试大纲,有助于全面提高读者的程序设计能力和取得“等考”资格证书。

全书共分 11 章,其中:

第 1 章主要介绍 C 语言的发展与特点,C 程序的基本组成以及程序开发过程;

第 2 章介绍 C 语言中基本数据类型及标准的输入输出函数;

第 3 章介绍了运算符与运算符的分类、表达式与表达式的计算、数据类型转换;

第 4 章介绍了基本语句和流程控制语句,介绍了结构化程序设计的三种基本结构——顺序、选择、循环;

第 5~7 章分别介绍了数组、函数、指针;

第 8 章介绍 C 语言的编译预处理功能,包括不带参数的宏定义以及带参数的宏定义,使用文件包含功能,条件编译的几种形式及应用;

第 9 章介绍结构体与共用体的相关知识;

第 10 章介绍位运算与位段的相关知识及使用;

第 11 章介绍 C 语言中文件的相关概念以及文件的各种操作方法。

全书由常东超、吕宝志、冯锡炜、郭来德编著,高文来、刘培胜、邵蓉、李会举、白雪、石元博等参加了部分章节的编写和校对工作;全书由常东超统稿。

魏海平教授审阅了全书,并提出了许多宝贵意见和建议,在此表示衷心的感谢。

本书内容翔实,层次分明,结构紧凑,叙述深入浅出,既可作为高等学校、高等职业院校、大专院校、计算机培训学校计算机及相关专业“C 语言程序设计”课程的教材,也可作为编程人员和 C 语言自学者的参考用书,还可作为全国计算机等级考试的辅导用书。限于编者水平有限,书中如有不足之处,敬请读者批评指正,以利作者再版改进。

编 者

2009 年 9 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言的发展历史	1
1.2 使用 C 语言的优势	1
1.3 C 语言程序基本结构	2
1.3.1 C 语言程序范例	2
1.3.2 C 语言程序结构特点	4
1.4 C 语言程序的开发过程	4
本章小结	6
习题 1	7
第 2 章 数据类型	8
2.1 数据类型的分类	8
2.1.1 什么是数据类型	8
2.1.2 C 语言的数据类型分类	9
2.2 常量	11
2.2.1 整型常量	11
2.2.2 字符串常量	14
2.2.3 实型常量	15
2.3 变量	15
2.3.1 什么是变量	15
2.3.2 变量定义语句	16
2.4 标准输入输出函数	17
本章小结	25
习题 2	25
第 3 章 运算符和表达式	29
3.1 运算符与运算符的分类	29

3.2 表达式与表达式的计算	36
3.2.1 表达式	36
3.2.2 复合表达式的计算	37
3.2.3 数据类型转换	40
本章小结	42
习题 3	42
第 4 章 语句	45
4.1 基本语句	45
4.2 流程控制语句	46
4.2.1 程序流程	47
4.2.2 分支语句	49
4.2.3 循环语句	55
4.2.4 转向语句	62
4.2.5 综合举例	66
本章小结	69
习题 4	69
第 5 章 数组	75
5.1 一维数组	75
5.1.1 一维数组的定义	75
5.1.2 一维数组元素的引用	76
5.1.3 一维数组的初始化	77
5.1.4 一维数组应用举例	77
5.2 二维数组和多维数组	82
5.2.1 二维数组和多维数组的定义	82
5.2.2 二维数组和多维数组的引用	84
5.2.3 二维数组和多维数组的初始化	84
5.2.4 二维数组和多维数组的举例	86
5.3 字符数组和字符串	89
5.3.1 字符数组	89
5.3.2 字符串	91
5.3.3 字符串处理函数	92
5.3.4 字符数组的举例	98
本章小结	101
习题 5	102

第 6 章 函数	104
6.1 函数的概念	104
6.2 函数的定义和说明	105
6.2.1 函数的定义	105
6.2.2 函数声明	107
6.3 函数调用	109
6.4 函数传递机制	113
6.5 递归	118
6.6 变量的作用域及存储类别	122
6.6.1 变量的作用域	122
6.6.2 变量的存储类别	125
6.7 库函数	129
6.7.1 库函数简介	129
6.7.2 常用库函数	130
本章小结	134
习题 6	135
第 7 章 指针	138
7.1 指针的概念	138
7.2 指针变量的定义	139
7.3 指针的运算	141
7.3.1 指针变量的赋值运算	141
7.3.2 指针变量的算术运算	142
7.3.3 指针变量的关系运算	146
7.4 指向数组的指针	146
7.5 字符串的指针	149
7.6 函数与指针	151
7.6.1 指针变量作为函数参数	151
7.6.2 指针型函数	154
7.6.3 指向函数的指针	155
7.7 指针数组和二维数组的行指针	157
7.7.1 指针数组的定义和应用	157
7.7.2 二维数组的行指针	160
7.7.3 main 函数的参数	162
7.8 指向指针的指针	164

7.8.1 二级指针的定义	164
7.8.2 二级指针的应用	164
7.9 有关指针的数据类型和指针运算的小结	165
7.9.1 有关指针的数据类型的小结	165
7.9.2 指针运算的小结	166
7.9.3 void 指针类型	166
7.10 与指针有关的 C 程序典型举例	166
本章小结	173
习题 7	174
第 8 章 文件包含和预编译	184
8.1 宏定义	184
8.1.1 不带参数的宏定义	184
8.1.2 带参数的宏定义	188
8.2 文件包含	190
8.3 条件编译	192
本章小结	194
习题 8	194
第 9 章 结构体与共用体	198
9.1 结构体类型变量定义和引用	198
9.1.1 结构体类型定义	198
9.1.2 结构体类型变量定义	199
9.1.3 结构体类型变量的初始化	201
9.1.4 结构体类型变量的引用	201
9.2 结构体数组	202
9.2.1 结构体数组的定义	202
9.2.2 结构体数组的初始化	203
9.2.3 结构体数组的引用	203
9.2.4 结构体数组的应用举例	203
9.3 结构体指针	205
9.3.1 指向结构体变量的指针	205
9.3.2 指向结构体数组的指针	208
9.3.3 用指向结构体的指针作函数参数	209
9.4 链表	210
9.4.1 用指针处理链表	211

9.4.2	处理动态链表的函数	213
9.4.3	建立链表	215
9.4.4	输出链表	216
9.4.5	链表的删除操作	217
9.4.6	链表的插入操作	218
9.5	共用体	220
9.5.1	概念	220
9.5.2	共用体类型的定义	220
9.5.3	共用体变量的定义	221
9.5.4	共用体变量的引用	221
9.5.5	共用体类型数据的特点	223
9.6	枚举类型	223
9.7	类型定义	225
	本章小结	227
	习题 9	227
第 10 章	位运算	230
10.1	概述	230
10.2	位运算符和位运算	230
10.2.1	按位与运算符(&)	230
10.2.2	按位或运算符()	232
10.2.3	按位取反运算符(~)	232
10.2.4	按位异或运算符(^)	233
10.2.5	左移运算符(<<)	234
10.2.6	右移运算符(>>)	235
10.2.7	位运算赋值运算符	236
10.3	位运算应用举例	236
10.4	位段	240
10.4.1	位段的概念	240
10.4.2	位段的定义	240
10.4.3	位段的引用	241
	本章小结	242
	习题 10	242
第 11 章	文件	244
11.1	C 语言文件的概念	244

11.2	文件指针	245
11.3	文件的打开与关闭	245
11.3.1	文件的打开函数 fopen	245
11.3.2	文件关闭函数 fclose	247
11.4	文件的读写	247
11.4.1	字符读写函数 fgetc 和 fputc	247
11.4.2	字符串读写函数 fgets 和 fputs	251
11.4.3	数据块读写函数 fread 和 fwrite	252
11.4.4	格式化读写函数 fscanf 和 fprintf	254
11.5	文件的随机读写	255
11.5.1	文件定位	255
11.5.2	获取文件指针当前位置	256
11.6	文件检测函数	256
11.6.1	文件结束检测函数 feof	256
11.6.2	读写文件出错检测函数	256
	本章小结	257
	习题 11	257
附录 A C 语言实验指导		259
附录 B 常用字符与 ASCII 码对照表		285
附录 C Turbo C 常用标准库函数		286
附录 D C 语言保留关键字		299
附录 E 课后习题参考答案		300
参考文献		310

1.1 C 语言的发展历史

计算机的高级语言具有使用方便、可移植性好、容易理解的特点,但是随着计算机应用的迅猛发展,计算机的语言要求能够直接操作计算机的硬件(如对内存操作、位操作等),在这种情况下,人们希望有一种语言既有高级语言使用方便的特点,又有低级语言能够直接操作计算机硬件的特点,20 世纪 70 年代初期,在种类繁多的程序语言家族中又增添了一名新成员——C 语言。

C 语言是一种编译型程序语言,它的前身是马丁·理查德(Martin Richards)在 20 世纪 60 年代开发的 BCPL 语言。BCPL 语言是计算机软件人员在开发系统软件时,作为记述语言使用的一种程序语言;1970 年,美国贝尔实验室的肯·苏姆普逊(Ken Thompson)在软件开发工作中,继承和发展了 BCPL 语言的特点,进而提出了 B 语言,当时最新型的小型计算机——美国 DEC 公司的 PDP-7 型机中的 UNIX 操作系统就是使用 B 语言记述和开发的。此后,在美国贝尔实验室为更新型的小型机 PDP-11/20 进行 UNIX 操作系统的开发工作中,戴尼斯·利奇(Dennis M Ritchie)和布朗·卡尼汉(Brian W Kernighan)对 B 语言做了进一步的充实和完善,于 1972 年推出了一种新型的程序语言——C 语言。

C 语言功能强大而灵活,因此很快被传播到贝尔实验室之外,世界各地的程序员都使用它来编写各种程序。然而,在 C 语言出现不久,不同的组织便开始使用自己的 C 语言版本,由于没有统一的标准,使得不同版本的 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准,目前流行的 C 语言编译器绝大多数都遵守这一标准。

1.2 使用 C 语言的优势

在当前的计算机编程领域中,有大量的高级语言可供选择,如 C、Perl、Java 和 C++。这些都是非常卓越的计算机语言,能够完成大部分编程任务。虽然如此,但基于以下几个原因,很多人认为 C 语言是其中最佳的。

(1) C 语言功能强大,并且语言简洁、紧凑,使用方便、灵活。C 语言仅有 32 个关键字(见附录 D),9 种控制语句,程序的书写形式也很自由。C 语言可用于开发操作系统、字处

理器、图形、电子表格等项目,甚至可用于编写其他语言的编译器。具体可参考本书第 12 章。

(2) C 语言具有结构化的控制语句,用函数作为程序模块以实现程序的模块化。

(3) 数据类型丰富。C 语言除具有基本数据类型整型(int)、实型(float 和 double)、字符型(char)外,还有各种构造类型。利用这些数据类型可以实现复杂的数据结构,如堆栈、队列、链表等。

(4) C 语言表达能力强、语言简练,可以直接访问内存物理地址和硬件寄存器,可以表达直接由硬件实现的针对二进制位(bit)的运算。在语言成分的表示方法上尽可能简洁,I/O 操作不是作为 C 的语法成分而是通过 C 库函数实现,因而程序简洁,编译程序体积小。

(5) C 语言是可移植的。这意味着为一种计算机系统(如 IBM PC)编写的 C 语言,可以在其他系统中编译并运行,而只需做少量的修改,甚至无需修改。例如,在使用 Windows 操作系统的计算机上编写的 C 程序,可以不必修改或做少量修改就可成功移植到使用 Linux 操作系统的计算机上。C 语言的 ANSI 标准进一步加强了可移植性。

(6) C 语言生成的目标代码质量高,程序执行效率高。代码质量是指 C 程序经编译后生成的目标程序在运行速度上和存储空间上开销的大小。一般而言,运行速度越高,占用的存储空间越少,则代码质量越高。一般高级语言相对于汇编语言而言其代码质量要低得多,但 C 语言在代码质量上几乎可以与汇编语言媲美。

由于 C 语言的上述众多的特点,使它成为一个实用的通用程序设计语言,既可用于编写系统软件,又可编写应用软件,特别适用于编写各种与硬件环境相关的系统软件,不愧为一种强有力的系统程序设计语言。

1.3 C 语言程序基本结构

任何计算机程序语言,都具有特定的语法规则和一定的表现形式。程序的书写格式和程序的构成规则是程序语言表现形式的一个重要方面。按照规定的格式和构成规则书写程序,不仅可以使程序设计人员和使用程序的人容易理解,更重要的是,当把程序输入给计算机时,计算机能够充分“认识”,从而能够正确执行它。

1.3.1 C 语言程序范例

一个完整的 C 语言程序可由若干个函数构成,其中必须有且只能有一个以 main 命名的主函数。下面将介绍几个简单的 C 程序例子,从中分析 C 程序的结构。

【例 1.1】 编写一个程序,显示出以下一行文字:“I love China.”。

```
#include "stdio.h"
void main()
{
    printf("I love China.\n");
}
```

运行这个程序时,在屏幕上显示一行英文:

```
I love China.
```

这是一个仅由 main 函数构成的 C 语言程序。main 是函数名,函数名后面一对圆括号内写函数参数,本程序的 main 函数没有参数,故不写,但圆括号不能省略。void main()后面被大括号括起来的部分称为函数体。一般情况下,函数体由“说明部分”和“执行部分”组成(本例中只有执行部分而无说明部分),执行部分由若干语句组成。“\n”是换行符,即在输出“I love China.”后换行。

【例 1.2】 计算圆的面积。

```
#include "stdio.h"
void main()
{
    float s,r;                /* 定义 s,r 为浮点型变量 */
    printf("Please input 1 numbers:"); /* 输出提示字符串信息 Please input 1 numbers: */
    scanf("%f",&r);           /* 从键盘输入圆的半径 */
    s = 3.14 * r * r;          /* 计算圆的面积 */
    printf("s = %.2f\n",s);    /* 输出圆的面积 */
}
```

程序的运行结果如下:

```
Please input 1 numbers:3.5 ↵    (↵表示按 Enter 键)
s = 38.47
```

本程序也是一个仅由 main 函数构成的 C 语言程序。其中:

- float s,r; 是变量定义语句,定义两个浮点型变量,名为 s、r。
- printf("Please input 1 numbers:"); 这是一条输出语句,通过调用 printf 库函数在显示屏上输出指定的内容,此例输出字符串“Please input 1 numbers:”。
- scanf("%f",&r); 这是一条输入语句,通过调用 scanf 库函数从键盘上输入 r 的值,&r 中的“&”含义是“取地址”,此 scanf 函数的作用是将一个数值输入到变量 r 的地址所标识的单元中,也就是输入给变量 r。关于 scanf 函数的介绍详见第 2 章内容。

【例 1.3】 求两个整数中的大数。

```
#include "stdio.h"
int MAX(int x, int y)
{
    return( x>y ? x : y );
}
void main()
{
    int num1,num2,max;
    scanf("%d, %d", &num1,&num2);
```

```
max = MAX(num1, num2);  
printf("max = %d\n", max);  
}
```

运行这个程序时,输入:

5,8 ↵ (分别输入 5 和 8 给 num1、num2)

在屏幕上显示:

```
max = 8
```

本程序是由 main 函数和一个被调用的函数 MAX 构成的。MAX()函数的作用是返回 num1 和 num2 中较大的值,通过 return 语句将 num1 和 num2 中较大的数返回给主调函数 main 中的变量 max。返回值通过函数名 MAX 带回到 main 函数的调用处。main 函数中第 3 行为调用 MAX 函数,在调用时将实际参数 num1、num2 的值分别传送给 MAX 函数中的形式参数 x、y。经过执行 MAX 函数得到一个返回值,然后输出这个值。printf 函数中双引号内“max=%d\n”在输出时,其中“%d”将由 MAX 的返回值代替,“max=”原样输出。

通过对上述几个例子的分析,我们可以归纳出 C 语言程序的结构特点。

1.3.2 C 语言程序结构特点

函数是 C 语言程序的基本单位。void main()函数的作用,相当于其他高级语言中的主程序;C 语言中的其他函数,相当于其他高级语言中的子程序。

C 语言程序总是从 void main()函数开始执行。一个 C 语言程序总是从 void main()函数开始执行,而不论其在程序中的位置。当主函数执行完毕时,亦即程序执行完毕。习惯上,将主函数 void main()放在最前头。

分号“;”是 C 语句的一部分。

C 程序书写格式自由,一行内可写多条语句,且语句中的空格和回车符均可忽略不计。

程序的注释部分应括在“/*”和“*/”之间,“/*”和“*/”必须成对出现。“/”和“*”之间不允许留有空格,并且不能嵌套注释(如/*/*...*/),注释可以用西文,也可以用中文。注释可以出现在程序中的任何位置上。注释部分对程序的运行不起作用。在注释中可以说明变量的含义、程序段的功能,以便帮助人们阅读程序,因此一个好的程序应该有详细的注释。

1.4 C 语言程序的开发过程

C 语言采用编译方式将源程序转换为二进制的目标代码。一个编写完成的 C 程序在成功运行之前,一般经过编辑源代码、编译、链接、运行四个步骤。

1. 编辑源代码

使用一个文本编辑器编辑 C 源代码,创建一个包含源代码的磁盘文件,文件的扩展名