



浙江省“十一五”重点建设教材



高等院校精品课程系列教材·省级

C语言程序设计与实践

凌云 吴海燕 谢满德 ©编著



*The Programming
and Practice in C*



机械工业出版社
China Machine Press



浙江省“十一五”重点建设教材

高等院校精品课程系列教材·省级

C语言程序设计与实践

凌云 吴海燕 谢满德◎编著



*The Programming
and Practice in C*



机械工业出版社
China Machine Press

本书由浅入深地讲授了 C 语言程序设计的技术与技巧。首先,介绍了 C 语言的基础语法知识;其次,通过项目开发全过程的全方位指导,从需求分析、算法设计到程序编写和过程调试,以项目实训的形式引导和帮助学生解决实际问题,提高学生解决具体问题的能力,并对程序设计竞赛中的常见算法及其应用进行了介绍;最后,介绍了编程风格与程序调试方法。

本书内容齐备、自成一体,可作为计算机及相关专业的本科或专科教材,也可以作为信息类或其他相关专业的辅助教材。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

C 语言程序设计与实践/凌云,吴海燕,谢满德编著. —北京:机械工业出版社, 2010.8

(高等院校精品课程系列教材)

ISBN 978-7-111-31007-5

I. C… II. ①凌… ②吴… ③谢… 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 (2010) 第 113007 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:李俊竹

三河市明辉印装有限公司印刷

2010 年 9 月第 1 版第 1 次印刷

184mm × 260mm · 19.25 印张

标准书号:ISBN 978-7-111-31007-5

定价:33.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991; 88361066

购书热线:(010) 68326294; 88379649; 68995259

投稿热线:(010) 88379604

读者信箱:hzjsj@hzbook.com

出版者的话

机械工业出版社华章公司秉承“全球采集内容，服务中国教育”的理念，经过十余年的不懈努力，引进、翻译、出版了大量在计算机科学界、电子科学界享有盛名的专家名著与名校教材，其中包括 Donald E. Knuth、Alfred V. Aho、Jim Gray、Jeffrey D. Ullman、R. Jacob Baker 等大师名家的一批经典作品，这些作品对国内计算机教育事业的发展起到了一定的推动作用。今天，全国高等学校精品课程建设工作的蓬勃开展为我们更好地服务于计算机教育、电子信息科学教育提供了良好的契机，我们将以严谨的治学态度及全面服务的专业出版精神，在国内广大院校老师们的支持与帮助下，陆续推出具有国内一流教学水平的“高等院校精品课程系列教材”。

精品课程是具有一流教师队伍、一流教学内容、一流教学方法、一流教材、一流教学管理等特点的示范性课程，是教育部实施的“高等学校教学质量与教学改革工程”的重要组成部分，是教育部深化教学改革，以教育信息化带动教育现代化的一项重要举措。自 2003 年精品课程建设项目持续推进以来，国内高校中的优秀教师纷纷在总结本校富有历史传统而又特色突出的课程教学方法与经验成绩的基础上，充分运用现代网络传播技术将优质的教学资源上网共享，使国内其他高校在实施同类课程教学的过程中能够借鉴、使用这些优质的教学资源，在更大范围内提高高等学校的教学和人才培养质量，提升我国高等教育的综合实力和国际竞争能力。经过几年的共同努力，已经建立起了较为齐全的各门类及各专业的校、省、国家三级精品课程体系，期间先后有总计 750 门课程通过了专家评审，获得了“国家精品课程”称号。

这些各个层次的“精品课程”建设过程都比较充分地体现了教育部所要求的七个重点，即：具有科学的建设规划，配备高水平的教学队伍，不断进行教学内容和课程体系的改革，使用先进的教学方法和手段，注重建设系列化的优秀教材，高度重视理论与实践两个环节，切实激励各方人员共同参与。也正因为这样的多方面积极参与，使得我国的高等教育在近年来由精英教育转向大众教育的跨越式发展中取得了教学

质量上的突破与飞跃。精品课教材作为精品课程的要件之一,比以往教材更加具有实践检验性,教学辅助资源经过不断地更新与补充更加丰富,是精品课教学团队智慧的共同体现。

“师者,所以传道授业解惑也”。教材是体现教学内容和教学要求的知识载体,是教师进行教学活动的基本工具,是提高教学质量的重要保证。精品课程教学团队中优秀的老师们集多年治学经验撰写出版相关教材,也是精品课程建设的一个重要方面。华章作为专业的出版团队,长久以来以“传承专业知识精华,服务中国教育事业”为使命,遵循“分享、专业、创新”的价值观,实践着“国际视野、专业出版、教育为本、科学管理”的出版方针,愿与高等院校的老师共同携手,为中国的高等教育事业走向国际化而努力。

为更好地服务于精品课程配套教材的出版,华章不仅密切关注高校的优秀课程建设,而且还将利用自身的优势帮助教师完善课程设置、提供教辅资料、准备晋级申报、推广教学经验。具体详情可访问专门网站 <http://www.hzbook.com/jpkc.aspx>,并可在线填写出版申请,欢迎您对我们的工作给予帮助和指导。

投稿专线:010-88379604

投稿 E-mail: hzjsj@hzbook.com



华章科技图书出版中心

前言

“C 语言程序设计”是一门理论与工程实践密切相关的专业基础课程,在计算机学科的教学中具有十分重要的作用。大力加强该课程的建设,提高该课程的教学质量,有利于教学改革和教育创新,有利于创新人才的培养。学生通过本课程的学习,可以培养良好的编程风格,掌握常见的算法思路,真正提高运用 C 语言编程解决实际问题的综合能力,为后续课程的实践环节的教学打好基础。

目前国内关于 C 语言的教材较多,有些教材语法知识介绍细致,较适合作为非计算机专业的等级考试类教学用书;有些教材起点较高,内容深奥,不适用于初学者。为了帮助广大学生更好地掌握 C 语言编程技术,我们组织浙江工商大学 C 语言程序设计课程组教师进行了深入的讨论和研究,针对学生学科竞赛和课时压缩的背景,将该课程的建设与其他信息类专业的课程体系改革相结合,发挥学院在计算机、电子商务和信息管理等专业上的办学优势,编写了本书。全书以程序设计为主线,采用了渐进式的体系结构,在详细阐述程序设计基本概念、原理和方法的基础上,结合实践教学和学科竞赛的实际情况,通过大量经典实例讲解和实训,使学习者掌握利用 C 语言进行结构化程序设计的技术和方法,培养和提高他们的实践动手能力和创新协作精神。

本书的框架结构分为三个部分。第一部分包括第 1~11 章,介绍 C 语言的基础语法知识,这部分内容按 C 语言的知识点循序渐进地介绍,同时,针对 C 语言中的重点和难点,如指针,精心设计了丰富的实例,用了大量的篇幅从不同方面对其进行讲解,帮助读者理解并掌握这些重点和难点。第二部分包括第 12~13 章,为项目实训和常用算法指导部分,这部分通过项目开发全过程的全方位指导,从需求分析、算法设计到程序编写和过程调试,以项目实训的形式引导和帮助学生实际问题,提高学生解决具体问题的能力,并对程序设计竞赛中的常见算法及其应用进行了介绍。第三部分即第 14 章,介绍编程风格与程序调试方法。

“C 语言程序设计”是一门强调实践练习的课程,对本教材的教学

组织可依据两条主脉络进行：从字、词、数据、表达式、语句到函数、指针结构、文件等，这也是语法范畴构成的基本脉络；从程序功能，即以组织数据和组织程序为另外一条基本脉络。安排课程内容时应注意以下几点：

1) 介绍程序设计语言语法时要突出重点。C 语言语法比较庞杂，有些语句可以相互替代，有些语法不常使用。课程中要重点介绍基本的、常用的语法，不要面面俱到。

2) 注重程序设计语言的共性。计算机的发展日新月异，大学期间不可能介绍所有的计算机语言。所以在本课程的学习过程中，教师应该介绍计算机程序设计语言共性的东西，使学生具有自学其他程序设计语言的能力。

3) 由于课时的限制，不能安排太多的时间专门讲授程序设计理论。在教学过程中，应以程序设计为主线，结合教材中的实例分析，将程序设计的一般方法和技术传授给学生。

本书由浅入深地讲授了程序设计的技术与技巧，内容齐备、自成一体，对启迪、提高读者的程序设计能力很有裨益，适用于不同层次的读者。本书可作为计算机及其相关专业的本科或专科教材，也可以作为信息类或其他相关专业的选修教材，还可以作为其他一些课程（如“数据结构”、“编译器设计”、“操作系统”、“计算机图形学”、“嵌入式系统”及其他要用 C 语言进行项目设计的课程）的辅助读物。

本教材的作者均为浙江工商大学承担程序设计、数据结构等课程的骨干教师，项目实践经验丰富，积累了不少的教学素材。本书由凌云负责全书的策划、组织，并对全书进行了统稿和校对，其中凌云编写了第 1、2 章，吴海燕编写了第 6、7、8、9、10、13 章，谢满德编写了第 3、4、5、11、12、14 章。

本教材也是浙江省精品课程“高级语言程序设计”的教学用书，课程教学小组同时注重立体化教材的建设，还配有多媒体电子教案、习题与实验指导，以及教学网站和教学资源库等。读者可以上网共享我们的网络资源，网址为 <http://e-lesson.zjgsu.edu.cn>。

在本书的编写过程中，参考了部分图书资料和网站资料，在此向其作者表示感谢。由于作者水平有限，书中难免出现遗漏和不足之处，恳请社会各界同仁及读者朋友提出宝贵意见和真诚的批评。

教学建议

教学内容	学习要点及教学要求	课时安排	
		计算机专业	非计算机专业
第1章 C语言与程序设计概述	了解指令与程序的概念。 了解程序设计的过程。 了解C语言的历史、特点及其程序结构。	2	2
第2章 例子驱动的C语言 语法元素概览	了解C语言的基本语法元素,包括变量与常量、算术运算、控制流、函数、数组、基本输入输出等,让学生对C语言有一个整体的感性认识,能编写简单的小程序。	2	2
第3章 基本数据类型和表达式	了解C语言的各种数据类型。 掌握整型常量、浮点常量、字符常量的表示法,以及各种运算符和表达式。	4	4
第4章 输入输出语句	掌握数据输出函数 (printf、putchar), 数据输入函数 (scanf、get-char), 熟练使用输入输出语句中常用的格式说明、控制字符串。	2	2
第5章 C程序结构	了解语句的分类、结构化程序设计的基本概念。 掌握循环、分支等控制语句的语法,并能熟练使用这些流程控制语句编写小的程序。	6	6
第6章 数 组	了解数组在内存中的表示方法。 掌握数组(一维、二维、字符数组)的定义、引用和应用,以及数组的典型应用实例,能利用数组编程解决实际问题。	6	6
第7章 函 数	了解基于函数的C语言程序组织方式。掌握函数的定义、函数的调用、函数参数的传递规则、函数的说明、变量的四种存储类别、变量的作用域和生命期。该章重点与难点在于基于函数参数的传递,嵌套函数和递归函数及其应用,变量的作用域和生命期及其应用。	6~8	6(选讲)
第8章 编译预处理	了解编译预处理的三种方式。 掌握包含和宏定义的使用方法。	2	2
第9章 指 针	了解地址的基本概念在C语言中的表示方法。 掌握变量和函数的地址在C语言中的表示法,以及指针变量的定义和引用,指针作为函数参数,指针与数组的关系,指针的运算,字符指针,字符串处理函数,指针数组和指向指针的指针,指向函数的指针,命令行参数的传递。	6~10	6(选讲)
第10章 结构与联合	掌握结构类型的定义,结构变量的说明和引用,结构数组的定义和应用,结构变量的参数传递规则,指向结构变量的指针,结构指针,链表的建立和链表元素的插入、删除、查找。掌握联合和枚举类型的定义及变量的定义使用。该章节的难点在于链表的基本操作。	6~10	6~8 (选讲)

(续)

教学内容	学习要点及教学要求	课时安排	
		计算机专业	非计算机专业
第11章 文件操作	了解文件、文本文件和二进制文件的概念,以及非缓冲文件的概念。 掌握缓冲文件指针的定义,缓冲文件的打开和关闭,缓冲文件读和写(文本文件方式、二进制文件方式)。	2	2
第12章 综合实训	通过项目开发过程的全方位指导,将所学的知识点串起来。该部分内容详细分析了几个实际项目的开发全过程,从需求分析到算法设计到程序编写到过程调试,通过实例指导,引导和帮助学生解决实际问题,提高学生解决具体问题的能力。	2	2~4 (选讲)
第13章 初涉 ACM/ICPC	结合程序设计大赛,将常见算法分为9大类,分别介绍这9大类算法及算法应用实例。	2~6 (选讲)	2(选讲)
第14章 编程风格与程序调试	对编程风格与程序调试中常见问题进行了阐述,帮助学生形成良好的程序设计风格,提高实际动手调试能力。	2(选讲)	2(选讲)
	教学总学时建议	50~64	50~54

说明:

- ①本书作为计算机专业的本科“C语言程序设计”课程的教材,建议课堂授课学时为50~64学时(包含习题课、课堂讨论等必要的课堂教学环节,实验另行安排学时),不同学校可以根据各自的教学要求和计划学时数酌情对教学内容进行取舍。其中,第12章实训部分可以选取其中一个例子详细讲解,其他例子让学生自学完成。第13章的内容可在开放实验教学中体现,第14章的内容贯穿在整个课程教学过程中。
- ②非计算机专业的师生在使用本书时可适当降低教学要求。第13章可以不介绍。若授课学时数少于50学时,则建议第6章中的递归,第9章中的链表部分内容可以适当简化。

课堂教学建议:

- ①本书的基础部分是第5~11章,这一部分从字、词、数据、表达式、语句到函数、指针等,是语法范畴构成的基本脉络。
- ②C语言程序设计是一门强调实践练习的课程,对本书的教学组织应以程序设计为主线,介绍程序设计语言语法时重点要突出,不要面面俱到。
- ③注重程序设计语言的共性。计算机的发展日新月异,大学期间不可能介绍所有的计算机语言。所以在本课程的学习过程中,教师应该介绍计算机程序设计语言共性的东西,使学生具有自学其他程序设计语言的能力。
- ④如果受学时的限制,第13章比较独立以及第14章都可以略去不讲。

实验教学建议:

本书各章节后面附有实验实例,可以参考这个来布置实验内容。

目 录

出版者的话

前言

教学建议

第1章 C语言与程序设计概述 1

1.1 初见C语言程序 1

1.2 计算机与程序设计 2

1.2.1 指令与程序 2

1.2.2 程序与程序设计 3

1.2.3 程序设计和程序设计语言 3

1.2.4 程序设计过程 4

1.3 C语言学习与自然语言学习的关系 5

1.4 C语言的发展历史、现状与特点 5

1.4.1 C语言的发展历史和现状 5

1.4.2 C语言的特点 6

习题 7

第2章 例子驱动的C语言语法元素

概览 8

2.1 变量与表达式 8

2.2 分支语句 9

2.2.1 if语句 9

2.2.2 switch语句 11

2.3 循环语句 12

2.3.1 while循环语句 12

2.3.2 for循环语句 12

2.4 符号常量 13

2.5 输入输出 14

2.6 数组 14

2.7 函数 15

2.8 算法 16

2.8.1 算法的概念 16

2.8.2 流程图与算法描述 17

习题 18

第3章 基本数据类型和表达式 19

3.1 基本语法单位 19

3.1.1 基本符号 19

3.1.2 关键字 19

3.1.3 标识符 20

3.2 数据类型 20

3.3 常量与变量 21

3.3.1 常量 21

3.3.2 变量 24

3.3.3 变量的初始化 25

3.4 表达式和运算符 26

3.4.1 算术运算符 26

3.4.2 赋值运算符 28

3.4.3 关系运算符 29

3.4.4 逻辑运算符 30

3.4.5 位运算符 32

3.4.6 逗号运算符 35

3.4.7 条件运算符 36

3.5 各类数值型数据间的混合运算 36

习题 37

第4章 输入输出语句 38

4.1 putchar函数 38

4.2 printf函数 39

4.2.1 printf函数的格式 39

4.2.2 格式说明字符 40

4.3	getchar 函数	45	6.1.3	一维数组元素的初始化	78
4.4	scanf 函数	46	6.2	二维数组	84
4.4.1	一般形式	46	6.2.1	双下标变量	84
4.4.2	格式说明	46	6.2.2	二维数组及其定义	85
4.4.3	scanf 函数的执行中应注意的问题	47	6.2.3	二维数组的初始化	86
4.5	程序举例	48	6.2.4	二维数组应用举例	87
	习题	49	6.3	综合应用举例	88
第5章	C 程序结构	51	6.4	字符数组	91
5.1	C 语句	51	6.4.1	字符串和字符串结束标志	92
5.2	程序设计基础	52	6.4.2	字符数组的输入输出	92
5.3	结构化程序设计的三种基本结构	53	6.4.3	字符串函数	94
5.3.1	顺序结构	53	6.4.4	二维字符数组	96
5.3.2	选择结构	53	6.4.5	字符数组应用举例	97
5.3.3	循环结构	54		习题	102
5.4	if 分支语句	55	第7章	函数	104
5.4.1	第一种 if 语句形式	55	7.1	函数的定义	105
5.4.2	第二种 if 语句形式	55	7.1.1	函数定义的格式	105
5.4.3	第三种 if 语句形式	56	7.1.2	形式参数和实际参数	106
5.4.4	if 语句的嵌套	58	7.1.3	函数的返回值	109
5.4.5	程序举例	60	7.2	函数的一般调用	110
5.5	switch 分支语句	61	7.2.1	函数调用的形式	110
5.6	while 循环语句	63	7.2.2	函数调用的方式	111
5.7	do-while 循环语句	64	7.2.3	主调函数和被调函数的相对位置关系	111
5.8	for 循环语句	66	7.2.4	函数调用时值的单向传递性	113
5.9	break 语句和 continue 语句	70	7.2.5	函数调用示例	114
5.9.1	break 语句	70	7.3	函数的嵌套调用	115
5.9.2	continue 语句	71	7.4	函数的递归调用	118
5.10	多重循环的嵌套	72	7.4.1	概述	118
5.11	程序举例	73	7.4.2	函数的递归调用应用举例	119
	习题	75	7.5	数组作为函数的参数	123
第6章	数组	77	7.5.1	数组元素作为函数的实参	123
6.1	一维数组	77	7.5.2	数组名作为函数的参数	123
6.1.1	一维数组的定义	77	7.5.3	多维数组作为函数的参数	124
6.1.2	一维数组元素的引用	78			

7.6 变量的作用域——局部变量和全局变量.....	125	9.5.1 多级指针的概念和使用.....	166
7.6.1 局部变量.....	125	9.5.2 多级指针和多维数组.....	167
7.6.2 全局变量.....	126	9.5.3 命令行参数.....	170
7.7 变量的存储类别和生存期.....	128	9.6 指针和动态存储管理.....	171
7.7.1 变量的存储类别.....	128	9.6.1 概述.....	171
7.7.2 动态变量.....	129	9.6.2 malloc 函数和 free 函数.....	172
7.7.3 静态变量.....	132	9.6.3 动态存储管理的应用.....	172
7.7.4 外部变量.....	133	9.7 指针和指针运算小结.....	174
7.8 内部函数和外部函数.....	136	习题.....	175
7.8.1 内部函数.....	136	第10章 结构与联合.....	177
7.8.2 外部函数.....	137	10.1 结构体类型变量的定义和引用.....	177
习题.....	138	10.1.1 结构体类型变量的定义.....	178
第8章 编译预处理.....	141	10.1.2 结构体类型变量的引用.....	179
8.1 宏定义.....	141	10.1.3 结构体类型变量的初始化.....	179
8.1.1 不带参数的宏定义.....	141	10.2 结构体数组的定义和引用.....	180
8.1.2 带参数的宏定义.....	143	10.3 结构体指针的定义和引用.....	181
8.2 文件包含.....	147	10.3.1 指向结构体类型变量的指针的使用.....	181
8.3 条件编译.....	149	10.3.2 指向结构体类型数组的指针的使用.....	182
8.3.1 条件编译语句1.....	149	10.4 链表的定义和操作.....	184
8.3.2 条件编译语句2.....	150	10.4.1 概述.....	184
8.3.3 条件编译语句3.....	150	10.4.2 链表的建立.....	185
习题.....	152	10.4.3 输出链表元素.....	187
第9章 指针.....	156	10.4.4 删除链表元素.....	187
9.1 地址和指针的概念.....	156	10.4.5 插入链表元素.....	188
9.2 指针变量和地址运算符.....	157	10.4.6 查询链表元素.....	188
9.2.1 指针变量的定义.....	157	10.5 联合.....	189
9.2.2 指针变量的使用.....	157	10.5.1 联合的定义.....	189
9.3 指针和数组.....	158	10.5.2 联合成员的使用.....	191
9.3.1 通过指针存取数组元素.....	159	10.5.3 应用举例.....	192
9.3.2 字符串和指针.....	160	10.5.4 数组、结构和联合三种数据类型 类型的比较.....	192
9.4 指针和函数.....	161	10.6 枚举类型.....	193
9.4.1 指针作为函数的参数.....	161	10.7 用 typedef 定义类型名.....	194
9.4.2 指针作为函数的返回值.....	163	习题.....	196
9.4.3 指向函数的指针.....	164		
9.5 多级指针.....	166		

第 11 章 文件操作	198	12.3.5 程序运行和测试	223
11.1 文件的基本概念	198	12.4 综合实训四: BASIC 程序	
11.1.1 概述	198	解释器	224
11.1.2 文件分类	198	12.4.1 问题描述	224
11.1.3 缓冲文件系统和非缓冲		12.4.2 问题分析	224
文件系统	199	12.4.3 数据结构和程序设计分析	228
11.1.4 流式文件	199	12.4.4 程序运行和测试	238
11.2 标准文件	199	12.4.5 讨论	239
11.3 文件类型指针	200	第 13 章 初涉 ACM/ICPC	242
11.4 文件的打开与关闭	200	13.1 ACM/ICPC 概述	242
11.4.1 文件的打开	200	13.2 小数近似值问题与枚举算法	243
11.4.2 文件的关闭	201	13.2.1 问题描述	243
11.5 文件的顺序读写	202	13.2.2 问题分析与求解	243
11.6 常用文件顺序读写函数	202	13.2.3 问题小结	244
11.7 文件顺序读写的应用举例	207	13.3 迷宫问题与深度优先搜索	244
11.8 文件的随机读写	208	13.3.1 问题描述	244
11.8.1 文件的定位	209	13.3.2 问题分析与求解	244
11.8.2 文件操作的出错检测	211	13.3.3 问题小结	245
11.9 非缓冲文件系统	211	13.4 经典 01 背包问题与动态规划	
习题	214	算法	245
第 12 章 综合实训	215	13.4.1 问题描述	245
12.1 综合实训一: 24 点程序	215	13.4.2 问题分析与求解	245
12.1.1 问题描述	215	13.4.3 问题小结	247
12.1.2 问题分析	215	13.5 经典部分背包问题与贪心算法	247
12.1.3 数据结构分析	216	13.5.1 问题描述	247
12.1.4 程序执行流程和设计分析	217	13.5.2 问题分析与求解	247
12.1.5 程序运行和测试	217	13.5.3 问题小结	249
12.2 综合实训二: 五子棋游戏	218	13.6 关系序列问题与图的拓扑排序	249
12.2.1 问题描述	218	13.6.1 问题描述	249
12.2.2 问题分析	218	13.6.2 问题分析与求解	249
12.2.3 数据结构分析	218	13.6.3 问题小结	252
12.2.4 程序执行流程和设计分析	219	13.7 公路建设图与最小生成树	252
12.2.5 程序运行和测试	221	13.7.1 问题描述	252
12.3 综合实训三: 通用的管理信息		13.7.2 问题分析与求解	252
系统	221	13.7.3 问题小结	254
12.3.1 问题描述	221	13.8 青蛙跳石头与图的单源最短	
12.3.2 问题分析	222	路径	254
12.3.3 数据结构分析	222		
12.3.4 程序执行流程和设计分析	223		

13.8.1 问题描述	254	14.3 结构化程序设计	268
13.8.2 问题分析与求解	254	14.3.1 自顶向下的程序设计方法	268
13.8.3 问题小结	255	14.3.2 程序的模块化	270
13.9 青蛙约会之解与模线性方程	255	14.4 健全程序的风格标准	271
13.9.1 问题描述	255	14.5 程序错误类型和调试	274
13.9.2 问题分析与求解	256	14.5.1 程序错误类型	274
13.9.3 问题小结	257	14.5.2 程序错误分析方法	275
13.10 练习参考网址	257	14.6 程序调试方法	276
第14章 程序设计风格与程序调试	258	14.7 常见错误分析	283
14.1 程序设计风格和程序设计方法	258	14.8 帮助的使用	289
14.2 C语言程序设计风格	259	习题	289
14.2.1 标识符的命名	259	附录 ASCII 码表	291
14.2.2 注释	260	参考文献	293
14.2.3 清晰简洁的表达	263		
14.2.4 书写格式	264		

第 1 章

C 语言与程序设计概述

1.1 初见 C 语言程序

中国古代数学家张邱建，在他的《算经》中提出了著名的“百钱买百鸡问题”：鸡翁一，值钱五，鸡母一，值钱三，鸡雏三，值钱一，百钱买百鸡，问翁、母、雏各几何？对于这个问题，很多读者在小学或初中的竞赛中可能都见到过，而且通常都采用不定方程的方式来求解。现在，我们演示通过计算机，用 C 语言解决该问题的方法。通过这个程序，一方面可以让读者对 C 语言有一个感性认识，另一方面可以初步领略计算机解决问题的高效和强大能力。

【例 1-1】 用 C 语言程序解决“百钱买百鸡问题”。

```
#include <stdio.h>
main()
{
    int i,j,k,count=0;
    for (i=0;i<=20;i++)
    {
        for (j=0;j<=33;j++)
        {
            for (k=0;k<=300;k++)
            {
                if (i+j+k==100&&k%3==0&&i*5+j*3+k/3==100)
                {
                    count++;
                    printf("方案 %d: 鸡翁 %d 个, 鸡母 %d 个, 鸡雏 %d 个\n",count,i,j,k);
                }
            }
        }
    }
}
```

运行程序，得到图 1-1 所示结果。

```
方案1: 鸡翁0个, 鸡母25个, 鸡雏75个
方案2: 鸡翁4个, 鸡母18个, 鸡雏78个
方案3: 鸡翁8个, 鸡母11个, 鸡雏81个
方案4: 鸡翁12个, 鸡母4个, 鸡雏84个
```

图 1-1 例题 1-1 所示运行结果

例 1-1 显示了一个完整的 C 语言程序，规模很小，功能很简单，但能解决一个实际的问题。从程序中可以看出，在该问题的求解过程中，我们采用了穷举的方法。根据题意，如果 100 元钱全部都用来买鸡翁，则最多可买鸡翁 20 只，现在我们假设可买鸡翁数量为 x ，可买的鸡母数量为 y ，可买的鸡雏数量为 z ，则 x 的范围为 $0 \sim 20$ ， y 的范围为 $0 \sim 33$ ， z 的范围为 $0 \sim 300$ 。对以上范围内所有的 x ， y ， z 的组合，如果 $x+y+z$ 的总和为 100，并且买 x ， y ， z 花费的钱总和为 100，则 x ， y ， z 就是满足条件的解。事实上，穷举法是计算机求解问题时常

用的一种方法。

例 1-1 所示程序称为 C 语言的源程序。在 C 语言源程序的描述中, 要注意以下几点:

- 1) C 语言源程序的扩展名必须为 .c 或 .cpp。
- 2) C 语言是大小写敏感的, 在 C 语言的源程序中, 大小写是有区别的。
- 3) 如果源程序中出现的符号不是出现在双引号的内部, 则均应该在英文半角状态下输入该符号, 比如分号不能写成中文分号, 而应写成英文半角分号。

例 1-1 是一个用 C 语言编写的解决实际问题的程序实例。读者可以思考一下, 我们生活中碰到的哪些问题可以用类似的方法让计算机帮助我们解决问题。

1.2 计算机与程序设计

计算机的功能非常强大, 能做非常复杂、人脑难以胜任的许多工作。然而, 当你从电子市场买回 CPU、内存、硬盘、机箱等自己组装好一台计算机后, 你会发现这台计算机什么也不能做。究其原因, 就是因为该计算机上还没有安装任何计算机程序(也就是软件)。硬件是计算机拥有强大功能的前提条件, 但是如果没有大脑(也就是计算机程序)去指挥它, 它将什么也不能做, 所以计算机程序的存在是计算机能够工作、能够按指定要求工作的前提条件。因此, 计算机程序(Program)可以简单理解为人们为解决某种问题而用计算机可以识别的代码编排的一系列加工步骤, 计算机能严格按照这些步骤去执行任务。计算机只是一台机器, 没有智慧, 只能按照既定的逻辑工作, 这种逻辑是为了实现某个目标而人为制定的, 因此我们制定的逻辑必须让计算机能够理解, 计算机才能按要求去工作。人们按照计算机能够理解的“语言”来制定这些规则的过程, 就是程序设计的过程。

1.2.1 指令与程序

计算机功能比较强大, 但是没有智能, 而且每次只能完成非常简单的任务, 复杂任务必须通过一系列的简单任务有序组合才能完成。因此, 人在向计算机发号施令的时候只能以一个简单任务接一个简单任务的方式来完成。这个简单任务称为计算机的指令。一条指令本身只能完成一个最基本的功能, 如实现一次加法运算或实现一次大小的判别, 不同的指令能完成不同的简单任务。但是通过对多条指令的有序组织, 就能完成非常复杂的工作, 这个一系列计算机指令(也可理解成人的司令)的有序组合就构成了程序, 对这些指令的组织过程就是编程的过程, 组织规则就是编程的语法规则。

[例 1-2] 假设计算机能识别的指令有以下四条:

Input X: 输入数据到存储单元 X 中。

Add X Y Z: 将 X、Y 相加并将结果存在 Z 中。

Inv X: 将 X 求反后存回 X。

Output X: 输出 X 的内容。

请编写一段由上述指令组成的虚拟程序, 实现功能: 输入 3 个数 A、B 和 C, 求 $A+B-C$ 的结果。

程序如下:

Input A;	输入第 1 个数到存储单元 A 中
Input B;	输入第 2 个数到存储单元 B 中
Input C;	输入第 3 个数到存储单元 C 中
Add A B D;	将 A、B 相加并将结果存在 D 中
Inv C	
Add C D D;	将 C、D 相加并将结果存在 D 中
Output D;	输出 D 的内容

通过例 1-2 可以看出: 通过指令的有序组合, 能完成单个指令无法完成的工作。上述程序中的

指令是假设的,事实上,不同的CPU支持的指令集也不同(由CPU硬件生产商决定提供哪些指令)。有点硬件常识的读者都知道,计算机的CPU和内存等都是集成电路的集成,其能存储和处理的对象只能是0、1组成的数字序列。因此这些指令也必须以0、1序列表示,最终程序在计算机中也是以0、1组成的指令码(用0、1序列编码表示的计算机指令)来表示的,这个序列能够被计算机CPU所识别。程序与数据一样,共同存储在存储器中。当要运行程序时,当前准备运行的指令从内存调入CPU中,由CPU处理这条指令,CPU一次处理内存中的部分指令,这就是程序的运行过程。

1.2.2 程序与程序设计

计算机程序是人们为解决某种问题用计算机可以识别的代码编排的一系列数据处理步骤,是计算机能识别的一系列指令的集合。计算机能严格按照这些步骤和指令去操作。程序设计就是针对实际问题,根据计算机的特点,编排能解决这些问题的步骤。程序是结果和目标,程序设计是过程。

1.2.3 程序设计和程序设计语言

程序设计是按指定要求,编排计算机能识别的特定指令组合的过程,而程序设计语言是为方便人进行程序设计而提供的一种手段,是人与计算机交流的语言,而且这种程序设计语言也在随着计算机技术的发展而不断地发展。

计算机能直接识别的是由“0”和“1”组成的二进制数,二进制是计算机语言的基础。一开始,人们只能降贵纡尊,用计算机能直接理解的语言去命令计算机工作,通过写出一串由“0”和“1”组成的指令序列交给计算机执行。这种语言称为机器语言。使用机器语言编写程序是一件十分痛苦的工作,特别是在程序有错需要修改时,更是如此。而且,由于每台计算机的指令系统往往各不相同,所以,在一台计算机上执行的程序,要想在另一台计算机上执行,必须重新修改程序,造成了重复工作。所以,现在已经很少有人用机器语言直接写程序。

为了减轻使用机器语言编程的痛苦,人们进行了一种有益的改进:用一些简洁的英文字母、有一定含义的符号串来替代一个特定指令的二进制串,比如,用“ADD”表示加法,“SUB”表示减法,“MOV”表示数据传递等,这样一来,人们很容易读懂并理解程序在干什么,纠错及维护都变得方便了。这种程序设计语言称为汇编语言,即第二代计算机语言。然而对于计算机而言,它只认识“0”和“1”组成的指令,并不认识这些符号,这就需要有一个专门的程序,来负责将这些符号翻译成计算机能直接识别和理解的二进制数的机器语言,完成这种工作的程序被称为汇编程序,它充当的就是一个翻译者的角色。汇编语言同样十分依赖于机器硬件,移植性不好,但效率十分高。现代的桌面计算机,性能已经非常强大,效率已经不是首要关注目标。所以,通常只有在资源受限的嵌入式环境或与硬件相关的程序设计时(如驱动程序),汇编语言才会作为一种首选的软件开发语言。

虽然机器语言发展到汇编语言已经有了很大的进步,但是由于每条指令完成的工作非常有限,因此编程过程仍然繁琐,语义表达仍然比较费力。于是,人们期望有更加方便、功能更加强大的高级编程语言的出现。这种高级语言应该接近于数学语言或人的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上通用。C语言就是一种能满足这种要求的语言,它由于既有高级语言的通用性又有底层语言的高效性而展示出了强大的生命力,几十年来一直被广泛应用。许多高校也基本上将C语言当作计算机专业和相关专业的重要必修课,作为高校学生接触的第一门编程语言。同样,计算机本身并不认识C语言程序,因此我们需要将C语言程序先翻译成汇编程序,再将汇编程序翻译成机器语言,这个过程往往由编译程序帮我们完成,而不需要我们自己来做。

为了使程序设计更加接近自然语言的表达,方便用户实现功能,包括C语言在内的所有程序