

高等学校计算机基础教育规划教材

C程序设计与应用

徐立辉 刘冬莉 编著

清华大学出版社

高等学校计算机基础教育规划教材

C程序设计与应用

徐立辉 刘冬莉 编著

清华大学出版社
北京

内 容 简 介

本书是为将 C 语言作为入门语言的程序设计课程编写的教材,目的是培养学生的程序设计基本能力和创新能力以及良好的程序设计风格。

本书以程序设计为主线,以编程应用为驱动,采取循序渐进、通俗易懂的方法,主要讲解程序设计的基本思想、方法,同时介绍了 C 语言相关的语法知识。本书以 2008 年全国计算机等级考试新需求为出发点,教学环境为 Visual C++ 6.0 平台。

本书的第 1 章介绍了 C 语言程序的基本结构、运行 C 语言程序的步骤与程序开发环境以及算法的表示。第 2 章介绍了 C 语言的数据类型、运算符与表达式。第 3~5 章介绍了基本控制结构组成,包括顺序、选择和循环结构。第 6 章介绍了数组,包括一维数组、二维数组和字符数组。第 7 章介绍了函数。第 8 章介绍了指针。第 9 章介绍了结构体与共用体。第 10 章介绍了文件。第 11 章介绍了 C 语言课程设计方案。第 12 章介绍了 UNIX/Linux 环境下的 C 语言编程入门。

本书可作为高等学校 C 语言程序设计课程的教学用书,也可作为自学 C 语言和参加全国计算机等级考试的参考书。

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 程序设计与应用/徐立辉,刘冬莉编著. —北京:清华大学出版社,2011.2

(高等学校计算机基础教育规划教材)

ISBN 978-7-302-24593-3

I. ①C… II. ①徐… ②刘… 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 012325 号

责任编辑:袁勤勇 薛 阳

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

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

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:20

字 数:469 千字

版 次:2011 年 1 月第 1 版

印 次:2011 年 1 月第 1 次印刷

印 数:1~3000

定 价:30.00 元

产品编号:040727-01

《高等学校计算机基础教育规划教材》

编 委 会

顾 问：陈国良 李 廉

主 任：冯博琴

副 主 任：周学海 管会生 卢先和

委 员：（按姓氏音序为序）

边小凡	陈立潮	陈 炼	陈晓蓉	鄂大伟
高 飞	高光来	龚沛曾	韩国强	郝兴伟
何钦铭	胡 明	黄维通	黄卫祖	黄志球
贾小珠	贾宗福	李陶深	宁正元	裴喜春
钦明皖	石 冰	石 岗	宋方敏	苏长龄
唐宁九	王 浩	王贺明	王世伟	王移芝
吴良杰	杨志强	姚 琳	俞 勇	曾 一
战德臣	张昌林	张长海	张 莉	张 铭
郑世钰	朱 敏	朱鸣华	邹北骥	

秘 书：袁勤勇

前言

程序设计是高等学校计算机基础课程,它是以编程语言为平台,介绍程序设计的基本思想、方法。

C语言是国内外广泛使用的一种程序设计语言,它除了具有强大的高级语言功能外,还具备低级语言的大部分功能,它已成为高校程序设计课程的首选语言。C语言程序设计是一门实践性很强的课程,它的教学重点应以程序设计为主,以介绍C语言相关的语法知识为辅,目的是培养学生的程序设计基本能力和良好的程序设计风格以及创新能力。

本书以程序设计为主线,以编程应用为驱动,采取循序渐进、通俗易懂的方法,主要讲解程序设计的基本思想、方法,同时介绍了C语言相关的语法知识。本书以2008年全国计算机等级考试新需求为出发点,教学环境为Visual C++ 6.0平台。

本书共12章。其中第1章介绍了C语言程序的基本结构、运行C语言程序的步骤与程序开发环境以及算法的表示。第2章介绍了C语言的数据类型、运算符与表达式。第3~5章介绍了基本控制结构组成,包括顺序、选择和循环结构。第6章介绍了数组,包括一维数组、二维数组和字符数组。第7章介绍了函数。第8章介绍了指针。第9章介绍了结构体与共用体。第10章介绍了文件。第11章介绍了C语言课程设计案例。第12章介绍了UNIX/Linux环境下的C语言编程入门。

本书由徐立辉进行整体策划。其中第1、8章和附录由徐立辉编写,第2章由李鹏、王永会编写,第3、4章由刘冬莉编写,第5、6章由冯毅宏编写,第7、10章由刘俊岭编写,第9、11章由何凯编写,第12章由牛志成、李鹏编写。全书由徐立辉、刘冬莉主编并统稿。

课堂教学使学生掌握程序设计的基本思想、方法,而要深刻理解还必须经过上机实验和大量的习题训练,以便学到课堂上无法学到的编程方法、程序调试方法和技巧。因此,我们还编写了配套的实验指导及习题,其中实验内容主要以Visual C++ 6.0为编程环境,由12个实验组成,每个实验都精心设计了编程样例或者调试样例、程序填空题、程序修改题和程序设计题。实验的项目按照C语言知识点展开,深入浅出,引导学生逐渐理解C语言程序设计的思想、方法和调试技巧;并且采用全国计算机等级考试题型,具有一定的实用性。习题部分精心选配了C语言教学内容的课外习题,涵盖了C语言的各种题型、各类数据类型、程序结构和典型算法。

随着计算机科学的不断发展,计算机教学的研究和改革也在不断深入。希望在从事计算机教学的各位同仁的共同努力下,不断提高我国高等学校 C 语言程序设计课程的教学水平。

由于作者水平有限,书中难免存在疏漏和不足之处,敬请读者批评指正。

编 者

2010 年 10 月

目录

第 1 章 C 程序设计概述	1
1.1 程序设计的基本概念	1
1.1.1 程序	1
1.1.2 程序设计	2
1.1.3 程序设计语言	2
1.2 C 语言简介	4
1.2.1 C 语言的发展历史	4
1.2.2 C 语言的特点	6
1.2.3 如何学习 C 语言	6
1.3 C 语言程序的基本结构	7
1.3.1 简单的 C 语言程序	7
1.3.2 C 语言程序的编写要求	10
1.4 运行 C 语言程序的步骤与程序开发环境	11
1.4.1 运行 C 语言程序的步骤	11
1.4.2 开发环境 Microsoft Visual C++ 6.0	12
1.4.3 开发环境 Turbo C 2.0	20
1.5 算法	21
1.5.1 算法概述	21
1.5.2 算法的表示	22
1.6 结构化程序设计的思想和方法	24
习题 1	25
第 2 章 数据类型与表达式	26
2.1 C 语言的数据类型	26
2.2 标识符、常量与变量	27
2.2.1 标识符	27
2.2.2 常量	28

2.2.3	符号常量	28
2.2.4	变量	29
2.3	整型数据	30
2.3.1	整型数据的表示	30
2.3.2	整型数据的存储	32
2.3.3	整型变量的定义与初始化	33
2.4	实型数据	33
2.4.1	实型数据的表示	33
2.4.2	实型数据的存储	34
2.4.3	实型变量的定义与初始化	34
2.5	字符型数据	35
2.6	运算符和表达式	36
2.6.1	C 运算符与表达式简介	36
2.6.2	算术运算符和算术表达式	37
2.6.3	赋值运算符和赋值表达式	40
2.6.4	逗号运算符和逗号表达式	42
2.6.5	关系运算符和关系表达式	43
2.6.6	逻辑运算符和逻辑表达式	43
2.6.7	条件运算符和条件表达式	44
2.6.8	位运算符和位运算表达式	44
2.7	数据类型转换	45
2.8	综合程序举例	47
	习题 2	50

第 3 章 顺序结构 51

3.1	C 程序的语句	51
3.1.1	C 语句概述	51
3.1.2	赋值语句	52
3.2	数据输出函数	53
3.2.1	printf 函数	54
3.2.2	putchar 函数	56
3.3	数据输入函数	57
3.3.1	scanf 函数	57
3.3.2	getchar 函数	60
3.4	标准库函数	61
3.5	顺序结构程序举例	63
	习题 3	66

第 4 章 选择结构	67
4.1 关系运算	67
4.1.1 关系运算符	67
4.1.2 关系表达式	67
4.2 逻辑运算	68
4.2.1 逻辑运算符	68
4.2.2 逻辑表达式	69
4.3 if 语句	70
4.3.1 if 语句和选择结构	70
4.3.2 if 语句的嵌套	74
4.3.3 条件运算符和条件表达式	76
4.4 switch 语句与 break 语句	77
4.5 选择结构程序举例	79
习题 4	82
第 5 章 循环结构	84
5.1 循环语句	84
5.1.1 while 语句	84
5.1.2 do-while 语句	85
5.1.3 for 语句	86
5.2 循环的嵌套	89
5.3 结束循环的语句	90
5.3.1 break 语句	90
5.3.2 continue 语句	91
5.4 循环结构应用举例	92
习题 5	96
第 6 章 数组	97
6.1 数组的概念	97
6.2 一维数组	97
6.2.1 一维数组的定义	97
6.2.2 一维数组的引用	99
6.2.3 一维数组的初始化	99
6.2.4 一维数组应用举例	100
6.3 二维数组	105
6.3.1 二维数组的定义	105
6.3.2 二维数组的引用	106

6.3.3	二维数组的初始化	107
6.3.4	二维数组应用举例	108
6.4	字符数组	110
6.4.1	字符数组的定义	111
6.4.2	字符数组的初始化	111
6.4.3	字符数组的引用	112
6.4.4	字符串	112
6.4.5	字符数组的输入输出	113
6.4.6	字符串处理函数	115
6.4.7	字符数组应用举例	118
习题 6		120
第 7 章	函数	122
7.1	函数的概念	122
7.2	定义函数的一般形式	123
7.3	函数的值和函数参数	125
7.3.1	实际参数和形式参数	125
7.3.2	函数的返回值	126
7.4	函数的调用形式	128
7.4.1	调用函数的一般形式	128
7.4.2	调用函数的方式	128
7.4.3	对被调用函数的原型声明	129
7.5	函数的嵌套调用	131
7.6	函数的递归调用	134
7.7	数组作为函数参数	136
7.7.1	数组元素作实参	136
7.7.2	数组名作函数的形参和实参	137
7.7.3	二维数组名作函数参数	139
7.8	局部变量与全局变量	139
7.8.1	局部变量	140
7.8.2	全局变量	141
7.9	变量的存储类别	143
7.9.1	静态存储方式与动态存储方式	143
7.9.2	自动型变量 auto	143
7.9.3	静态型变量 static	143
7.9.4	寄存器型变量 register	144
7.9.5	外部参照型变量 extern	145
7.9.6	用 static 声明全局变量	147

7.10	内部函数与外部函数	148
7.10.1	内部函数	148
7.10.2	外部函数	148
7.11	编译预处理	149
7.11.1	宏定义	149
7.11.2	文件包含	152
7.11.3	条件编译	153
习题 7		155
第 8 章	指针	156
8.1	指针的基本概念	156
8.1.1	变量的地址	156
8.1.2	指针变量	157
8.2	指针变量的定义与引用	157
8.2.1	指针变量的定义与初始化	157
8.2.2	指针变量的引用	158
8.2.3	使用指针变量作为函数参数	160
8.3	指针的运算	162
8.3.1	指针变量的赋值运算	162
8.3.2	指针变量的算术运算	164
8.3.3	指针变量的关系运算	165
8.4	指针与数组	165
8.4.1	指向一维数组元素的指针	165
8.4.2	使用指针引用一维数组元素	166
8.4.3	使用一维数组名或指向一维数组的指针作为函数的参数	167
8.4.4	指针与二维数组	171
8.4.5	指向二维数组元素的指针	172
8.4.6	使用指向二维数组的指针作为函数的参数	174
8.5	指向字符串的指针变量	175
8.5.1	字符串的表示形式	175
8.5.2	使用指向字符串的指针作为函数参数	178
8.5.3	字符指针变量与字符数组的区别	180
8.6	指针数组	180
8.6.1	指针数组的一般定义形式	180
8.6.2	指针数组的应用	181
8.7	指针数组作为 main 函数的形参	183
8.7.1	带参数的 main 函数的一般形式	183
8.7.2	命令行参数的应用	184

8.8	指向指针的指针变量	185
8.8.1	指向指针的指针变量的一般定义形式	186
8.8.2	指向指针的指针变量的应用	186
8.8.3	多级指针的概念	188
8.9	指向函数的指针变量	188
8.9.1	指向函数的指针变量的一般定义形式	188
8.9.2	使用函数指针变量调用函数	189
8.9.3	使用指向函数的指针作为函数参数	190
8.10	返回指针的函数	192
8.10.1	返回指针的函数的定义形式	192
8.10.2	返回指针的函数的应用	192
8.11	指向 void 的指针变量和指针的数据类型小结	193
8.11.1	指向 void 的指针变量	193
8.11.2	指针的数据类型的小结	194
8.12	指针程序举例	194
	习题 8	203

第 9 章 结构体与共用体

204

9.1	结构体	204
9.1.1	结构体类型和结构体变量	204
9.1.2	结构体类型的定义	205
9.1.3	结构变量的定义	206
9.2	结构体变量成员的引用方法	208
9.2.1	结构体变量的引用	208
9.2.2	结构体变量的赋值	208
9.3	结构体变量的初始化	209
9.4	结构体数组	209
9.5	结构体指针	211
9.5.1	结构体指针定义	211
9.5.2	指向结构体数组的指针	213
9.5.3	结构体在函数传递中的应用	214
9.6	动态存储分配	215
9.7	用结构体和指针处理链表	217
9.7.1	链表的概念	217
9.7.2	建立链表	219
9.7.3	输出链表	220
9.7.4	对链表的插入操作	220
9.7.5	对链表的删除操作	222

9.8	共用体	226
9.8.1	共用体概念	226
9.8.2	共用体类型的定义	227
9.8.3	共用体变量的定义	227
9.8.4	共用体变量成员的引用方法	228
9.9	枚举类型	229
9.9.1	枚举类型的定义	229
9.9.2	枚举变量的说明	230
9.9.3	枚举变量的赋值和使用	230
9.10	类型定义符 typedef	232
9.10.1	典型用法	232
9.10.2	典型类型的别名定义形式	233
习题 9		234
第 10 章	文件	236
10.1	文件概述	236
10.1.1	文件的基本知识	236
10.1.2	文件类型指针	237
10.2	文件的打开与关闭	238
10.2.1	打开文件	238
10.2.2	关闭文件	239
10.3	文件的读写	239
10.3.1	字符的输入输出	240
10.3.2	字符串的输入输出	241
10.3.3	文件的格式化输入输出	242
10.3.4	文件的数据块输入输出	242
10.4	文件的其他常用函数	244
10.4.1	文件的定位	244
10.4.2	出错检测	247
10.5	位运算与位运算符	247
10.5.1	按位与运算符(&)	248
10.5.2	按位或运算符()	248
10.5.3	按位取反运算符(~)	249
10.5.4	按位异或运算符(^)	249
10.5.5	左移运算符(<<)	250
10.5.6	右移运算符(>>)	251
10.5.7	位运算赋值运算符	251
10.5.8	位运算举例	252

习题 10	253
第 11 章 课程设计方案例	254
第 12 章 UNIX/Linux 下 C 语言编程入门	276
12.1 UNIX/Linux 简介	276
12.1.1 UNIX/Linux 简介	276
12.1.2 基于 Red Hat Linux 的实验环境	277
12.2 源程序编辑	279
12.2.1 vi 的工作状态及常用命令	279
12.2.2 vi 基本操作	279
12.3 编译与调试	280
12.3.1 使用 gcc 编译	280
12.3.2 使用 gdb 调试	282
12.4 操作举例	283
12.4.1 基本操作举例	283
12.4.2 高级操作举例	284
附录 A C 语言中的关键字	287
附录 B 运算符优先级和结合性	288
附录 C 常用字符与 ASCII 编码对照表	290
附录 D ANSI C 的常用库函数	292
附录 E VC++ 6.0 菜单中英文对照	298
参考文献	303

第 1 章

C 程序设计概述

程序是一系列计算机指令的有序组合。著名的计算机科学家沃思(Nikiklaus Wirth)提出了一个公式：

程序 = 算法 + 数据结构

其中,算法是指对操作的描述,即解决问题的具体操作步骤;数据结构是指对数据的描述,即数据的类型和组织形式;程序是算法在计算机中的实现。在实际应用中,一个程序除了包括算法和数据结构以外,还应该使用某种计算机语言来表示并且采用结构化程序设计方法进行程序设计。例如对 10 个整数从小到大排序,程序中可以使用冒泡法作为排序算法,使用整型数组来组织存储数据,使用 C 语言来表示并且采用结构化程序设计方法编写程序。

程序设计语言是人与计算机交流使用的“语言”。日常生活中人与人之间的交流需要使用某种彼此理解的语言,例如汉语、英语。同样,程序员或操作人员使用某种“计算机语言”与计算机交流并通过语言的规范来设计程序以便控制计算机工作,从而完成指定的任务。就像人类使用不同的语言一样,计算机语言也有许多种不同的语言,它们各自有不同的特点,适合在不同的场合使用。C 语言是目前国内外广泛使用的一种计算机语言,它的功能丰富、表达能力强、使用灵活方便、应用面广、可移植性好,既具有高级语言的优点,又具有低级语言的许多特点,因此学习 C 语言对掌握程序设计十分重要。

1.1 程序设计的基本概念

1.1.1 程序

一个完整的计算机系统是由硬件系统和软件系统构成的。例如,一台新组装的计算机只是硬件设备,用户不能做任何工作,只有安装了各种软件后计算机才能完成不同的工作。安装了办公自动化软件就可以打字、排版、制作电子表格和幻灯片,安装了多媒体播放软件就可以听 MP3、看电影,安装了上网软件就可以浏览网页、聊天。

软件是程序、数据以及相关文档的完整集合。程序是软件中最重要的部分,计算机的工作都是在程序的控制下进行的。

程序是计算机语言的语句序列,更确切地说是一系列指令的有序组合。它存储在计算机中,当计算机执行程序时,将自动按照一定的顺序逐条地调用指令来完成工作。

指令是计算机最基本的处理数据的单元。一条指令只能完成一个最基本的功能,如实现一次加法运算。计算机所有能实现的指令的集合称为计算机的指令系统。

1.1.2 程序设计

程序设计是指使用某种计算机语言并采用合适的方法编写程序,以便指挥计算机解决具体的问题。

程序设计一般包括以下几个部分:

- (1) 分析问题。
- (2) 划分模块。模块化设计有利于开发大型软件。当程序出现错误时,只要修改相关的模块及其连接即可。
- (3) 确定数据结构。根据要求,确定存储数据的数据结构。
- (4) 确定算法。确定解决问题的方法和步骤。
- (5) 编写代码。选择某种计算机语言(如 C 语言),编写程序代码。
- (6) 调试程序。在计算机上输入和调试程序,消除语法错误、逻辑错误和运行错误等,用各种可能的数据进行测试,使之得到正确的结果。
- (7) 分析、整理结果并且写出相关文档。

1.1.3 程序设计语言

在日常生活中,人与人之间的交流需要使用某种彼此理解的语言,例如汉语、英语、俄语、日语和法语等。同样,人们为了编制计算机程序以便于人与计算机交流,也发明了各种计算机专用的“语言”。这些用来编写程序的语言,就是程序设计语言。

程序设计语言一般分为机器语言、汇编语言和高级语言三大类。

1. 机器语言

机器语言是离计算机硬件最接近的语言。在计算机发展初期,程序员使用机器语言编写程序。机器语言中的每一条指令都是用二进制形式表示的,每一条指令都由操作码和地址码组成,计算机的硬件可以直接识别。

例如:使用机器语言编写程序完成 $1+2$ 功能。

```
1011 0000 0000 0001    /* 将 1 送到 CPU 中的累加器中 */
0000 0100 0000 0010    /* 将 2 与累加器中保留的内容相加,结果还放在累加器中 */
      1111 0100    /* 程序结束,停机 */
```


运行：机器语言编写的程序中的指令可以由硬件直接执行。

优点：执行效率高，能充分发挥计算机硬件的速度性能。

缺点：编写程序难度很大，容易出错，不易理解，通用性差。

2. 汇编语言

为了克服机器语言带来的不便，人们又发明了使用助记符(英文缩写)来帮助程序指令的理解和记忆，如用指令助记符来表示操作码(如 ADD 表示累加，MOVE 表示数据传送)，用地址符号来表示地址码(如用 AL 来表示某个累加器，代替二进制码)。用指令助记符及地址符号书写的指令称为汇编指令，使用汇编指令编写的程序称为汇编语言源程序。

例如：使用汇编语言编写程序完成 $1+2$ 功能。

```
MOVE AL,1          /* 将 1 送到累加器 AL 中 */
ADD AL,2            /* 将 2 与累加器 AL 中的内容相加，结果还放在累加器中 */
HLT                /* 程序结束，停机 */
```

运行：需要将汇编语言写成的源程序“翻译”成计算机硬件能直接识别的机器语言，这个“翻译”的过程叫汇编。完成汇编功能的软件叫汇编程序。

优点：比机器语言易理解、易编写、易检查、易修改。

缺点：通用性仍然不强，硬件不能直接执行，需要经过汇编过程。

3. 高级语言

机器语言和汇编语言都是面向机器的语言，一般称为低级语言。低级语言对机器依赖性太大，由它开发的程序通用性较差。

从 20 世纪 50 年代中期开始，发展出了面向解决问题的程序设计语言——高级语言。高级语言已经有上百种之多，其中广泛使用的有十几种，不同的语言有其最适合的应用领域，不同的语言也在不断推出新的版本。

几种常用的高级语言简介如下：

(1) BASIC(Beginners' All-purpose Symbolic Instruction Code, 初学者通用符号指令代码)语言适合初学者的教学和小型程序的开发。

如今，美国的 Microsoft 公司已经把可视化、模块化、面向对象程序设计等技术引入到 BASIC 中，并推出了 Visual Basic，它适合科学计算、事务处理、自动控制等。

(2) FORTRAN(Formula Translation, 公式翻译)语言适合科学与工程计算。

(3) FOXPRO 语言适合数据库管理程序的开发。

(4) PASCAL 语言适合计算机专业教学。

(5) COBOL(Common Business Oriented Language, 公共商用语言)适合商业、企业管理等。

(6) C 语言是最靠近机器的程序设计语言。它在科学计算、事务处理、自动控制以及数据库技术等方面都有广泛应用，它也适合系统软件和硬件底层的开发。