



励志照亮人生 编程改变命运

零基础学 C语言

康莉 李宽 等编著

- ◎ 循序渐进，全面掌握：从入门到提高，从语法、语义到数据结构，一点一滴吃透C语言
- ◎ 实例丰富，迅速上手：提供133个实例和1个综合案例讲解，实例可轻松移植，易于掌握
- ◎ 示例简洁，紧扣主题：每个示例程序短小精炼，可以很好地展示所讲知识要点
- ◎ 要点突出，贴心提示：对不好理解的技术点或容易出错的地方给出特别提示，加深读者理解
- ◎ 编程经验，心得体会：给出编者多年开发经验的精心总结，让读者少走弯路

 机械工业出版社
China Machine Press

编程
零基础学


CD-ROM

零基础学 C语言



康莉 李宽 等编著

目录

第1章 C语言概述

1.1 C语言的发展

1.2 C语言的特点

1.3 C语言的应用

1.4 C语言的学习方法

1.5 本章小结

习题1

习题2

习题3

习题4

习题5

习题6

习题7

习题8

习题9

习题10

零基础学
编程



机械工业出版社
China Machine Press

C语言是编程方式灵活多样、功能强大、应用广泛的一种程序设计语言。从程序设计语言的发展历程来看,尽管后来出现了以C++、Java和C#等为代表的新语言,但C语言的基础地位不可撼动。

C语言的核心是结构化的编程思想,本书一共分为四篇,循序渐进地讲述了C语言的语法规则和编程思想,从基本概念到具体实践、从入门知识到高阶主题、从语法语义到数据结构 and 算法都进行了详细的阐述,并通过实例对具体知识点进行了详细的讲解。

本书的特点是内容全面、翔实,通俗易懂,循序渐进地介绍了C语言各方面的知识,重点突出。本书含有大量实例,代码短小精炼,紧扣所讲要点的本质,以加深读者的印象,同时结合笔者多年使用C语言的经验,阐述了很多代码编写技巧,读者可将代码复制到自己的机器上进行实验,自行实践和演练。

本书适合C语言的初级、中级读者阅读,并可作为开发人员的参考手册。

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

零基础学C语言/康莉,李宽等编著. —北京:机械工业出版社,2009.4
(零基础学编程)

ISBN 978-7-111-26603-7

I. 零… II. ①康… ②李… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2009)第040257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:李华君

北京瑞德印刷有限公司印刷

2009年4月第1版第1次印刷

203mm×260mm·25印张

标准书号:ISBN 978-7-111-26603-7

ISBN 978-7-89451-036-5(光盘)

定价:48.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010) 68326294

写给未来的程序员

很多初学者对编程心生恐惧，不知道从何学起。尤其是看了别人的经验，说要先学操作系统原理、数据结构、算法等，一大堆概念涌过来，让人不知所措。其实，编程没有那么复杂，跟我们来吧，“零基础学编程”系列正是为不会编程的你准备的！

学习编程需要注意的几点

- 不要死记语法：很多初学者试图把各种语法背下来，其实这是极其错误的，程序开发的语法、规范特别多，不可能都记得下来，你只要知道有这么一个功能就可以了，需要的时候翻阅书籍，或者查找帮助文件，这样省时省力。
- 多动手，多练习：只知道死啃书本的人，是不会成为开发高手的，只有多上机编写程序，才能在实践中提高对编程的认识。
- 遇到问题，首先尝试自己解决：自己先用一天的时间，尽力解决，实在不行再去找人帮助，千万不要遇到问题立刻找人帮忙，这样永远也不会提高。
- 多用Google、Baidu：网络是一个大知识库，是最好的老师，你遇到的问题，别人也遇到过，多去搜索一下吧。
- 多阅读别人的源代码：要看懂别人的设计思想，不断地融为己用。

丛书书目

- 《零基础学Ajax》
- 《零基础学JavaScript》
- 《零基础学PHP》
- 《零基础学Java》
- 《零基础学Java Web开发》
- 《零基础学Visual C++》
- 《零基础学Visual Basic》
- 《零基础学Visual Basic+SQL Server》
- 《零基础学Visual C# 2005》
- 《零基础学ASP.NET 3.5》
- 《零基础学SQL Server 2005》
- 《零基础学Dreamweaver CS3+ASP》
- 《零基础学C语言》

- 《零基础学C# 3.0》
- 《零基础学ASP.NET 3.5》
- 《零基础学HTML+CSS》
- 《零基础学C++》
- 《零基础学PHP+MySQL》
- 《零基础学Excel VBA》
- 《零基础学Struts》

丛书特点

本丛书考虑了初学者的认知特点，在讲解的过程中遵循如下原则：

- 从基本的环境配置讲起，告诉你如何搭建开发环境。
- 细致讲解第一个例子，让你在第一个例子中学会程序开发的基本步骤。
- 强调动手实践，对每个知识点都有相应的代码段，让你在操作中领会编程的真谛。
- 注释极为详尽，方便读者阅读代码，理解其中的含义。

完善的售后服务

我们提供了论坛：<http://www.rzchina.net>，读者可以在上面提问交流，另外我们还会在论坛上定期发布一些小的教程、视频动画，以提高读者的学习效率。

最后送给各位读者一句话：“天下事有难易乎？为之，则难者亦易矣；不为，则易者亦难矣。”还等什么，现在就动手学习吧！

丛书目录

- 《零基础学Ajax》
- 《零基础学JavaScript》
- 《零基础学PHP》
- 《零基础学Java》
- 《零基础学Java Web开发》
- 《零基础学Visual C++》
- 《零基础学Visual Basic》
- 《零基础学Visual Basic+SQL Server》
- 《零基础学Visual CW 2002》
- 《零基础学ASP.NET 3.5》
- 《零基础学SQL Server 2002》
- 《零基础学Dreamweaver CS3+ASP》
- 《零基础学C语言》

前 言

C语言自1972年诞生于贝尔实验室以来，以其灵活和实用的特点得到了广大用户的喜爱，迅速发展成一种应用广泛的高级语言。从网站后台到底层操作系统，从多媒体应用到大型网络游戏，均可使用C语言来开发。在工业领域，C语言也是首选的系统语言。各种操作系统，如UNIX、Linux和Windows等的内核都是采用C语言和汇编语言编写的。

创新推动着软件开发的不断进步，在C语言之后，各种新的语言相继诞生，如C++、Java、C#等，但C语言的基础地位不可撼动，因为C比较贴近操作系统，尤其在Linux操作系统下，C开发一直都是主流。

本书特点

笔者精心编写了本书，以通俗的语言深入浅出地带领读者全面学习C语言。清晰的结构、丰富的图例、简单易懂却又一针见血点破本质的示例代码使得读者能轻松迈进C语言的大门。

本书从最简单的“Hello World”程序写起，逐步深化细化，系统全面地介绍了C语言各个方面的知识。书中每个知识和技术要点都给予了翔实的例程及代码分析，代码力求短小精悍，直击要点。在知识点讲解的同时辅以笔者多年的C语言编程经验，以加深读者的理解。

本书的特点主要体现在以下几个方面。

- 本书的内容安排采用循序渐进的方式，每章主题鲜明、要点突出，适合初级、中级读者逐步掌握C语言的语法规则和编程思想。
- 范例丰富，关键知识点都辅以范例帮助读者理解。范例程序简洁，不是简单的代码罗列，而是采用短小精炼的代码紧扣所讲的技术细节，并配以详细的代码解释和说明，使读者印象鲜明，理解透彻。
- 范例可移植性强，与编译环境和平台无关，读者可轻易将代码复制到自己的机器上进行实验，自行实践和演练，直观体会所学要点，感受C语言的无限魅力。本书中的所有例子、源代码都附在随书光盘中，方便读者使用。
- 结构清晰，内容全面，兼顾了C语言的几乎所有特性。
- 大量使用了图例、表格等，帮助读者对知识点建立直观印象。
- 结合笔者多年的C语言编程和系统开发经验，本书特别标注出易出错的技术点或初学者易误解的细节，读者在学习中可少走弯路，加快学习进度。
- 本书介绍了数据结构和算法的基础知识，阐述了结构化程序设计的思想，探讨了高质量编程的内容，为读者以后进一步深入学习软件开发打下基础。
- 注重使读者对技术本质的理解，对诸如“编译器如何为程序实体分配内存”、“函数调用细节”等做了很多独创性的介绍。

本书内容

本书分为四篇，共22章。第一篇从C语言的基础知识讲起，使读者对C语言的语法和编程机制有个初步的了解。第二篇讲述如何将这些基本知识组织起来构成完整的C语言程序。第三篇是进阶内容，讨论了一些深层次的技术细节和理解困难、易出错的要点。第四篇给出C程序设计的综合实例。

第一篇C语言基础（第1章~第9章）。

本篇讲述了C语言的基础知识，包括C语言介绍、C语言程序开发步骤、不同的开发环境、C语言程序的组成、变量及数据类型、输入与输出、运算符和表达式、语句、分支、循环等。读者通过本篇可对C语言程序有个初步而全面的认识，了解C语言的由来及其强大的功用，熟悉C语言程序的结构，知道如何声明变量，如何组织语句。学完本篇，读者便可自行书写简单的C语言程序，这9章的知识是进一步学习的基础。

第二篇一窥C语言门庭（第10章~第15章）。

C语言博大精深，掌握了第一篇中的基础知识可以说只到了大门口。本篇从C语言的核心——函数讲起，介绍了数组、指针、字符串和结构体相关的内容。学完本篇才算迈进了C语言的大门。

第三篇C语言进阶主题（第16章~第21章）。

第二篇中从较为独立的角度讲述了函数、数组和指针和结构体的知识，在实际应用中，这些要素彼此交叉，应用组合千变万化，这也是C语言灵活性的具体体现。本篇安排两章分别介绍了指针和函数的技术细节，对初学者来说，理解起来可能略有难度，但这是通往高层次C语言学习的必经之路。此外，本篇还介绍了文件处理、编译及预处理、变量的生存期、作用域、可见域及数据结构方面的内容，本篇将使读者对C语言有更深入的理解。

第四篇讲述了几款游戏的制作过程：黑白棋、五子棋、扫雷、速算24。

本书由浅入深，由理论到实践，尤其适合初级、中级读者学习和完善自己的知识结构。

本书适合的读者

- C语言的初级、中级读者。
- 了解C语言，但所学不全面的人员。
- 高等院校理科学习C语言课程的学生。
- 使用C语言进行毕业设计的学生。
- 使用C语言进行项目开发的人员。
- 熟悉其他语言以此书为参考的开发人员。

本书的作者

本书由康莉、李宽主笔编写，其他参与编写和资料整理的人员有高会东、王建超、邓薇、黄丽莉、吝晓宁、汪洋、白广元、蔡念光、陈辉、冯彬、刘长江、刘明、沙金、张士强、张洪福、多召英、贾旭、江宽、陈科、方成林、班晓娟、方中纯、刘兰军、郑雪峰等。

编者

目 录

写给未来的程序员
前言

第一篇 C语言基础

第1章 踏上征程前的思想动员	1
1.1 为什么选择C语言	1
1.2 如何学好C语言	2
1.3 语言概述	3
1.3.1 什么是语言	3
1.3.2 什么是机器语言	3
1.3.3 什么是汇编语言	4
1.3.4 面向过程的语言	5
1.3.5 什么是C语言	5
1.4 程序开发周期	5
1.4.1 编辑C源代码	6
1.4.2 编译C源代码	6
1.4.3 连接目标文件	7
1.4.4 编译连接过程示例	7
1.4.5 运行程序	8
1.5 VC++、C++、C和TC——认识开发环境和语言的区别	8
1.6 小结	11
第2章 跟我写Hello World	12
2.1 自己动手创建程序	12
2.2 认识LCC-Win32开发环境	12
2.2.1 为什么选择LCC-Win32	13
2.2.2 启动LCC-Win32	13
2.2.3 新建Hello World工程	13
2.2.4 定义新工程	14
2.2.5 添加源代码到工程	15
2.2.6 编译器设置	15
2.2.7 连接器设置	15

2.2.8	调试器设置	16
2.2.9	开始编辑代码	16
2.3	编译运行——欣赏您的杰作	17
2.4	查看代码如何一步一步被执行——调试排错 (debug)	18
2.4.1	debug一词的由来	18
2.4.2	设置断点	19
2.4.3	调试运行	19
2.5	小结	20
第3章	分解Hello World——最简单C程序的组成	21
3.1	C程序概貌	21
3.2	程序员的解释——注释	21
3.3	从哪里开始，到哪里结束——main函数	23
3.4	如何在屏幕上显示文字——函数调用	24
3.5	编译器如何认识printf函数——#include预处理器指示符	25
3.5.1	函数声明及其作用	25
3.5.2	试验寻找#include的作用	26
3.5.3	#include的作用	27
3.6	计算1+1——有些小作为的程序	27
3.7	数据从哪里来，又到哪里去——变量	28
3.7.1	变量在内存中的表现形式	28
3.7.2	编译器使用变量符号表	29
3.7.3	变量及其使用	30
3.8	自己设计C函数	31
3.8.1	在main函数中计算3个整数的平均数	31
3.8.2	在main函数中分3次计算3个整数的平均数	32
3.8.3	自编函数实现计算3个整数的平均数	32
3.8.4	如何自编写函数	33
3.8.5	试验观察总结函数声明和函数定义的意义	34
3.9	语句构成程序	35
3.10	向好程序员看齐——代码风格	36
3.11	小结	36
第4章	如何存储和获取数据——变量及数据类型	37
4.1	二进制、八进制、十六进制	37
4.2	计算机中如何表示数据——字节和位	38
4.3	内存是存储数据的“房间”——认识计算机内存	39
4.4	数据类型	40
4.4.1	整数类型	41
4.4.2	整数的正负——有符号和无符号	42
4.4.3	实数类型	44
4.4.4	字符类型	45

4.4.5 数据类型总结	45
4.5 常量	46
4.5.1 直接常量(字面常量)和符号常量	46
4.5.2 符号常量的优点	48
4.5.3 字面常量的书写格式	48
4.5.4 转义字符	50
4.5.5 字符串	51
4.6 变量	52
4.6.1 变量的声明	52
4.6.2 变量声明的意义	53
4.6.3 标识符和关键字	54
4.6.4 变量在内存中占据的空间和变量的值	56
4.6.5 变量赋初值	59
4.7 几个与变量相关的经典算法	60
4.7.1 累加和累乘	60
4.7.2 交换两个变量的值	62
4.8 小结	63
第5章 用屏幕和键盘交互——简单的输出和输入	64
5.1 输入—处理—输出：这就是程序	64
5.2 向屏幕输出——printf函数详解	64
5.2.1 printf函数的一般形式	65
5.2.2 格式字符串的一般形式	67
5.2.3 %d——十进制形式输出带符号整数	67
5.2.4 %最小宽度d格式符	68
5.2.5 %ld格式符——以长整型方式输出	68
5.2.6 %o格式符——以八进制形式输出整数	69
5.2.7 %x格式符——以十六进制的形式输出整数	69
5.2.8 %u格式符——以十进制形式输出unsigned型数据	70
5.2.9 %c格式符	70
5.2.10 %s格式符——输出字符串	71
5.2.11 %f格式符——输出实型数据	74
5.2.12 %e格式符——以指数形式输出实数	75
5.2.13 %g格式符——自动选择%f或者%e形式输出	76
5.2.14 printf函数小结	76
5.3 字符输出putchar和字符串输出puts	76
5.3.1 putchar函数(字符输出函数)	77
5.3.2 puts函数(字符串输出函数)	78
5.4 接收键盘输入——scanf函数详解	79
5.4.1 一般形式	80
5.4.2 scanf函数的工作原理	80

5.4.3	多数据输入分隔规则	82
5.4.4	格式字符	83
5.4.5	scanf使用方法说明	84
5.4.6	使用scanf函数的注意事项	87
5.5	字符输入函数getchar	89
5.6	输入输出程序举例	89
5.7	小结	92
第6章	程序的基本构成——运算符和表达式	93
6.1	认识C中的运算符和表达式	93
6.1.1	认识运算符和表达式	93
6.1.2	操作数、运算符和表达式	94
6.1.3	C运算符简介	94
6.2	算术运算符和算术表达式	95
6.2.1	基本的算术运算符	95
6.2.2	++自增、--自减运算符	96
6.2.3	算术表达式和运算符的优先级及结合性	99
6.3	逗号运算符和逗号表达式	100
6.3.1	逗号表达式的一般形式	100
6.3.2	逗号表达式的优先级和结合性	100
6.4	关系运算符和关系表达式	102
6.4.1	关系运算符的种类	102
6.4.2	关系表达式的一般形式	102
6.5	逻辑运算符和逻辑表达式	103
6.5.1	逻辑运算符	103
6.5.2	逻辑真值表	104
6.6	赋值运算符和赋值表达式	105
6.6.1	赋值表达式	105
6.6.2	复合赋值运算符	105
6.6.3	类型转换	106
6.6.4	左值与程序实体	106
6.7	强制类型转换和自动类型转换	107
6.7.1	强制类型转换	107
6.7.2	自动类型转换	108
6.7.3	函数调用时的转换	109
6.8	运算符的优先级	109
6.8.1	优先级、结合性汇总	110
6.8.2	短路表达式	110
6.8.3	sizeof运算符与sizeof表达式	111
6.9	小结	111

第7章 程序的最小独立单元——语句	112
7.1 5种语句类型	112
7.1.1 表达式语句	112
7.1.2 函数调用语句	113
7.1.3 空语句	113
7.1.4 复合语句(块语句)	113
7.1.5 流程控制语句	114
7.2 结构化程序设计	114
7.2.1 什么是算法	114
7.2.2 算法的表示	114
7.2.3 算法的伪代码表示	115
7.2.4 算法的流程图表示	115
7.2.5 3种控制结构	115
7.2.6 算法示例	116
7.3 小结	117
第8章 条件判断——分支结构	118
8.1 简单的条件判断——if语句	118
8.1.1 判断表达式	118
8.1.2 花括号和if结构体	119
8.2 两条岔路的选择——if else结构	120
8.2.1 关键在else	120
8.2.2 if else结构体	121
8.3 多分支语句和分支语句嵌套	122
8.3.1 多分支	122
8.3.2 多分支if结构	124
8.3.3 分支语句嵌套	125
8.4 开关语句——switch结构	126
8.4.1 一般形式	126
8.4.2 为什么叫开关语句	128
8.4.3 default语句	130
8.4.4 if结构和switch结构之比较	130
8.5 小结	130
第9章 一遍又一遍——循环结构	131
9.1 构造循环	131
9.1.1 循环的条件	131
9.1.2 当型循环和直到型循环	132
9.2 简洁循环语句——while和do while结构	132
9.2.1 语法规则	133
9.2.2 代码块	133
9.3 最常用的循环语句——for结构	134

9.3.1 基本形式	134
9.3.2 表达式省略	136
9.3.3 循环终止和步长	137
9.4 循环嵌套	137
9.4.1 嵌套示例	137
9.4.2 嵌套的效率	138
9.5 与循环密切相关的流程转向控制语句	139
9.5.1 跳出循环——break	139
9.5.2 重来一次——continue	140
9.5.3 随心所欲——goto	140
9.6 小结	141

第二篇 一窥C语言门庭

第10章 写程序就是写函数——函数入门	143
10.1 什么是函数——根据输入进行处理返回输出	143
10.1.1 分割	143
10.1.2 库函数和自定义函数	144
10.2 自定义函数	144
10.2.1 定义的语法	144
10.2.2 函数定义范例	145
10.2.3 不要重复定义	146
10.3 函数调用与返回	146
10.3.1 形参和实参	146
10.3.2 传址调用	147
10.3.3 函数返回	149
10.4 告诉编译器有这么一个函数——函数原型声明	149
10.4.1 函数声明的语法	150
10.4.2 声明不同于定义	151
10.4.3 标准库函数的声明	152
10.5 面向过程的程序结构	152
10.5.1 模块化	152
10.5.2 函数的调用过程——模块的配合	152
10.5.3 一个入口一个出口	153
10.5.4 自顶向下，逐步求精	153
10.5.5 封装和可重用	153
10.5.6 高内聚，低耦合	154
10.6 小结	154
第11章 同一类型多个元素的集合——简单数组	155
11.1 什么是数组	155

11.1.1	数组是一大片连续内存空间	155
11.1.2	数组元素的访问	156
11.2	一维数组	156
11.2.1	一维数组的声明	156
11.2.2	一维数组元素的访问	156
11.2.3	数组的初始化	157
11.2.4	不合法的数组操作	158
11.3	二维数组	159
11.3.1	二维数组的声明	159
11.3.2	二维数组的初始化	159
11.3.3	二维数组应用举例	160
11.4	更高维的数组	161
11.4.1	高维数组的声明和元素访问	161
11.4.2	初始化	161
11.4.3	多维数组在内存中是如何排列元素的	162
11.5	小结	163
第12章 C语言难点——指针初探		164
12.1	计算机中的内存	164
12.1.1	内存地址	164
12.1.2	内存中保存的内容	164
12.1.3	地址就是指针	165
12.2	指针的定义	165
12.2.1	指针变量的声明	165
12.2.2	指针变量的初始化	166
12.2.3	指针变量的值	166
12.2.4	取地址操作符&	166
12.2.5	指针变量占据一定的内存空间	166
12.2.6	指向指针的指针	167
12.3	使用指针	168
12.3.1	运算符*	168
12.3.2	指针的类型和指针所指向的类型	169
12.3.3	同类型指针的赋值	169
12.3.4	指针的类型和指针所指向的类型不同	170
12.4	指针的运算	172
12.4.1	算术运算之“指针+整数”或“指针-整数”	172
12.4.2	指针-指针	173
12.4.3	指针的大小比较	174
12.5	指针表达式与左值	174
12.5.1	指针与整型	175
12.5.2	指针与左值	175

12.5.3	指针与const	175
12.6	动态内存分配	176
12.6.1	动态分配的好处	177
12.6.2	malloc与free函数	177
12.6.3	calloc与free函数	178
12.6.4	free函数与指针	179
12.6.5	内存泄漏	180
12.6.6	释放已经释放了的内存会出问题	180
12.7	小结	180
第13章 字符串及字符串操作		181
13.1	C风格字符串	181
13.1.1	C风格字符串的声明	181
13.1.2	C风格字符串在内存中的表示	181
13.2	字符数组的输入输出	182
13.2.1	字符数组可以进行整体输入输出	182
13.2.2	使用gets函数读入整行输入	183
13.2.3	访问字符数组中某个元素	184
13.2.4	更便捷的输出——使用puts函数	184
13.3	字符串处理函数	185
13.3.1	理解：数组名是常指针	185
13.3.2	strlen函数与size_t	187
13.3.3	字符串复制函数strcpy	188
13.3.4	字符串比较函数strcmp	189
13.3.5	字符串连接函数strcat	190
13.3.6	全转换为大写形式	190
13.3.7	链式操作	191
13.4	小结	192
第14章 结构体、共用体、枚举和typedef		193
14.1	结构体	193
14.1.1	结构体的定义	193
14.1.2	声明结构体变量	194
14.1.3	初始化结构变量	194
14.1.4	访问结构体成员	194
14.1.5	结构体定义的位置	195
14.1.6	结构体变量赋值	196
14.2	特殊结构体	196
14.2.1	结构体嵌套	197
14.2.2	匿名结构体	200
14.3	共用体	200
14.3.1	什么是共用体	200

14.3.2	共用体的定义	200
14.3.3	声明共用体变量	201
14.3.4	共用体变量的初始化	201
14.3.5	共用体成员访问	201
14.3.6	共用体赋值	202
14.4	结构体和共用体的内存差异	202
14.4.1	结构体变量和共用体变量内存形式的不同	203
14.4.2	结构体变量的内存大小	203
14.4.3	字节对齐	204
14.4.4	最宽基本类型	204
14.4.5	共用体的大小	206
14.5	枚举类型	206
14.5.1	枚举类型的定义	206
14.5.2	声明枚举变量	207
14.5.3	枚举常量是什么	207
14.6	给类型取个别名——typedef	208
14.6.1	typedef基本用法	208
14.6.2	#define用法	208
14.7	小结	208
第15章	如何节省内存——位运算	209
15.1	什么是位运算	209
15.1.1	开灯关灯	209
15.1.2	改变状态	209
15.2	位逻辑运算符	210
15.2.1	位取反操作	210
15.2.2	位与运算	210
15.2.3	位或运算	210
15.2.4	位异或	210
15.2.5	实例分析	211
15.3	移位运算	212
15.3.1	基本形式	212
15.3.2	移位举例	212
15.4	小结	213

第三篇 C语言进阶

第16章	存储不仅仅局限于内存——文件	215
16.1	什么是文件	215
16.1.1	文件	215
16.1.2	流	216

16.1.3	重定向	216
16.1.4	文件的处理形式——缓冲区和非缓冲区	216
16.1.5	文件的存储形式——文本形式和二进制形式	217
16.2	C语言如何使用文件	217
16.2.1	文件型指针	217
16.2.2	文件操作的步骤	218
16.3	文件的打开与关闭	218
16.3.1	打开文件——fopen	218
16.3.2	打开是否成功	219
16.3.3	关闭文件——fclose	219
16.4	文件的读写	220
16.4.1	读写的相对参照	220
16.4.2	如何判断文件已经结束	220
16.4.3	字符读写函数——fgetc和fputc	220
16.4.4	字符串读写函数——fgets和fputs	222
16.4.5	块读写函数——fread和fwrite	224
16.4.6	格式化文件输入输出——fprintf与fscanf	227
16.5	文件的定位	229
16.5.1	移到开头——rewind	229
16.5.2	得到当前位置——ftell	229
16.5.3	移动指针——fseek	229
16.6	小结	230
第17章 灵活却难以理解——指针进阶		232
17.1	指针与数组	232
17.1.1	数组名指针	232
17.1.2	使用数组名常指针表示数组元素	233
17.1.3	指向数组元素的指针变量	233
17.1.4	指向数组的指针变量	234
17.1.5	指针数组	235
17.2	指针、结构体和结构体数组	236
17.2.1	两种访问形式	236
17.2.2	声明创建一个结构数组	236
17.2.3	结构数组的初始化	236
17.2.4	结构数组的使用	238
17.2.5	指向结构数组的指针	238
17.3	函数指针	239
17.3.1	函数名指针	239
17.3.2	指向函数的指针	240
17.3.3	函数指针类型	241
17.3.4	函数指针作函数参数	242