

21
世纪

应用型本科计算机科学与技术专业规划教材

C 语言程序 设计教程



葛雷	王怀柱	主 编
吴铁峰	李涛 康婧	等 副主编
周洪玉		主 审



清华大学出版社

九江学院图书馆



1822823

1524329

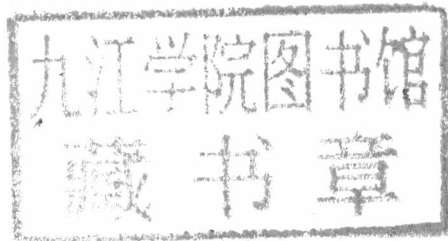
21 世纪应用型本科计算机科学与技术专业规划教材

C 语言程序设计教程

主 编：葛 雷 王怀柱
副主编：吴铁峰 李 涛 康 婧 李 云 王宏欣
主 审：周洪玉

TP312/21333

不外借



清华大学出版社
北 京

0281571

内 容 提 要

本书紧密围绕结构化程序设计思想,以案例驱动的方式深入浅出地引领学生掌握 C 语言程序设计知识。全书内容系统,理论通俗易懂,案例丰富、实用,大量习题方便复习和考核。

全书共分 13 章,内容包括 C 语言程序设计概述、数据类型、运算符与表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、指针、函数、预处理命令、结构体与共同体、位运算、文件和综合案例等;最后提供的综合案例——学生成绩管理系统,在贯穿结构化程序设计思想的同时,综合应用了 C 语言程序设计的知识。本书以 Visual C++ 6.0 作为开发、调试环境,书中案例都提供了完整的源代码并调试通过,方便读者边学边做。

本书适合作为高等学校各相关专业“C 语言程序设计”课程的教材,也可供 C 语言自学者、程序设计培训学员及工程技术人员学习使用。

为了方便教学和读者上机操作练习,本书配备了电子教案、案例源代码、课后习题题解和习题库等内容,可在清华大学出版社网站上下载。

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 语言程序设计教程/葛雷,王怀柱主编. —北京:清华大学出版社,2012.1

(21 世纪应用型本科计算机科学与技术专业规划教材)

ISBN 978-7-302-26480-4

I. ①C… II. ①葛… ②王… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 166423 号

责任编辑:索 梅

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:010-62795954, jsjic@tup.tsinghua.edu.cn

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

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:22.5 字 数:557 千字

版 次:2012 年 1 月第 1 版 印 次:2012 年 1 月第 1 次印刷

印 数:1~3000

定 价:34.00 元

产品编号:040646-01

编写委员会成员名单

名誉主任：李建中

主任：郝忠孝

副主任：周洪玉

委	员：	王乃茂	王培东	王国权	苏晓东	周屹
		牛皖闵	马慧彬	朱景富	黄凤岗	陈本土
		李晓峰	高巍巍	李人贤	常键斌	刘丕娥
		韩凤来	杜凯	王克家	孙斐朗	贾宗福
		马英瑞	马春华			

秘书：马宪敏



21

世纪是信息产业大发展的时代,计算机技术成为信息社会的重要支柱。信息化社会对人才的培养提出了更高的要求 and 标准。掌握计算机技术并具有应用计算机的能力是适应信息化社会的基础。

这套计算机系列教材适用于培养应用型人才,突出实验教学,突出实用,培养学生动手能力,掌握最新技术,适应社会需求。

本套教材在编写模式上和思路有了较大变化,采取面向任务,面向目标,先提出问题,然后指出解决问题的方法和所需要的知识的项目驱动式教材编写指导思想。针对目标,明确任务,做什么项目,用什么知识;用什么,学什么,学什么,会什么;急用先学,学以致用;突出重点,突出有用;然后由此及彼,由表及里,由浅入深,先感性,后理性,先实践,后理论,先认识,后提高;先掌握基本应用,然后做理论讲解、扩展与延伸,最后落实到具体操作,指导学生动手设计,用实践检验对知识的掌握程度。

本套教材特点是:内容丰富,知识全面,项目驱动,图文并茂,案例教学,贯彻始终。结构严谨,层次分明,条理清晰,通俗易懂,由浅入深,深入浅出,循序渐进。减少交叉,避免重复,编排合理,精心设计,突出重点,化解难点。学习理论,上机实验,举一反三,学用结合,配备习题,提供试题,联系实际,提高能力。

我们从计算机技术的发展趋势和信息社会对人才培养的需求出发,实现知识传授与能力培养的有效结合,通过对教学内容的基础性、科学性和应用性的研究,体现以有效知识为主体,构建支持学生终身学习的计算机知识基础和能力基础,提高学生计算机的应用能力。本系列教材强调理论与实践相结合,既注重基本原理、基本概念的介绍,又注重基本操作、基本能力的培养,根据计算机技术的发展和应用,加重了项目实训的内容。提高学生的动手能力。本套教材由三个部分组成,一是教材本身,二是实践实验教程,三是配套电子课件和素材(可到清华大学出版社网站 www.tup.com.cn 上下载)。

教育是科学,其价值在于求真。教育是艺术,其生命在于创新。大学教育真正要教会学生的应该是学习精神、学习能力、应用和创新能力。学习应该是超越课本知识的一个过程。本系列教材内容广泛新颖、取材丰富实用、阐述深入浅出、结构合理清晰。本系列教材的出版,不仅是编者们的努力的结果,同时也凝结了编委会许多人的心血,清华大学出版社的编辑们为系列教材的出版任劳任怨、一丝不苟。因此,本系列教材的出版是集体智慧的结晶,是各院校优势互

补、突出学校特色、进行计算机应用型人才培养的一次有益尝试。在此,编委会向所有为本系列教材的出版付出辛勤劳动的教师们及清华大学出版社的同仁们表示崇高的敬意和衷心的感谢!本系列教材在编写过程中也得到黑龙江省教育厅的悉心指导以及许多高校的大力支持,特别是黑龙江外国语学院院长邓中兴教授给予了热情帮助和大力支持,也得到了许多计算机公司的帮助,编委会在此向他们表示衷心感谢!

本系列教材既可作为高等学校计算机专业的教材,也可作为信息技术的培训教材或参考书。

由于时间仓促,书中粗浅疏漏或叙述欠严密之处在所难免,恳请读者批评指正,热切期待着授课教师在教学实践中对系列教材提出宝贵意见和建议。我们将每年对系列教材进行一次认真的修订。

郝忠孝

前言

FOREWORD



C语言是世界上应用最为广泛的程序设计语言之一,有着其他程序设计语言没有的天生优势,同时一些优秀的程序设计语言也衍生自C语言,可见,C语言在程序设计中占据着非常重要的地位。因此,几乎所有高校的理工类专业只要开设程序设计相关课程,大多会选择C语言作为程序设计课程的入门语言,计算机及相关专业更是如此。

C语言程序设计课程发展数十年,相关教材也层出不穷,各具特色,其中有很多非常优秀的精品教材。由于这些教材各有定位,也有难以全面顾及的地方。一方面有些教材中C语言知识讲授有余,而程序设计思想和应用实践阐述不足,导致学生学完C语言之后对程序设计的理解仍然支离破碎,难以形成宏观的整体认识;另一方面,许多教材缺乏从计算机硬件角度理解程序设计语言的相关阐述,使得作为程序设计的初学者在学习C语言低级特性如指针等相关内容时难以透彻理解。

本书在两个方面做出了努力和尝试。一方面,为了帮助学生对程序设计形成较为清晰的整体认识,本书首先对结构化程序设计思想及程序设计基础知识做了较为完整的阐述,后续各章也紧密围绕结构化程序设计思想这一主要线索,以案例驱动的方式做到了理论与实践的深度融合,全书最后还提供了一个较为完整的应用案例,从而实现了结构化程序设计从思想到实践的一脉贯通。另一方面,本书从冯·诺依曼计算机体系结构入手,阐述了计算机程序运行与计算机硬件之间的紧密联系,以比较通俗的方式帮助学生从硬件的角度来理解计算机程序的运行。当然,本书大部分篇幅仍然是对C语言程序设计基本知识的阐述,其中也吸收和借鉴了许多优秀C语言教材的思想和内容。

本书是集体智慧的结晶,参加本书编写的都是长期工作在高校教学、科研一线的经验丰富的教师。本书由葛雷、王怀柱任主编,吴铁峰、李涛、康婧、李云、王宏欣任副主编,周洪玉任主审,参加本书编写的还有王斌、于虹博、赵佳华、马宪敏、郝钢等。本书共分13章,具体分工为:第1、第12、第13章由葛雷编写;第2章由李涛编写;第3章由王宏欣编写;第4章由李云编写;第5章由康婧编写;第6章由于虹博、赵佳华编写;第7和第10章由王怀柱编写;第8章的8.1~8.8节由吴铁峰编写,8.9~8.11节由郝刚编写;第9和第11章由王斌编写;附录由马宪敏编写,全书由葛雷、马宪敏统一编排定稿。

感谢对本书编写给予支持的各高校领导和专家,感谢周洪玉教授对本教材编写的悉心指导,感谢为本教材的编写、出版提供支持、帮助的老师 and 朋友们,感谢对本教材提出宝贵建议和批评的老师和同学们。

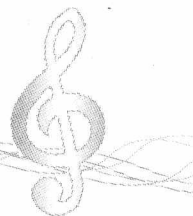
由于编者水平有限,时间较为仓促,疏漏和不妥之处在所难免,殷切希望读者及同行专家批评指正,共同为学生打造更为适用的教材。编者联系方式:葛雷 E-mail: goldenlane@163.com。

编 者

2012年1月

目 录

CONTENTS



第 1 章 概述	1
1.1 计算机与程序	1
1.1.1 冯·诺依曼计算机模型	1
1.1.2 存储程序思想	3
1.1.3 程序——计算机的灵魂	3
1.2 程序设计语言概述	4
1.2.1 程序设计语言及分类	4
1.2.2 编译与解释	5
1.3 Visual C++ 6.0 编译环境的使用	6
1.3.1 Visual C++ 6.0 开发环境简介	6
1.3.2 Visual C++ 6.0 启动及界面	6
1.3.3 打开/新建 C 语言源程序文件	7
1.3.4 程序保存	7
1.3.5 编译、连接和执行程序	7
1.4 C 语言程序设计概述	9
1.4.1 C 语言简介	9
1.4.2 C 语言源程序的组成元素	9
1.4.3 C 语言源程序的结构	12
1.4.4 C 语言多文件组织	14
1.5 面向过程(结构化)程序设计方法概述	15
1.5.1 面向过程程序设计的基本步骤	15
1.5.2 数据与数据结构	16
1.5.3 算法及其描述	17
本章小结	20
习题 1	21
第 2 章 数据类型、运算符与表达式	22
2.1 C 语言的数据类型概述	22
2.2 标识符、常量和变量	22
2.2.1 标识符	23
2.2.2 常量和符号常量	23
2.2.3 变量	24
2.3 C 语言的简单数据类型	25

2.3.1 整型数据	26
2.3.2 实型数据	29
2.3.3 字符型数据	31
2.4 运算符和表达式	34
2.4.1 C 运算符和表达式概述	34
2.4.2 算术运算符与算术表达式	35
2.4.3 关系运算符与关系表达式	36
2.4.4 逻辑运算符与逻辑表达式	37
2.4.5 条件运算符与条件表达式	39
2.4.6 赋值运算符与赋值表达式	40
2.4.7 逗号运算符与逗号表达式	42
2.4.8 位运算符与位运算表达式	43
2.4.9 求字节运算符	43
2.4.10 运算符及其优先级小结	44
2.5 数据的类型转换	44
2.5.1 系统自动进行的类型转换	44
2.5.2 强制类型转换	45
本章小结	46
习题 2	47
第 3 章 顺序结构程序设计	50
3.1 C 语言语句概述	50
3.1.1 C 语言程序结构	50
3.1.2 C 语言语句分类	51
3.2 输入输出函数语句	52
3.2.1 流	52
3.2.2 字符输入输出函数	52
3.2.3 格式输入输出函数	53
3.3 顺序结构程序举例	59
本章小结	61
习题 3	61
第 4 章 选择结构程序设计	63
4.1 if-else 语句结构	63
4.1.1 if-else 语句结构的三种形式	63
4.1.2 if 语句的嵌套	69
4.1.3 条件运算符和条件表达式	71
4.2 switch 语句结构	73
4.3 选择结构程序举例	79
本章小结	81
习题 4	81

第 5 章 循环结构程序设计	87
5.1 while 语句	87
5.1.1 while 语句的一般形式	87
5.1.2 while 语句的执行过程	87
5.1.3 while 语句的几点说明	90
5.2 do-while 语句	92
5.2.1 do-while 语句的形式	92
5.2.2 do-while 语句的执行过程	92
5.2.3 do-while 语句的几点说明	94
5.2.4 while 语句与 do-while 语句的比较	94
5.3 for 语句	95
5.3.1 for 语句的一般形式	95
5.3.2 for 语句的执行过程	96
5.3.3 for 语句的几点说明	96
5.4 循环嵌套	99
5.5 goto 语句、break 语句、continue 语句	103
5.5.1 goto 语句	103
5.5.2 break 语句	103
5.5.3 continue 语句	105
5.6 经典例题	106
本章小结	111
习题 5	111
第 6 章 数组	117
6.1 数组的概念	117
6.1.1 数组的引入	117
6.1.2 数值数组和字符数组	118
6.2 一维数组	118
6.2.1 一维数组的定义	118
6.2.2 一维数组元素的引用	119
6.2.3 一维数组的初始化	119
6.2.4 一维数组程序举例	119
6.3 二维数组	122
6.3.1 二维数组的定义	123
6.3.2 二维数组元素的引用	124
6.3.3 二维数组的初始化	124
6.3.4 二维数组程序举例	126
6.4 字符数组	129
6.4.1 字符数组的定义	129
6.4.2 字符数组元素的引用	129

6.4.3	字符数组的初始化	129
6.4.4	字符数组的输入输出	130
6.4.5	字符串处理函数	132
	本章小结	138
	习题 6	138
第 7 章	指针	141
7.1	变量的地址和指针变量	141
7.1.1	变量的地址以及变量的存取方式	141
7.1.2	指针变量的定义和指针变量的基类型	142
7.1.3	给指针变量赋值	142
7.1.4	指针变量的引用	144
7.2	数组的指针与指向数组的指针变量	149
7.2.1	指向数组的指针变量	149
7.2.2	通过指针引用数组元素	150
7.2.3	指向多维数组的指针和指针变量	152
7.3	字符串的指针和指向字符串的指针变量	155
7.3.1	字符串的表示形式	155
7.3.2	字符数组与字符串指针的区别	156
7.4	指针数组与指向指针的指针	158
7.4.1	指针数组的概念	158
7.4.2	指向指针的指针	159
7.5	指针与动态内存分配初步	161
	本章小结	164
	习题 7	164
第 8 章	函数	170
8.1	函数概述	170
8.1.1	模块化程序设计	170
8.1.2	函数的分类	170
8.2	函数的定义	172
8.3	函数间的参数传递	173
8.3.1	形式参数和实际参数	173
8.3.2	函数的返回值	175
8.4	函数的调用	178
8.4.1	函数调用的形式	178
8.4.2	函数调用的方式	179
8.4.3	被调函数的声明	180
8.5	数组作函数的参数	182
8.6	函数与指针	193
8.6.1	指针作为函数的参数	193

8.6.2 指针型函数	203
8.6.3 指向函数的指针	205
8.7 函数的嵌套调用	207
8.8 函数的递归调用	209
8.9 局部变量和全局变量	214
8.9.1 局部变量	215
8.9.2 全局变量	216
8.10 变量的存储类别	218
8.10.1 动态存储方式与静态存储方式	218
8.10.2 自动变量	219
8.10.3 静态局部变量	219
8.10.4 寄存器变量	221
8.10.5 外部变量	221
本章小结	222
习题 8	222
第 9 章 预处理命令	229
9.1 宏定义	229
9.1.1 无参宏定义	230
9.1.2 有参宏定义	235
9.2 文件包含	239
9.3 条件编译	243
本章小结	250
习题 9	250
第 10 章 结构体与共用体	253
10.1 结构体类型	253
10.1.1 结构体类型与结构体变量	254
10.1.2 结构体变量的引用	257
10.1.3 结构体变量的初始化	258
10.2 结构体数组	259
10.2.1 结构体数组的定义与引用	259
10.2.2 结构体数组的初始化	260
10.3 指向结构体的指针变量	260
10.3.1 结构体指针变量的定义	261
10.3.2 结构体和函数调用	265
10.4 链表	267
10.4.1 链表概述	267
10.4.2 处理链表的函数	267
10.4.3 链表的建立	268

10.4.4	链表的遍历	271
10.4.5	链表的插入	271
10.4.6	链表的删除	273
10.5	共用体	275
10.5.1	共用体类型定义	275
10.5.2	共用体变量的引用	276
10.6	枚举类型	278
10.7	用 typedef 定义类型	280
10.7.1	typedef 的用法	280
10.7.2	typedef 的基本功能	280
10.7.3	typedef 使用说明	280
本章小结		281
习题 10		281
第 11 章 位运算		285
11.1	位运算符及其功能	285
11.1.1	位运算符	285
11.1.2	位运算符的运算功能	286
11.2	位域结构	291
11.2.1	位域	291
11.2.2	位域程序举例	293
本章小结		294
习题 11		294
第 12 章 文件		296
12.1	流及文件的相关概念	296
12.1.1	流的概念	296
12.1.2	文件的概念及分类	297
12.2	文件类型指针及文件的打开与关闭	298
12.2.1	文件类型指针	298
12.2.2	文件的打开与关闭	298
12.3	文件的顺序读写	300
12.3.1	文件的字符读写	300
12.3.2	文件的字符串读写	304
12.3.3	文件的格式化读写	306
12.3.4	文件的记录读写	308
12.4	文件的随机读写	309
12.5	文件检测函数	311
本章小结		311
习题 12		312

第 13 章 综合案例：学生成绩管理系统	314
13.1 学生成绩管理系统需求分析	314
13.1.1 系统功能需求描述	314
13.1.2 需求分析	315
13.2 系统设计	320
13.2.1 模块设计	320
13.2.2 数据结构设计	322
13.2.3 系统界面设计	323
13.3 详细设计与编码实现	324
13.3.1 程序结构详细设计	324
13.3.2 编码实现	325
本章小结	335
习题 13	335
附录 A C 语言 ASCII 表	336
附录 B C 语言关键字及用途	337
附录 C 运算符和结合性	338
附录 D 常用的 ANSI C 库函数表	339
参考文献	341

概 述

本章学习目标：

- 理解计算机及程序运行原理；
- 了解程序设计语言基本知识；
- 掌握 Visual C++ 6.0 基本用法；
- 掌握 C 语言概要知识；
- 熟悉结构化程序设计方法及相关概念。

本章首先讲述了冯·诺依曼计算机模型及存储程序思想,说明了计算机程序运行的基本原理,然后介绍了程序设计语言的相关知识,重点介绍了 VC++ 6.0 开发环境的使用方法和 C 语言程序设计的概述知识,最后概要介绍了结构化程序设计方法及相关概念。

1.1 计算机与程序

1.1.1 冯·诺依曼计算机模型

在很久以前人类就发明了替代和延伸自身体力劳动的各种工具,但直到 20 世纪 40 年代,人类才真正找到可以部分的替代和延伸脑力劳动的工具——计算机,它是人类最伟大的发明之一,它使人类进入了一个全新的信息化时代。

体力劳动直接作用于客观事物,进而改造客观世界,那么脑力劳动如何作用于现实世界? 计算机作为人类脑力劳动的替代或延伸工具,应该具有怎样的功能和构成?

假设一个场景:一个原始部落有 N 个家庭,猎手们一次打猎获得了许多相同的猎物,回来后首领要按照规则分配猎物,规则为:如果猎物数量小于或等于家庭个数,则按家庭数量平均分配猎物;如果猎物数量大于家庭数量,则每个家庭分配一个猎物后,剩余猎物按参与狩猎的劳动力人数平均分配。我们可以想象一下首领分配猎物的过程。

首先,首领要搞清猎物总数量 L 、部落中家庭个数 N ;参与狩猎的劳动力总人数 Z ,以及每个家庭参与人数 R_i ($i=1,2,\dots,N$),这些数要么记在首领的大脑中,要么记在某处。最后要得出的结果是第 i 个家庭应得的猎物数 W_i ($i=1,2,\dots,N$)。

然后,首领要经过多步的判断和计算:

如果猎物数量 L 小于或等于家庭数量 N ,则:每个家庭的猎物数 = 猎物数量/家庭数量,否则(即猎物数量大于家庭数量),从 1 到 N 重复以下运算 N 次,并记下结果:

第 i 个家庭猎物数 $W_i = 1 + (L - N) / Z \times R_i$

最后,首领给出每个家庭应得的猎物数量,分配完成。

这是一个典型的脑力劳动的过程模型,从这个过程中我们可以分析出一台作为数据处理设备的计算机需要具备的几种功能:

(1) 具有存储、记忆数据和信息的功能。也就是能够从待处理的事物中获得数据和信息并进行表示和存储,处理过程的中间结果和最后结果也要记忆和存储;因此计算机需要有数据和信息的存储功能,也就是需要具有数据存储的部件。

(2) 具备获取和输出数据及信息的手段和功能。待处理问题的数据和信息要通过一定的方式进入信息处理装置(类似于人脑),还要把处理结果通过一定方式反馈给现实世界。比如首领通过眼睛观察搞清猎物的数量,或者耳朵听别人说出猎物的数量,通过嘴巴说出分配结果等。因此计算机需要具备数据输入部件和数据输出部件。

(3) 具有进行加、减、乘、除以及判断等算术、逻辑运算的功能。数据的处理可能包含算术运算、逻辑推理和判断等运算。因此,计算机需要具备一个算术逻辑处理的部件。

(4) 具有按照特定步骤控制各个部分共同协作的功能。比如,首领应该很清楚第一步做什么,第二步做什么,什么情况执行或不执行某种操作,什么时候结束等,这需要一个控制部件来起作用。因此,计算机需要具备一个控制部件。

经过以上分析,不难得出一台计算机应该具有的部件:存储部件、运算部件、控制部件、输入部件和输出部件,这就是20世纪40年代,美籍匈牙利人冯·诺依曼提出的著名的冯·诺依曼计算机组成模型。现代计算机尽管性能飞速提升,样子千差万别,组成上也多种多样,但都没有从根本上突破这一经典模型。

冯·诺依曼提出计算机应该由五大功能部件组成,即运算器、控制器、存储器、输入设备和输出设备。组成原理图如图1-1所示(图中实线为数据流,虚线为控制流)。

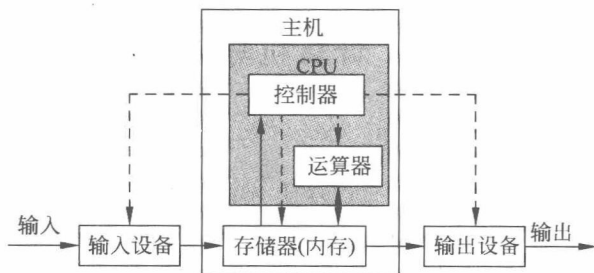


图 1-1 冯·诺依曼计算机模型

其中各部件的主要功能归纳如下:

- 运算器是计算机中执行算术运算和逻辑运算的功能部件;
- 控制器是控制计算机各个功能部件协调工作的功能部件,是神经中枢;
- 存储器是对数据或信息进行存储记忆的功能部件;
- 输入设备把外界数据或信号输入到计算机内部进行表示和存储的功能部件;
- 输出设备把计算机内部的数据或信息输出到计算机外部介质的功能部件。

这五大部件,与我们前面问题的分析是相互吻合的。可见,计算机具有这五大部件,也就具备了数据处理的能力。

思考: 既然控制器可以控制各个部件协调运行以解决问题,那么控制器进行控制的依据是什么呢? 它以什么形式存在,被存储在哪里呢?