



卓越工程技术人才培养特色教材

C YUYAN
CHENGXU SHEJI

C语言程序设计 (第2版)

主 编 耿焕同



江苏大学出版社
JIANGSU UNIVERSITY PRESS



卓越工程技术人才培养特色教材

C语言程序设计

(第2版)

主 编 耿焕同

参编人员 李振宏 陈 遥 朱节中



江苏大学出版社
JIANGSU UNIVERSITY PRESS

镇 江

图书在版编目(CIP)数据

C 语言程序设计 / 耿焕同主编. — 2 版. — 镇江 :
江苏大学出版社, 2013. 11
ISBN 978-7-81130-584-5

I. ①C… II. ①耿… III. ①C 语言—程序设计 IV.
①TP312

中国版本图书馆 CIP 数据核字(2013)第 254386 号

C 语言程序设计(第 2 版)

主 编/耿焕同

责任编辑/李菊萍

出版发行/江苏大学出版社

地 址/江苏省镇江市梦溪园巷 30 号(邮编: 212003)

电 话/0511-84446464(传真)

网 址/http://press. ujs. edu. cn

排 版/镇江新民洲印刷有限公司

印 刷/扬中市印刷有限公司

经 销/江苏省新华书店

开 本/787 mm×1 092 mm 1/16

印 张/18.25

字 数/445 千字

版 次/2013 年 11 月第 2 版 2013 年 11 月第 2 次印刷

书 号/ISBN 978-7-81130-584-5

定 价/38.00 元

如有印装质量问题请与本社营销部联系(电话:0511-84440882)

序

深化高等工程教育改革、提高工程技术人才培养质量,是增强自主创新能力、促进经济转型升级、全面提升地区竞争力的迫切要求。近年来,江苏高等工程教育飞速发展,全省 46 所普通本科院校中开设工学专业的学校有 45 所,工学专业在校生约占全省普通本科院校在校生总数的 40%,为“十一五”末江苏成功跻身全国第一工业大省做出了积极贡献。

“十二五”时期是江苏加快经济转型升级、发展创新型经济、全面建设更高水平小康社会的关键阶段。教育部“卓越工程师教育培养计划”启动实施以来,江苏认真贯彻教育部文件精神,结合地方高等教育实际,着力优化高等工程教育体系,深化高等工程教学改革,努力培养造就一大批创新能力强、适应江苏社会经济发展需要的卓越工程技术后备人才。

教材建设是人才培养的基础工作和重要抓手。培养高素质的工程技术人才,需要遵循工程技术教育规律,建设一套理念先进、针对性强、富有特色的优秀教材。随着知识社会和信息时代的到来,知识综合、学科交叉趋势增强,教学的开放性与多样性更加突出,加之图书出版行业体制机制也发生了深刻变化,迫切需要教育行政部门、高等学校、行业企业、出版部门和社会各界通力合作,协同作战,在新一轮高等工程教育改革发展中抢占制高点。

2010 年以来,江苏大学出版社积极开展市场分析和行业调研,先后多次组织全省相关高校专家、企业代表就应用型本科人才培养和教材建设工作进行深入研讨。经各方充分协商,拟定了“江苏省卓越工程技术人才培养特色教材”开发建设的实施意见,明确了教材开发总体思路,确立了编写原则:

一是注重定位准确,科学区分。教材应符合相应高等工程教育的办学定位和人才培养目标,恰当把握与研究型工程人才、设计型工程人才及技能型工程人才的区分度,增强教材的针对性。

二是注重理念先进,贴近业界。吸收先进的学术研究与技术成果,适应经济转型升级需求,适应社会用人单位管理、技术革新的需要,具有较强的领先性。

三是注重三位一体,能力为重。紧扣人才培养的知识、能力、素质要求,



着力培养学生的工程职业道德和人文科学素养、创新意识和工程实践能力、国际视野和沟通协作能力。

四是注重应用为本,强化实践。充分体现用人单位对教学内容、教学实践设计、工艺流程的要求以及对人才综合素质的要求,着力解决以往教材中应用性缺失、实践环节薄弱、与用人单位要求脱节等问题,将学生创新教育、创业实践与社会需求充分衔接起来。

五是注重紧扣主线,整体优化。把培养学生工程技术能力作为主线,系统考虑、整体构建教材体系和特色,包括合理设置课件、习题库、实践课题以及在教学、实践环节中合理设置基础、拓展、复合应用之间的比例结构等。

该套教材组建了阵容强大的编写专家及审稿专家队伍,汇集了国家教学指导委员会委员、学科带头人、教学一线名师、人力资源专家、大型企业高级工程师等。编写和审稿队伍主要由长期从事教育教学改革实践工作的资深教师、对工程技术人才培养研究颇有建树的教育管理专家组成。在编写、审定教材时,他们紧扣指导思想和编写原则,深入探讨、科学创新、严谨细致、字斟句酌,倾注了大量的心血,为教材质量提供了重要保障。

该套教材在课程设置上基本涵盖了卓越工程技术人才培养所涉及的有关专业的公共基础课、专业公共课、专业课、专业特色课等;在编写出版上采取突出重点、以点带面、有序推进的策略,成熟一本出版一本。希望大家在教材的编写和使用过程中,积极提出意见和建议,集思广益,不断改进,以期经过不懈努力,形成一套参与度与认可度高、覆盖面广、特色鲜明、有强大生命力的优秀教材。

江苏省教育厅副厅长 丁晓昌

2012年8月

江苏省卓越工程技术人才培养特色教材建设

指导委员会

主任委员：丁晓昌（江苏省教育厅副厅长）

副主任委员：史国栋（常州大学党委书记）

孙玉坤（南京工程学院院长）

田立新（南京师范大学副校长）

梅 强（江苏大学副校长）

徐子敏（江苏省教育厅高教处处长）

王 恬（南京农业大学教务处处长）

委 员 会：（按姓氏笔画为序）

丁晓昌 马 铸 王 兵 王 恬

方海林 田立新 史国栋 冯年华

朱开永 朱林生 孙玉坤 孙红军

孙秀华 芮月英 李江蛟 吴建华

吴晓琳 沐仁旺 张仲谋 张国昌

张明燕 陆雄华 陈小兵 陈仁平

邵 进 施盛威 耿焕同 徐子敏

徐百友 徐薇薇 梅 强 董梅芳

傅菊芬 舒小平 路正南

前言

从 1971 年诞生到现在, C 语言已经成为最重要和最流行的高级程序设计语言之一。C 语言经久不衰的根源在于其具有方便性、灵活性和通用性等特点。在 C 语言的应用过程中,除了可开发各种类型的软件外,程序员还可直接对可编程硬件操作。因此, C 语言不仅是计算机学科重要的核心课程,也是其他理工科专业计算机基础知识的必修课。

随着时代的进步,掌握一门编程语言已经成为现代科技人员的一项基本技能。近年来,学习和掌握 C 语言的需求越来越迫切,特别是对在校理工类大学生,程序设计能力成为现代科技人才必备的一种能力。

虽然市面上已有很多关于 C 语言程序设计的书籍,但是存在着诸如从抽象枯燥的语法开始学习,或是将枯燥的数学问题作为实例等不足,这在某种程度上偏离了程序设计的核心,不仅容易挫伤初学者学习程序设计的信心,还会使初学者对程序设计丧失兴趣。我们根据多年从事 C 语言一线教学工作积累的经验和教学心得,充分借鉴现有 C 语言书籍的优点,采用抽象知识的学习方法,即以“为什么学? 如何学习? 学什么?”的顺序,科学设计教程和巧妙组织内容。

本书的主要特色和创新之处有:

(1) 理念新颖。围绕现代理工科专业技术人才的培养目标,取舍恰当、精心选择内容并科学编排,力求使非计算机专业的学生能系统、全面、快速地掌握 C 语言程序设计方法,为后续应用开发建立坚实的语言基础。

(2) 针对性强。针对二级独立院校本科学生重实践的学习特点,兼顾全国和江苏省二级等级考试大纲(C 语言版)的要求,采用先理论后实践的学习规律,变抽象为具体的学习策略,增强学习的目的性,合理组织 C 语言知识点和相关考点。

(3) 科学组织。全书包括程序设计基础、C 语言程序设计基础和 C 语言程序设计能力三大部分,力求做到结构严谨,概念准确,教材内容组织合理,语言使用规范,符合教学规律。

(4) 注重能力。围绕能力培养的目标,从易于读者学习的角度出发,以实例和易接受的图形方式阐述枯燥的理论知识,并在实例讲解中详细列出问题分析、流程图、C 语言程序代码、程序分析和运行结果等内容,目的是帮助学习者学会编程,切实提高其应用能力。

本书循序渐进,按由方法到实践、由入门到掌握的思路,分为三大篇:第 1 篇为程序设计基础篇,为初学者介绍程序设计的方法、算法、开发环境和相关的程序设计基础知识等;第 2 篇为 C 语言程序设计基础篇,从最简单的 C 语言程序开始,逐步地介绍 C 语言程序的构成要素,使略有计算机基础知识的读者都能很容易地学会 C 语言编程;第 3



篇为 C 语言程序设计能力篇,主要介绍 C 语言中的数组、指针、函数以及文件操作等内容。

本教材再版工作由南京信息工程大学滨江学院耿焕同教授主持,并负责对全书进行统稿和主审。其中,耿焕同老师负责第 1 至 4 章;陈遥老师负责第 6 至 8 章;朱节中老师负责第 5,11 和 12 章;李振宏老师负责第 9,10,13 章和附录。

本书在编写过程中,得到诸多专家和领导的有力指导与支持,在此表示衷心的感谢。

限于编者水平有限,书中难免有错误和不足之处,恳请专家和广大读者批评指正。

编 者

2013 年 8 月

内容提要

随着信息技术的飞速发展,掌握一门编程语言是社会对现代科技人才一种内在需求,更是成为现代理工类学生应熟练掌握的一项基本技能。在国内外 C 语言作为结构化程序设计语言的典型代表,非常适合作为掌握程序设计技术的入门语言,更为重要的是,C 语言在应用上的特点不仅适合软件开发,而且能够直接对各类可编程器件进行底层操作。

本书为《C 语言程序设计》的第 2 版,既保留了第 1 版在内容组织与设计上的优点,解决了在教材使用过程中出现的错误与不妥之处,又对部分内容进行了取舍,增加了对等级考试相关考点的设计。与此同时,针对学生上机实践和自学时缺乏合适的配套指导书问题,精心设计并组织编写了《C 语言程序设计实验指导与习题集》一书,并与第 2 版同步出版。

本书仍以 ISO C89 语言规范为蓝本,以 Visual C++ 6.0 为实践环境,循序渐进、深入浅出、系统全面地讲解了从语法到问题编程求解的各个环节,内容包括程序设计理论基础篇、C 语言程序设计基础篇、C 语言程序设计能力篇等。

本书紧紧结合 C 语言的学习方法和学生的学习特点,科学设计、精心组织教程内容,以浅显易懂的语言进行撰写,并配有大量的图解、例题和程序实例等,力争让非计算机专业人员也能快速地理解和掌握编程的技巧与精髓,具备独立使用 C 语言进行编程的基本技能。

作为一本计算机语言类的教材,教材设计过程中形成了鲜明的特点和创新之处:以二级独立学院本科学生为主要学习对象,进行教学内容的科学选择和巧妙编排;以学生编程能力的培养为设计重点,不陷于繁琐而使用少的细枝末节;以表述易于理解为编写理念,巧妙选择典型例题,并配合问题分析及程序分析等内容,因此最大限度地丰富了教材内涵,提高学生的学习兴趣和主动性,从而使学生能系统、全面、快速地掌握 C 语言程序设计方法。

本书语言表达严谨、流畅、实例丰富,不仅非常适合二级独立学院本科理工类学生的学习教程,也非常适合其他层次学生的学习教材,还可作为 C 语言编程人员的自学教材和全国计算机等级考试(C 语言)的参考教材。

目 录

第1篇 程序设计基础

第1章 程序设计方法学

1.1 程序设计方法学简介	001
1.2 结构化程序设计方法	002
1.2.1 概 述	002
1.2.2 程序设计步骤	004
1.2.3 方法举例	005
1.3 面向对象程序设计方法	005
1.3.1 概 述	005
1.3.2 程序设计步骤	008
1.3.3 方法举例	009
习题1	010

第2章 算法——程序的关键

2.1 算法的含义及其特征	011
2.1.1 算法的由来	011
2.1.2 算法的含义	011
2.1.3 算法的特征	012
2.2 算法的表示	012
2.2.1 程序的3种基本结构	012
2.2.2 流程图及其表示	013
2.2.3 N-S图及其表示	014
2.3 简单算法举例	015
习题2	018

第3章 程序设计过程

3.1 高级语言与编译器	019
3.2 程序设计过程	020
习题3	023

第4章 相关的程序设计基础知识

4.1 基本的软硬件知识	024
4.1.1 基本的软件知识	024
4.1.2 基本的硬件知识	028



4.2 程序在内存中的布局	031
4.2.1 C 语言程序的存储区域	032
4.2.2 C 语言可执行程序内存布局	032
4.2.3 举例说明	034
4.3 源程序编写的一般规范	034
4.3.1 标识符命名及书写规则	034
4.3.2 注释及格式要求	035
4.3.3 缩进规则	036
4.3.4 代码的排版布局	036
4.3.5 函数的编写规范	037
习题4	037

第2篇 C语言程序设计基础

第5章 C语言基础

5.1 基本字符集、标识符、常量和变量	039
5.1.1 基本字符集	039
5.1.2 标识符	039
5.1.3 常量	040
5.1.4 变量	041
5.2 基本数据类型	042
5.2.1 整型数据	043
5.2.2 实型数据	045
5.2.3 字符型数据	046
5.3 运算符与表达式	049
5.3.1 算术运算符与表达式	049
5.3.2 逻辑运算符与表达式	050
5.3.3 关系运算符与表达式	052
5.3.4 自增、自减运算符	053
5.3.5 逗号运算符与表达式	053
5.3.6 赋值运算符及表达式	054
5.3.7 类型转换	055
5.4 位运算	057
5.4.1 按位与运算符	058
5.4.2 按位或运算符	059
5.4.3 按位异或运算符	060
5.4.4 按位取反运算符	061
5.4.5 按位左移运算符	061
5.4.6 按位右移运算符	062
5.4.7 位运算赋值运算符	062

5.5 综合程序举例	062
习题 5	064
第 6 章 顺序结构程序设计	
6.1 顺序结构概述	067
6.2 数据输出	069
6.2.1 printf 函数的一般调用形式	070
6.2.2 printf 函数的常用格式说明	070
6.2.3 使用 printf 函数的注意事项	072
6.2.4 putchar 函数输出字符	073
6.3 数据输入	074
6.3.1 scanf 函数的一般调用形式	074
6.3.2 scanf 函数的常用格式说明	075
6.3.3 使用 scanf 函数从键盘输入数据	076
6.3.4 使用 getchar 函数从键盘输入数据	077
6.4 综合程序举例	077
习题 6	079
第 7 章 分支结构程序设计	
7.1 分支结构概述	082
7.2 if 语句	083
7.2.1 单分支 if 语句	083
7.2.2 双分支 if...else 语句	085
7.3 多分支结构	087
7.3.1 嵌套的 if 语句	087
7.3.2 switch 语句	090
7.4 无条件转移语句	093
7.4.1 语句标号	093
7.4.2 goto 语句	093
7.5 综合程序举例	093
习题 7	097
第 8 章 循环结构程序设计	
8.1 循环结构概述	099
8.2 简单循环结构	099
8.2.1 while 循环	099
8.2.2 do...while 语句	102
8.2.3 for 语句	104
8.3 break 和 continue 语句	106
8.3.1 break 语句	106
8.3.2 continue 语句	108
8.4 循环的嵌套	109



8.5 综合程序举例	111
习题8	113

第3篇 C语言程序设计能力

第9章 数组

9.1 数组概述	115
9.2 一维数组	117
9.2.1 一维数组的定义	117
9.2.2 一维数组元素的引用	119
9.2.3 一维数组元素的初始化	120
9.2.4 应用举例	120
9.3 二维及以上高维数组	124
9.3.1 二维数组的定义	124
9.3.2 二维数组元素的引用	125
9.3.3 二维数组的初始化	126
9.3.4 其他高维数组	127
9.3.5 应用举例	128
9.4 字符串与字符数组	132
9.4.1 字符串的表示	132
9.4.2 字符串的输入与输出	133
9.4.3 字符串处理的函数	134
9.4.4 字符串数组	138
9.4.5 应用举例	140
9.5 综合程序举例	140
习题9	146

第10章 地址与指针变量

10.1 地址和指针概述	149
10.2 指针变量的定义	150
10.3 指针变量的赋值	150
10.4 指针变量的操作	152
10.4.1 指针引用	152
10.4.2 指针移动	154
10.4.3 指针比较	154
10.5 一维数组和指针	155
10.5.1 一维数组和数组元素的地址	155
10.5.2 指针与数组元素操作	155
10.6 二维数组和指针	158
10.6.1 二维数组和数组元素的地址	158
10.6.2 指针与数组元素操作	159

10.7 指针数组	161
10.8 字符指针	162
10.9 多级指针	164
10.10 动态内存分配	165
10.11 动态数组	167
10.12 综合程序举例	168
习题 10	174
第 11 章 函 数	
11.1 函数概述	176
11.2 函数的定义	179
11.2.1 函数定义	179
11.2.2 函数的返回值	180
11.3 函数的调用	180
11.3.1 函数调用语法	180
11.3.2 函数的嵌套调用	182
11.4 函数的声明	187
11.4.1 函数声明的形式	187
11.4.2 函数声明的位置	188
11.5 参数传递	189
11.5.1 值传递方式	189
11.5.2 地址传递方式	191
11.6 函数与数组	193
11.7 函数与指针	195
11.7.1 指针作为函数参数	195
11.7.2 指针型函数与函数指针	200
11.8 变量的作用域、存储类型和生存期	202
11.8.1 变量的作用域	202
11.8.2 存储类型和生存期	205
11.9 main 函数中的参数	208
11.10 带参数的宏与函数	209
11.11 综合程序举例	210
习题 11	212
第 12 章 结构体、共用体、枚举及用户定义类型	
12.1 结构体	216
12.1.1 定义结构类型和结构变量	216
12.1.2 访问结构体成员	219
12.1.3 结构体数组	221
12.1.4 结构体指针	222
12.1.5 链表	223



12.1.6 结构体与函数	232
12.2 共用体	234
12.2.1 共用体类型与共用体变量	234
12.2.2 共用体变量的引用	235
12.2.3 共用体指针	236
12.3 枚举与自定义类型	237
12.3.1 枚举类型	237
12.3.2 自定义类型	238
12.4 综合程序举例	239
习题 12	241
第 13 章 文件	
13.1 文件概述	244
13.1.1 文件名	245
13.1.2 文件分类	245
13.1.3 文件缓冲区	246
13.2 文件类型指针	246
13.2.1 文件的存取方式	247
13.2.2 文件的定位	247
13.3 文件的打开与关闭	247
13.3.1 文件打开	248
13.3.2 文件关闭	249
13.4 顺序读写文件	250
13.4.1 字符读写	250
13.4.2 字符串读写	252
13.4.3 数据块读写	254
13.4.4 格式化读写	256
13.5 随机读写数据文件	257
13.6 文件读写出错检测函数	259
13.7 综合程序举例	260
习题 13	263
附 录	
附录 A 标准 ASCII 码表	265
附录 B C 语言的关键字	266
附录 C C 语言常用语法提要	267
附录 D C 语言常用的标准库函数	270
附录 E C 语言运算符优先级	273
参考文献	275

第1篇 程序设计基础

第1章 程序设计方法学

常言道“方法决定效率”。对于计算机初学者来说,C 语言程序设计是一门非常重要的基础课程,对其掌握的效果如何直接影响学习者后续课程的学习以及对编程的兴趣。因此,了解必要的程序设计方法学基本知识就显得非常重要,同时对其他编程课程学习也有很好的借鉴作用。

1.1 程序设计方法学简介

众所周知,随着科技和信息技术的迅猛发展,越来越多的工作和业务由程序进行支撑与管理,如数值天气预报、数字化校园以及网上购物等。那究竟什么是程序呢?通俗来说,程序是用来控制计算机操作的一系列代码,而程序设计的目的是利用计算机对现实问题进行求解。计算机科学家、图灵奖获得者尼克劳斯·威茨(Niklaus Wirth)教授对程序进行了经典定义:

程序 = 算法 + 数据结构

此公式对计算机科学的影响程度类似于物理学中爱因斯坦的 $E = mc^2$,它揭示了程序的本质。

随着软件产业的迅猛发展和软件开发工程化进程加快,程序与软件开发环境的关系越来越紧密,开发工具的选择对程序的开发效率有重大的影响,有时会取得事半功倍的效果。因此,人们对程序的定义进行了扩充:

程序 = 算法 + 数据结构 + 开发环境

本书重点讨论 C 语言开发环境,而不是复杂的算法和数据结构,目的是让读者利用 C 语言设计简单程序,建立与计算机之间的会话交流,以培养具有一定编程思想和程序设计能力的软件人才。

程序设计方法学是探讨程序设计理论和方法的学科,用以指导程序设计各阶段工作的原理和原则,以及依此提出的设计技术。程序设计方法学起源于 20 世纪 70 年代,主要包括程序理论、研制技术、支持环境、工程规范和自动程序设计等课题。程序设计方法学的发展、软件的发展以及编程语言的发展三者之间有着密切的关系,通过对其研究,可不断地提高编程人员的程序设计水平,丰富编程人员的思维方法;问题求解规模



和复杂性大大地促进了程序设计技术的发展,而程序设计的提高也推动着程序设计方法学这一学科不断发展。

通常而言,程序设计方法学的概念有狭义和广义之分。狭义的程序设计方法学是指传统的有关结构化程序设计的理论、方法和技术;广义的程序设计方法学包括程序设计语言和程序设计的所有理论和方法。特别是在结构化程序设计的研究逐步衰退以后,程序设计方法学已成为一个笼统的概念。随着软件产业的快速发展,人们对程序设计方法提出了更高的要求,如设计过程简单化、代码跨平台化、代码重用化等,这促使程序设计方法学成为一门学科。因此,学习者首先要弄清楚程序设计方法学的基本研究目标。

从学科定义来说,程序设计方法学的目标是设计出可靠、高效、易读且代价合理的程序。更通俗地说,程序设计方法学的最基本目标是通过程序本质属性的研究,说明什么样的程序是一个“优秀”的程序,怎样才能设计出“优秀”的程序。

一般的程序设计过程是通过借助某种编程语言对求解问题的计算机算法进行编程实现的,其产出是软件产品(俗称程序),功能是利用计算机求解问题,因此在程序设计时,最重要的是程序的正确性和执行效率。程序的正确性和执行效率是由程序的结构和算法决定的,当然也与程序的易读性、可维护性有密切的关系。程序设计的一般过程应包括:分析实际问题并抽象,利用数学建模技术构建问题的数学模型,借助计算方法和数据模型构造合适的数据结构,进而设计算法,最后借助计算机语言实现算法并形成程序。

程序设计方法大致经历手工作坊式、结构化、模块化、面向对象等程序设计阶段。下面以主流的结构化程序设计和面向对象程序设计为例,分别讲解它们的设计方法。

1.2 结构化程序设计方法

1.2.1 概述

迪克斯特拉(E. W. Dijkstra)在1969年提出了结构化程序设计方法,即以模块化设计为中心,将待开发的软件系统划分为若干个相互独立的模块,使完成每一个模块的工作变得简单且明确,这为设计一些较大的软件打下了良好的基础。由于模块相互独立,因此在设计其中一个模块时不会受到其他模块的牵连,从而可将原来较为复杂的问题化简为一系列简单模块的设计。模块的独立性还为扩充已有的系统、建立新系统带来了极大的方便,因为人们可以充分利用现有的模块进行积木式的集成与扩展。

结构化程序的概念首先是从以往编程过程中无限制地使用转移语句而提出的。转移语句可以使程序的控制流程强制性地转向程序的任一处,在传统流程图中,用“很随意”的流程线来描述转移功能。如果一个程序中多处出现这种转移情况,将会使程序流程无序可寻,程序结构杂乱无章,这样的程序是让人难以理解和接受的,并且容易出错。尤其是在实际软件产品的开发中,更多追求的是软件的可读性和可修改性,像这种结构和风格的程序是不允许出现的。因此,人们提出了程序的3种基本结构。

算法的实现过程是由一系列操作组成的,这些操作之间的执行次序就是程序的控