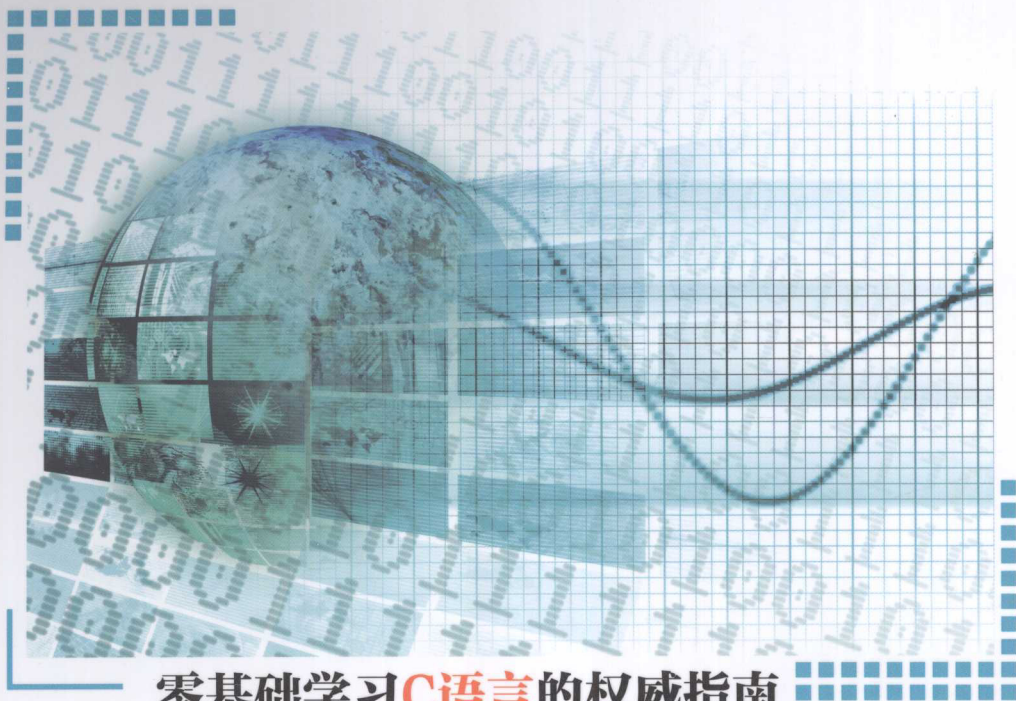




零基础学习**C语言**的权威指南

C语言

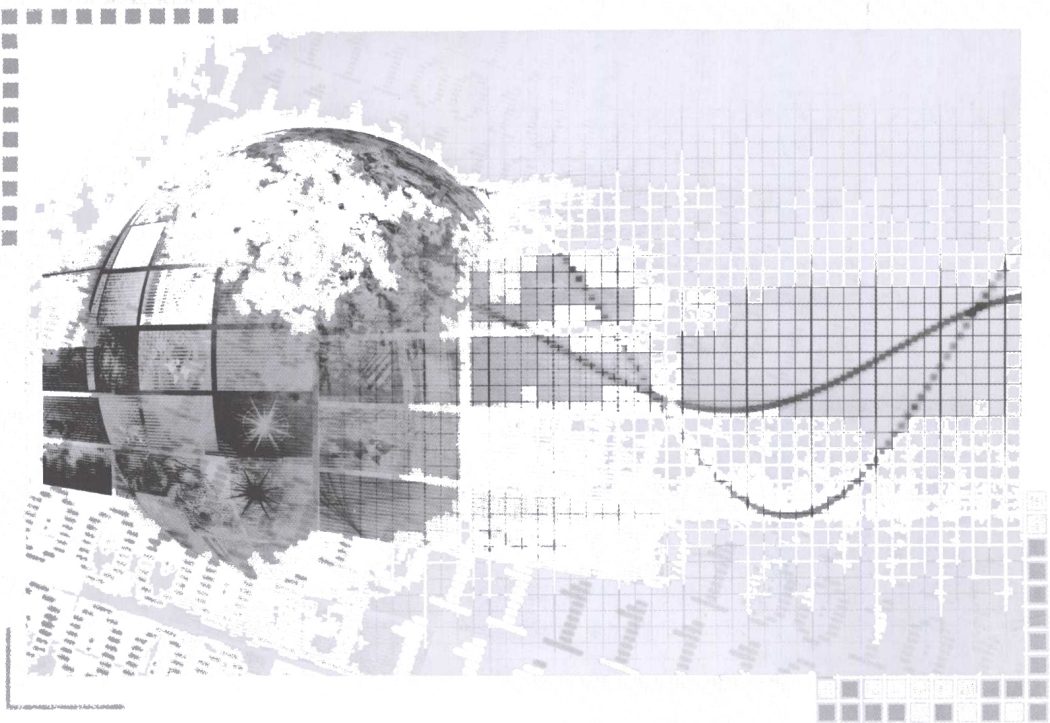
王金鹏◎著

可以这样学

- 本书从零基础讲起，以探究式的方法，用凝炼而准确的语言系统地讲述了**C语言**的各个知识点。
- 书中荟集了**C语言**的精华和作者20多年的教学经验，并且纠正了一些教材中存在的若干错误说法。
- 本书提供全套教学课件、所有（100多个）例题的**源代码**和**习题答案**。
- 提供**OnlineJudge**平台进行编程实践和测验，支持QQ、电子邮件、论坛等方式与作者交流。



清华大学出版社



语言 可以这样学

王金鹏◎著

清华大学出版社
北 京

内 容 简 介

本书从计算机基础知识讲起,继而介绍标准 C 语言的内容。除此之外,书中还包含了 C 编程必需的若干重要内容。

本书深入浅出,文字简练,将复杂的问题简单化,篇幅不大但内容全面;对各章节的重点、难点把握准确,处理得当;注重培养编程思维能力,对编程时易犯的错误,点评到位。书中对 C 语言中最重要的内容——函数、指针、数组、文件四部分的编写,会使读者的认识上升一个层次。尤其是指针部分,全面纠正一些读者对指针的若干错误认识,给出指针的明晰、准确的说法和定义。

本书作者讲授 C 语言 20 多年,有丰富的编程和教学经验,对学生的思维方式和学习状况非常了解,对 C 语言的知识体系烂熟于心。在书中,作者奉献了自己对许多问题的独到见解。书中大量的编程经验和注意事项,蕴含着作者长期的积累,凝聚着 C 语言的精华。

本书非常适合作为高等学校各专业程序设计基础、C 语言程序设计等课程的正式教材,也可作为自学教材或学习参考书,尤其适合作为 C 程序员的编程指导手册和入门教材。

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 语言可以这样学/王金鹏著. —北京:清华大学出版社,2016
ISBN 978-7-302-44570-8

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

中国版本图书馆 CIP 数据核字(2016)第 174912 号

责任编辑:白立军

封面设计:杨玉兰

责任校对:白 蕾

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:23

字 数:529 千字

版 次:2016 年 9 月第 1 版

印 次:2016 年 9 月第 1 次印刷

印 数:1~2000

定 价:49.00 元

产品编号:070254-01



C 语言是高等学校计算机及相关专业必修的专业基础课之一,是培养学生算法思维能力、动手能力的主要课程和工具,也是面向对象程序设计、数据结构等后续课程的先导课。学生对 C 语言的掌握情况在很大程度上决定着大学四年的学习情况。

鉴于 C 语言的重要地位,近些年来出现了无数的 C 语言教材,有的教材在内容和顺序上存在问题,导致长期以来国内众多教师和学生对一些知识的理解出现偏差。虽然有些教材在形式上做了不少改进,如案例教学法、启发式教学法等,但核心内容并无变化,教材中的问题依然存在。

针对目前国内教材在诸多方面的错误和缺陷,本书作者在 2015 年编写了《深入浅出新编 C 语言教程》,受到广大读者的广泛好评。在此基础上,作者对原版进行了修订,在内容上进行了调整、改进和补充。并通过“探究式”的讲解,使读者对每个知识点的理解都更加准确、透彻。

编写本书的指导思想

(1) 零基础讲起。针对刚入大学的新生及初级程序员,将 C 语言编程的一些必要的计算机知识纳入本书第 1 章。

(2) 按符合读者认知规律的自然顺序安排章节,而不是为了所谓的“系统”、“全面”不顾及知识点的自然顺序把前后相关的所有内容都堆放在一起。那样做很容易把初学者的思维搞乱,因为对于初学者,理顺众多知识点之间的关系很难。

(3) 化繁为简、化整为零。重要的知识点单独写成一章,每章内容相对独立,与其他知识点关联少,条理清楚,易于初学者掌握。

(4) 对“指针”一章的内容和“文件”一章的内容重点着墨,纠正诸多错误说法和错误代码,澄清若干似是而非的问题。

(5) 将作者多年积累的教学经验、对若干问题的独到见解、编程注意事项、典型例题和习题写到书中,奉献给广大读者。

本书在以下几方面做了较大改进。

1. 对教材内容的改进

1) 增加了以下几部分内容

要学好 C 语言,下面的知识是必要的。

(1) 计算机基础知识。绝大多数学校都把 C 语言放在大一的第一学期开设,对于没



有任何计算机基础的新生来说,C 语言的知识仿佛“天书”。因此,本书从计算机基础知识讲起。这些基础知识包括内存和内存地址的概念,二进制,不同数制之间的转换,原码、反码和补码的求法,计算机语言及语言处理过程等。

(2) 有关路径和输入输出重定向的概念。C 语言中,很多地方需要用到路径和输入输出重定向的概念,故也把这两部分内容编入本书。其中路径部分放在第 1 章,作为选讲或自学的内容;输入输出重定向部分放在附录中,供需要的读者自学。

(3) 缓冲区及键盘缓冲区的概念。学习 C 语言的输入输出,缓冲区是个绕不开的话题。如果不知道数据从键盘到缓冲区的处理过程,就很难掌握输入输出,就很难解释为什么程序会出现那些意想不到的运行结果。

(4) 函数的作用和函数设计的原则。函数是被调用的,因此函数的适用性和灵活性是衡量一个函数优劣的重要指标。多数教材只注重讲解函数定义和函数调用的格式、函数参数传递的特点,对于函数的作用和设计原则(追求通用性、可利用率等)极少谈及。本书从函数返回值的设定和参数设定两方面详细讲述函数设计的原则。

(5) 文件操作原理及相关细节问题。文件一章的内容非常重要,但又特别难懂。难懂的原因有三:一是几乎所有教材都未给出文件操作的原理,学生知其然,不知其所以然;二是有几个概念特别容易混淆,如写数据有文本和二进制两种方式,文件分为文本和二进制两类文件,文件的打开方式也分为文本方式和二进制方式。有的教材未明确指出它们之间有无关联,区别是什么,导致学生概念混乱;三是有的教材在介绍文件操作时都不详细,对一些重要内容不予讲解,导致学生一编程就出错,望文件而生畏。本书对上面所说问题均做了详细讲述并予以例证。

2) 纠正了读者认知的若干错误

(1) 纠正了关于指针的一些错误说法。例如,“指针就是地址”、“若有定义 `int a[10], *p=a;` 则 `p` 指向数组 `a`”、“若有定义 `int a[3][5], *p=a;` 则 `p` 是指向二维数组的指针”、“指向字符串的指针变量”、“数组名就是数组的首地址”、“`&a` 是 `a` 的地址”等错误说法。

(2) 对文件操作中的一些问题进行了纠正或澄清。例如,函数 `feof()` 何时返回非零值问题(多数教材所讲都是错的)、用二进制方式能否打开文本文件、用文本方式打开文件后能否以二进制方式向其中写数据等问题。

(3) 澄清了共用体变量能否初始化、共用体变量能否作为参数等问题。

2. 对各章节的内容分配及前后顺序都做了较大调整和优化

1) 对指针内容的分解

指针是 C 语言的精华,但指针又非常难学。因为 C 语言中指针的类型很多。如此多的类型本就容易混淆,一般教材又把它们全部放在一章中讲解,显得很全面、很系统。但学生要在一章中(一两周的时间)弄懂如此多且抽象难懂的内容,实在是勉为其难。实际效果表明,这样安排很不合理。也正是因为这样安排才使得指针如此难学。

本书将指针最重要的两个应用——用指针变量访问变量、用指针变量访问下标变量两部分抽出来作为单独的两章来讲解。其中,“用指针变量访问变量”一章放在函数之后、数组之前讲解,“用指针变量访问下标变量”一章放在数组之后讲解,其余内容放在“指针



综述”一章中介绍。如此分解可化繁为简,具有重点突出、针对性强、易于接受等优点,也彰显了指针的这两个重要用途。另外,如此设计也可把对指针的学习从短短一两周的时间扩展到前后约一个月的时间。较长时间的消化,有利于学生更好地理解指针、掌握指针。

2) 各章节顺序的调整

(1) 数组和指针的顺序问题。一般教材都是先讲数组,再讲指针。带来的问题就是,无法对数组名进行解释,于是产生了“数组名是个地址”的错误说法。实际上数组名在多数情况下都是一个指针。在不介绍指针的情况下,很难把数组一章的内容讲清、讲透。

先讲数组再讲指针,也无法讲明在数组名作为参数的情况下被调函数形参 `int x[]` 中的 `x` 是个指针变量,只好把它说成是“与实参数组具有相同地址的数组”,不仅难以令人信服,而且对很多现象(比如为什么 `x` 可以进行自增运算)也无法解释。

(2) 数组和函数的顺序问题。一般教材都是把函数放在数组之后讲解,原因是便于把数组名作为参数放在函数一章中。看起来似乎归类得当,岂不知这样一来就掩盖了函数一章的重点。函数一章,最应该教给学生的是如何把函数设计得当,便于其他函数调用,只应突出这一重点。如果把数组问题也放在函数一章中,就会喧宾夺主,因为数组名作为参数本身也是非常重要的一个知识点。

综上所述,最合适的顺序安排应是函数、指针(1)、数组、指针(2)、指针综述。

本书的使用建议

建议理论授课学时数:56~64,建议实验学时数:32。

第1章计算机基础知识,若授课对象不是大一第一学期新生,已有基础,可以不讲,或只讲需要的部分。其中带星号的内容(路径及其表示)为选讲内容。

本书适用对象:高等院校本、专科所有开设“程序设计基础”或“C语言程序设计”课的学生,或自学C语言的读者,以及初级向高级进阶的程序员。

其他说明

1. 本书所用编译器

本书讲解时兼顾 Visual C++ 6.0(简称 VC)和 Turbo C 2.0,但程序主要是针对 VC 编写的,所有源程序都在 VC 中调试、运行过,例题中的运行结果都是在 VC 中得到的。

2. 例题和源代码

本书配套资料(可以 www.tup.com.cn 下载)中含有全部 103 个例题的源代码,例题编号与源程序的编号一一对应。例如,例 2.1 的源代码对应资料中的源文件 `s2_1.c`,若该例题有 3 种解法,则对应的源文件分别是 `s2_1_1.c`、`s2_1_2.c` 和 `s2_1_3.c`。

本书的编写获山东省高等学校教学改革项目资助。原达教授、谢青松教授对本书的编写给予了热情帮助和大力支持,在此向两位深表谢意。此外,本书的编写参考了大量的书籍和文献资料,谨向这些书籍和文献资料的作者表示感谢。本书在编写出版过程中也



得到了清华大学出版社的大力支持和帮助,在此一并表示感谢。

由于时间仓促和作者水平所限,书中难免存在疏漏和欠妥之处,请各位专家、读者不吝指正。

作 者

2016 年 6 月 25 日



第 1 章	计算机基础知识	1
1.1	计算机的硬件组成	1
1.1.1	运算器	1
1.1.2	控制器	1
1.1.3	存储器	1
1.1.4	输入设备	3
1.1.5	输出设备	3
1.2	数制及数制间的转换	3
1.2.1	二进制	3
1.2.2	八进制	5
1.2.3	十六进制	5
1.3	原码、反码和补码	6
1.3.1	原码	6
1.3.2	反码	6
1.3.3	补码	6
1.4	路径及其表示*	6
1.4.1	路径的概念	6
1.4.2	当前盘和当前目录	7
1.4.3	绝对路径和相对路径	7
1.5	计算机语言	8
1.5.1	机器语言	8
1.5.2	汇编语言	9
1.5.3	高级语言	10
1.6	算法	11
1.6.1	算法的概念	11
1.6.2	算法的特性	12
1.6.3	算法的表示	12
1.6.4	程序的 3 种基本结构	13
习题 1		14



第 2 章	C 程序和 C 编译器简介	16
2.1	C 语言及 C 标准简介	16
2.1.1	C 语言的出现	16
2.1.2	C 语言的特点	16
2.1.3	C 标准	17
2.2	简单的 C 程序	18
2.3	C 程序的构成	22
2.4	C 编译器及操作简介	24
2.4.1	Turbo C 2.0 编程环境及常用操作简介	24
2.4.2	Visual C++ 6.0 编程环境及常用操作简介	28
	习题 2	33
第 3 章	C 编程基础知识	35
3.1	常量和变量	35
3.1.1	常量	35
3.1.2	变量	35
3.2	基本数据类型	40
3.2.1	整型数据	40
3.2.2	实型数据	42
3.2.3	字符型数据	45
3.2.4	字符串	47
3.3	符号常量和常变量	48
3.3.1	符号常量	48
3.3.2	常变量	48
3.4	运算符和表达式	49
3.4.1	算术运算符	49
3.4.2	赋值运算符和赋值表达式	50
3.4.3	自增自减运算符	51
3.4.4	逗号运算符和逗号表达式	53
3.4.5	类型转换运算符	54
3.5	数据的类型转换	55
	习题 3	55
第 4 章	顺序结构程序设计	59
4.1	赋值语句	59
4.1.1	赋值语句及其执行过程	59
4.1.2	赋值的几种数据处理方式	59



4.2	输入输出函数	63
4.2.1	缓冲区的概念及作用	63
4.2.2	getchar()和 putchar()	64
4.2.3	printf()和 scanf()	65
4.3	顺序结构程序设计举例	70
习题 4		72
第 5 章	选择结构程序设计	76
5.1	关系运算符和关系表达式	76
5.1.1	关系运算符	76
5.1.2	关系表达式	76
5.2	逻辑运算符和逻辑表达式	77
5.2.1	逻辑运算符	77
5.2.2	逻辑表达式	78
5.3	if 语句	79
5.3.1	if 语句的格式	79
5.3.2	if 语句的使用说明	80
5.3.3	嵌套的 if 语句	83
5.3.4	if 语句应用举例	85
5.3.5	if 语句编程的常见问题	87
5.4	条件运算符和条件表达式	92
5.5	switch 语句	93
5.5.1	switch 语句的格式及执行过程	94
5.5.2	switch 语句应用举例	96
5.5.3	switch 语句编程的常见错误	97
习题 5		98
第 6 章	循环结构程序设计	104
6.1	循环及其实现思想	104
6.2	循环语句	105
6.2.1	while 循环	105
6.2.2	do-while 循环	107
6.2.3	for 循环	108
6.2.4	3 种循环的比较	110
6.3	循环的控制	111
6.3.1	计数器控制循环和其他条件控制循环	111
6.3.2	break 和 continue	112
6.3.3	循环结束后循环变量的值与终值的比较	115



6.4	多重循环	116
6.5	循环编程举例	117
	习题 6	126
第 7 章	函数	132
7.1	函数的作用	132
7.2	函数的定义	134
7.2.1	函数定义的格式	134
7.2.2	函数的返回值	134
7.2.3	函数参数的设置	138
7.3	函数的调用	140
7.3.1	函数调用前的声明	140
7.3.2	函数调用的方式	142
7.4	函数的参数传递	143
7.4.1	形参与实参	143
7.4.2	参数的传递	143
7.4.3	参数传递的单向性	144
7.5	函数的嵌套调用	145
7.6	递归函数	146
7.6.1	递归的条件	146
7.6.2	递归函数的执行过程	147
7.6.3	递归与迭代	149
7.7	函数编程举例	150
7.8	内部函数和外部函数	153
	习题 7	154
第 8 章	变量的作用域和存储类别	158
8.1	变量的作用域	158
8.1.1	局部变量	158
8.1.2	全局变量	158
8.2	同名变量的辨析	160
8.3	变量的存储类别和生存期	162
8.3.1	内存的存储区域	162
8.3.2	动态变量	162
8.3.3	静态变量	163
8.4	变量的作用域和生存期	164
	习题 8	165



第 9 章	编译预处理	168
9.1	宏定义	168
9.1.1	无参宏定义	168
9.1.2	有参宏定义	169
9.1.3	嵌套的宏定义	170
9.2	文件包含	170
9.2.1	文件包含的格式	170
9.2.2	文件包含的作用	171
9.2.3	文件包含两种格式的区别	171
9.3	条件编译	172
9.3.1	条件编译的格式	172
9.3.2	条件编译应用举例	174
	习题 9	175
第 10 章	用指针变量访问变量	178
10.1	指针和指针变量	178
10.1.1	指针和指针变量的概念	178
10.1.2	直接寻址和间接寻址	179
10.1.3	指针变量的值、地址及类型	181
10.2	通过指针变量访问变量	181
10.2.1	指针变量的定义	181
10.2.2	指针变量的赋值	182
10.2.3	通过指针变量间接访问一个变量	183
10.3	指针变量在函数传递中的作用	183
	习题 10	188
第 11 章	数组	191
11.1	一维数组	191
11.1.1	一维数组的定义	191
11.1.2	一维数组的元素构成及一维数组的存储结构	192
11.1.3	数组名的指针类型	192
11.1.4	数组元素的表示方法	193
11.1.5	一维数组的引用	194
11.1.6	一维数组的初始化	196
11.1.7	一维数组应用举例	196
11.2	二维数组	201
11.2.1	二维数组的定义	201



11.2.2	二维数组的元素构成及二维数组的存储结构	202
11.2.3	二维数组名的指针类型	202
11.2.4	二维数组中下标变量的表示方法	203
11.2.5	二维数组的引用	204
11.2.6	二维数组的初始化	205
11.2.7	二维数组应用举例	205
11.3	字符数组和字符串处理函数	206
11.3.1	字符数组	206
11.3.2	字符串处理函数	207
11.3.3	字符数组应用举例	211
习题 11		213
第 12 章	用指针变量访问下标变量	218
12.1	用指针变量访问下标变量的方法	218
12.1.1	知识回顾	218
12.1.2	用指针变量访问一维数组中的下标变量	218
12.1.3	用指针变量访问二维数组中的下标变量	220
12.2	用指针变量访问下标变量的适用场合	221
习题 12		224
第 13 章	指针综述	228
13.1	指针类型简介	228
13.2	指向变量的指针	229
13.2.1	指向变量的指针常量	229
13.2.2	指向变量的指针变量	229
13.3	指向数组的指针	230
13.3.1	指向一维数组的指针常量	230
13.3.2	指向一维数组的指针变量	231
13.3.3	指向一维数组的指针变量的适用场合	232
13.4	指针与字符串	233
13.4.1	字符串的表示方式	233
13.4.2	用指针变量处理字符串	235
13.5	指针与函数	237
13.5.1	函数的入口地址	237
13.5.2	指向函数的指针变量	237
13.5.3	指向函数的指针变量的作用	238
13.5.4	指针函数	239
13.6	指针数组	240



13.6.1	指针数组的定义	240
13.6.2	指针数组的引用	241
13.6.3	指针数组应用举例	241
13.7	指向指针变量的指针	241
13.7.1	指向指针变量的不可变指针	241
13.7.2	指向指针变量的指针变量	242
13.7.3	应用举例	242
13.8	带参数的 main() 函数	245
13.8.1	C 语言对 main() 函数参数的规定	245
13.8.2	带参数 main() 函数的作用	245
13.8.3	带参数的 main() 函数的执行过程	246
13.8.4	程序举例	246
13.9	动态内存分配	247
13.9.1	动态内存分配函数	247
13.9.2	动态内存分配举例	248
习题 13		249

第 14 章 数据类型的自定义 254

14.1	结构体的定义和结构体变量的定义	254
14.1.1	结构体的概念和结构体的定义	254
14.1.2	结构体变量的定义和空间分配	256
14.1.3	结构体变量的初始化	258
14.1.4	结构体数组的定义和初始化	259
14.2	结构体变量的引用	259
14.2.1	结构体变量的引用方法	259
14.2.2	结构体变量引用举例	260
14.3	用指针变量操作结构体变量	261
14.3.1	为什么要通过指针变量访问结构体变量	261
14.3.2	应用举例	262
14.4	链表及链表操作简介	263
14.4.1	链表的概念	263
14.4.2	使用链表的优点	264
14.4.3	链表操作简介	264
14.5	共用体	269
14.5.1	共用体的概念	269
14.5.2	共用体的作用	270
14.5.3	共用体及共用体变量的定义	271
14.5.4	共用体变量(数组)的初始化	272



14.5.5	共用体变量的引用	272
14.6	枚举类型	273
14.6.1	枚举类型的定义	273
14.6.2	枚举变量的定义	274
14.6.3	枚举变量的使用	274
14.6.4	枚举应用举例	274
14.7	用 typedef 定义类型别名	275
习题 14		277
第 15 章	位运算	280
15.1	C 语言中的位运算符	280
15.2	位运算及应用	281
15.2.1	按位与	281
15.2.2	按位或	282
15.2.3	异或	283
15.2.4	取反	284
15.2.5	左移	284
15.2.6	右移	285
习题 15		286
第 16 章	文件	288
16.1	文件及相关的概念	288
16.1.1	文件的范畴	288
16.1.2	文件中存储数据的两种方式	288
16.1.3	文件的种类	289
16.1.4	文件操作函数及缓冲区的概念	290
16.2	文件读写的原理	291
16.3	文件的读写位置指针和文件结束标志	292
16.3.1	读写位置指针	292
16.3.2	文件结束标志	292
16.4	文件的打开和关闭	293
16.4.1	文件的打开	293
16.4.2	文件的关闭	299
16.5	文件的读写	300
16.5.1	fgetc() 和 fputc()	301
16.5.2	fread() 和 fwrite()	303
16.5.3	fgets() 和 fputs()	307
16.5.4	fscanf() 和 fprintf()	308



16.6	读写位置指针的移动和定位·····	310
16.6.1	移动读写位置指针的函数·····	310
16.6.2	两个与读写位置指针有关的函数·····	311
16.7	文件的出错检测·····	312
16.8	文件操作举例·····	313
习题 16	·····	317
第 17 章	综合应用举例 ·····	320
17.1	简单的计算器(一)·····	320
17.2	简单的计算器(二)·····	322
17.3	数据库文件的读取·····	325
附录 A	C 语言规约 ·····	335
附录 B	输入输出重定向 ·····	337
附录 C	C 语言的关键字 ·····	340
附录 D	常用字符与 ASCII 码对照表 ·····	341
附录 E	运算符的优先级和结合性 ·····	342
附录 F	常用库函数 ·····	344
参考文献	·····	349

第 1 章



计算机基础知识

本章内容提要

- (1) 计算机的硬件组成。
- (2) 数制及数制间的转换。
- (3) 原码、反码及补码。
- (4) 路径及其表示。
- (5) 计算机语言及语言处理过程。

学习 C 语言,需要先掌握一些计算机的基础知识。本章介绍的内容就是 C 编程将要用到的一些基础知识和基本概念。本章的重点是内存及其特性、不同进制之间的转换、补码的求法。

1.1 计算机的硬件组成

计算机的硬件由 CPU(Central Processing Unit)、存储器、输入设备和输出设备组成。

CPU 是计算机的大脑,负责从内存获取指令并执行之。CPU 通常由控制单元(控制器)和算术逻辑单元(运算器)组成。

1.1.1 运算器

负责进行算术运算和逻辑运算。组成运算器的部件有寄存器、控制电路和执行部件等。

1.1.2 控制器

负责从内存中取回指令、分析指令并控制其他部件共同完成指令的执行。控制器相当于人的大脑或乐队的指挥。

1.1.3 存储器

存储器用来存储程序(程序由若干指令组成)和数据。