

21世纪高等学校计算机教育实用规划教材

新编C程序设计案例教程

张秀国 主编

马金霞 刘 博 宋传磊 编著

清华大学出版社

21世纪高等学校计算机教育实用规划教材

新编C程序设计案例教程

张秀国 主编
马金霞 刘 博 宋传磊 编著

清华大学出版社
北京

内 容 简 介

本书通过解决一些实际案例引出 C 语言的相关知识点,全面介绍了 C 语言程序设计中常用的选择结构、循环结构、数组、函数、指针、结构体、文件等知识点。通过进行案例分析学习理论知识,启发读者如何利用 C 语言去解决实际问题,以提高分析问题、解决问题的能力。

全书共分 4 篇:第 1 篇(第 1~4 章)为基础篇,着重介绍编程中必备的计算机基础知识、数据的表示、存储、数据运算等知识点;第 2 篇(第 5~7 章)为流程控制篇,着重讨论程序设计中选择结构、循环结构及模块化功能函数的使用方法;第 3 篇(第 8~11 章)为数据操作篇,着重介绍解决实际问题时对数据进行处理的基本方法和技术;第 4 篇(第 12~14 章)为高级应用篇,主要讲述编程中常用的一些库函数及文件的使用方法,最后通过一个综合案例来说明 C 语言程序设计的应用。全书提供了大量的应用实例,每章后均附有习题。

本书适用于 C 语言程序设计的初学者,既可以作为应用型高等院校中计算机、软件工程专业本科生、专科生的教材,也可以作为非计算机专业学生及有兴趣学习 C 语言的自学教材。

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

新编 C 程序设计案例教程/张秀国主编. --北京:清华大学出版社,2015

21 世纪高等学校计算机教育实用规划教材

ISBN 978-7-302-40437-8

I. ①新… II. ①张… 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(2015)第 122219 号

责任编辑:刘 星 薛 阳

封面设计:常雪影

责任校对:李建庄

责任印制:宋 林

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:18.5 字 数:463 千字

版 次:2015 年 8 月第 1 版 印 次:2015 年 8 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:064123-01

前 言

随着信息技术和计算机技术的快速发展,计算机已经渗入到各个专业。对非计算机专业人才来说,掌握一门计算机编程语言能够加深理解对本专业软件的应用,从而提高专业软件的操作能力。C语言是使用最为广泛的计算机基础语言,无论是计算机专业人员,还是非计算机专业人员,学好C语言可以为今后的学习、工作打下专业基础。

在很多应用型院校中,都将C语言作为一门极为重要的专业基础课程。C语言作为程序设计课程之一,主要培养学习者掌握基本问题的求解过程和基本思路,学会如何应用计算机解决实际问题和分析问题。为了能够帮助读者较好地学习和掌握好C语言,提高学习者对编程语言的学习兴趣,作者结合对编程语言教学改革,编写了案例式教学教材。案例式教学是以案例的分析解决为主线,通过对案例中的问题进行分析讨论,激发学生的求知欲和主动性,教给学生分析问题和解决问题的方法和道理。它既是一种互动式的教学方法,更是实现理论联系实际的有效手段。鉴于此,本教材讲解知识点的主要思路是先提出问题,然后分析问题,最后解决问题。在课堂上,采用引导式教学方法,使在学习编程语言过程中轻松入门,领会编程语言的魅力,真正掌握分析解决问题的能力。

本书具有如下特色。

1. 突出案例式驱动教学

分析归纳了C语言的主要知识点,精选出典型案例,通过对每个具体案例的分析引出相关知识点,介绍知识点,最后应用知识点解决问题,帮助读者了解C语言中各个知识点的学习目的性。

2. 通俗易懂,层次分明

本书按照编程语言的特点,分为基础篇、流程控制篇、数据操作篇、高级应用篇,循序渐进,由简单到复杂,做到讲解每个知识点具有较强的目的性和针对性,并有一定的层次性。基础篇介绍C语言数据类型、基本结构等基础知识,帮助读者认识编程语言,学会解决实际生活中的简单问题;流程控制篇主要介绍选择结构、循环结构、函数等用来控制程序执行顺序的相关内容,通过设计出多样的程序流程结构来解决实际生活中较为复杂的问题;数据操作篇主要介绍数组、字符串、指针、结构体等数据处理的相关内容。学习本部分内容使读者在解决问题过程中,既能较方便地对数据进行处理,又能降低处理数据的复杂度;高级应用篇主要介绍如何使用C语言提供的标准库函数及文件等内容来说明编程语言的实际应用。

3. 实用性强

本教材的实用性主要表现在:有较强的课堂操作性,让教师感到便于进行知识的传授和技能的训练,让学习者感到学起来方便容易;通过对实际问题进行分析,让学习者明显感

到学习编程语言的用处所在；内容编排上遵循由易到难，注重编程语言的内在联系和区别，使自学本书者也能得心应手。

4. 知识点全面，讲解内容简练

本书涵盖了许多 C 语言教材中未讲解的内容，如标准库函数、复杂指针的使用方法等。对每个知识点的讲解都是以案例为中心进行展开，注重了对知识点的“用”。

适用的读者：

本书适用于学习计算机编程语言类课程的初学者使用，也可以作为应用型本科院校、高职高专院校中各专业学习程序设计基础的教材，还可以供有兴趣学习 C 语言的读者进行自学。

出版说明：

本书主要以程序设计为主，未涉及具体的编程环境，整套教材提供的代码都可以在 Visual C++、Borland C++、Turbo C、Dev C++ 中运行。

本书为作者在多年对编程语言类课程教学与程序设计实践的基础上，结合多次编写相关讲义的经验总结而成。全书共分为 14 章，其中第 1~4 章由刘博编写，第 5、11、12 章由宋传磊编写，第 6、7、13、14 章由马金霞编写、第 8~10 章由张秀国编写。最后全书由张秀国统编定稿。

在本书的写作过程中，江连海、王磊提供了编写本书所用到的附录及习题，王小妮、吴伟伟、薛晓亚对本书的资料进行了整理及校对，在此向他们表示衷心的感谢。

限于编者的水平和经验，加之时间比较仓促，书中疏漏或者不足之处在所难免，敬请读者批评指正。有兴趣的朋友可发送邮件到 zxg@qdc.edu.cn 与作者交流。

编 者

2015 年 3 月



基 础 篇

第 1 章 编程中必备的基础知识	3
1.1 编程语言	3
1.2 计算机基础知识	4
1.2.1 计算机的工作过程	4
1.2.2 计算机系统的组成	5
1.2.3 存储器	6
1.3 数制及其转换与数值型数据的存储表示	7
1.3.1 数制进位中的基本概念	7
1.3.2 数制之间的转换	8
1.3.3 二进制数的运算	11
1.3.4 数据在计算机内部的存储形式	12
1.4 算法	13
1.4.1 算法描述方法	14
1.4.2 算法特性	15
习题 1	15
第 2 章 从认识 C 语言开始	18
2.1 从显示“Hello World!”开始	18
2.2 常量	19
2.2.1 整型常量	19
2.2.2 实型常量	20
2.2.3 字符型常量	20
2.2.4 字符串常量	21
2.3 常用的运算符	21
2.3.1 算术运算符与算术表达式	21
2.3.2 关系运算符与关系表达式	21
2.3.3 逻辑运算符和逻辑表达式	22
2.3.4 条件运算符和条件表达式	23

2.3.5	赋值运算符及赋值表达式	24
2.3.6	逗号运算符及逗号表达式	25
2.3.7	自增、自减运算符	26
2.4	综合运算	27
2.4.1	运算符之间的优先级	27
2.4.2	综合运算实例	27
习题 2		28
第 3 章	数据类型及变量	31
3.1	数据类型	31
3.2	变量	32
3.2.1	标识符	32
3.2.2	对变量的理解	33
3.2.3	定义变量	34
3.2.4	变量的初始化	34
习题 3		34
第 4 章	数据的输入与输出	37
4.1	格式化输入输出	37
4.1.1	printf 函数	37
4.1.2	scanf 函数	42
4.2	字符输入输出函数	46
4.2.1	putchar 函数	47
4.2.2	getchar 函数	48
习题 4		49

流程控制篇

第 5 章	选择控制语句	55
5.1	案例一 温度转换	55
5.1.1	案例描述及分析	55
5.1.2	单分支 if 语句	56
5.1.3	程序实现	57
5.1.4	双重分支 if...else 语句	58
5.2	案例二 计算股票经纪人的佣金	59
5.2.1	案例描述及分析	59
5.2.2	多分支结构	60
5.2.3	程序实现	61
5.3	案例三 判定成绩等级	62

5.3.1	案例描述及分析	62
5.3.2	switch 语句	63
5.3.3	程序实现	64
习题 5		65
第 6 章	循环控制语句	70
6.1	案例一 猴子吃桃.....	70
6.1.1	案例描述及分析	70
6.1.2	while 语句	71
6.1.3	程序实现	72
6.1.4	do...while 语句	74
6.2	案例二 判定素数.....	74
6.2.1	案例描述及分析	74
6.2.2	for 语句.....	75
6.2.3	程序实现	79
6.3	案例三 由星组成的倒三角.....	80
6.3.1	案例描述及分析	80
6.3.2	循环的嵌套	81
6.3.3	程序实现	83
6.4	案例四 猜数游戏.....	84
6.4.1	案例描述及分析	84
6.4.2	break 语句	85
6.4.3	continue 语句	87
6.4.4	具体实现	88
习题 6		90
第 7 章	模块化函数	94
7.1	案例一 简单计算器.....	94
7.1.1	案例描述及分析	94
7.1.2	函数概述	95
7.1.3	函数的定义	97
7.1.4	函数的参数和函数的值	98
7.1.5	函数的调用.....	100
7.1.6	程序实现.....	102
7.2	案例二 数值交换的“废品”	105
7.2.1	案例描述及分析.....	105
7.2.2	函数间的参数传递.....	105
7.2.3	变量的作用域.....	106
7.2.4	程序实现.....	109

7.3 案例三 求阶乘	111
7.3.1 案例描述及分析	111
7.3.2 函数的嵌套调用	112
7.3.3 函数的递归调用	112
7.3.4 程序实现	113
习题 7	115

数据操作篇

第 8 章 数组的妙用	121
8.1 案例一 找最大	121
8.1.1 案例描述及分析	121
8.1.2 一维数组的定义	122
8.1.3 一维数组的初始化	123
8.1.4 数组元素的引用	124
8.1.5 程序实现	125
8.2 案例二 排序	126
8.2.1 案例描述及分析	126
8.2.2 冒泡法排序	126
8.2.3 选择法排序	129
8.3 案例三 魔方阵	131
8.3.1 案例描述及分析	131
8.3.2 二维数组的定义	131
8.3.3 二维数组的初始化	133
8.3.4 二维数组的引用	134
8.3.5 程序实现	134
8.4 案例四 链对计数器	136
8.4.1 案例分析及描述	136
8.4.2 程序实现	137
8.5 案例五 赛马	138
8.5.1 案例描述及分析	138
8.5.2 数组元素作函数的实参	138
8.5.3 程序实现	138
习题 8	140
第 9 章 指针的灵活运用	144
9.1 案例一 数值交换的“真品”	144
9.1.1 案例描述及其分析	144
9.1.2 地址、指针和指针变量	144

9.1.3	指针变量的定义与赋值	146
9.1.4	指针变量作为函数参数	150
9.1.5	程序实现	150
9.2	案例二 猴子选大王	152
9.2.1	案例描述及分析	152
9.2.2	指针变量与一维数组	152
9.2.3	程序实现	154
9.3	案例三 一维数组中的“大在前小在后”	155
9.3.1	案例描述及分析	155
9.3.2	数组元素的地址作为函数实参	156
9.3.3	数组名作为函数参数	156
9.3.4	程序实现	157
9.4	案例四 二维数组中的“大在前小在后”	159
9.4.1	案例描述及分析	159
9.4.2	二维数组元素的地址	159
9.4.3	用二维数组名作函数的参数	161
9.4.4	程序实现	162
9.5	案例五 一组数据的累加、累乘计算	165
9.5.1	案例描述及分析	165
9.5.2	返回指针的函数	165
9.5.3	指向函数的指针变量	166
9.5.4	程序实现	168
	习题 9	169
	第 10 章 字符串的处理	173
10.1	案例一 存储“Hello World!”	173
10.1.1	案例描述及分析	173
10.1.2	用字符数组存放字符串	173
10.1.3	指针变量指向一个字符串	175
10.1.4	字符串的输入和输出	176
10.1.5	程序实现	178
10.2	案例二 单词计数器	178
10.2.1	案例描述及分析	178
10.2.2	字符串的输入函数 gets 和输出函数 puts	179
10.2.3	用指针变量处理字符串	180
10.2.4	程序实现	181
10.3	案例三 单词排序	182
10.3.1	案例描述及分析	182
10.3.2	指针数组的定义与使用	182

10.3.3	字符串作函数参数	184
10.3.4	程序实现	185
习题 10		187
第 11 章	结构体的构造	190
11.1	案例一 建立学生信息库	190
11.1.1	案例描述及分析	190
11.1.2	结构体类型的定义	191
11.1.3	结构体变量的定义及初始化	191
11.1.4	引用结构体变量中的成员	193
11.1.5	结构体数组的定义与初始化	194
11.1.6	程序实现	195
11.2	案例二 按学生成绩排序	197
11.2.1	案例描述及分析	197
11.2.2	指向结构体类型的指针变量	198
11.2.3	指针变量引用结构体中的成员	198
11.2.4	程序实现	199
11.3	案例三 统计候选人票数	201
11.3.1	案例描述及分析	201
11.3.2	用结构体类型的数据作函数参数	202
11.3.3	程序实现	204
11.4	案例四 创建链表	206
11.4.1	案例描述及分析	206
11.4.2	利用结构体变量构成链表	206
11.4.3	动态存储分配	207
11.4.4	程序实现	208
习题 11		211

高级应用篇

第 12 章	常用的库函数	219
12.1	案例一 多功能计算器的制作	219
12.1.1	案例描述及分析	219
12.1.2	数学计算 math.h	220
12.1.3	程序实现	223
12.2	案例二 显示提醒列表	224
12.2.1	案例描述及分析	224
12.2.2	字符串处理函数	225
12.2.3	程序实现	227

12.3 案例三 数值转换	230
12.3.1 案例描述及分析	230
12.3.2 通用的实用工具函数	230
12.3.3 程序实现	231
习题 12	233
第 13 章 文件	236
13.1 案例一 建立一个存储学生基本信息的信息文件	236
13.1.1 案例描述及分析	236
13.1.2 创建新文件	236
13.1.3 程序实现	240
13.2 案例二 文件复制	242
13.2.1 案例描述及分析	242
13.2.2 文件的其他常用读写函数	242
13.2.3 程序实现	244
习题 13	245
第 14 章 综合案例——贪吃蛇游戏	248
14.1 案例描述及分析	248
14.2 程序设计	248
14.3 代码实现	249
附录 A 常用字符与 ASCII 代码对照表	256
附录 B C 语言关键字及其用途	257
附录 C 运算符的优先级和结合性	258
附录 D C 语言标准库函数	259
附录 E 课后习题参考答案	265
参考文献	284

基 础 篇

通过计算机来解决实际问题,事实上是将实际问题的信息转换成数据,计算机通过对数据进行一定的计算处理,得到运算结果再转换成信息,从而解决实际问题。计算机对数据的处理过程是通过编程语言来实现的,编程语言是一门计算机语言,既然是语言,就需要按照一定的规则进行组织,像汉语、英语等一样,都有各自一整套的语法规则。在编程语言中,对数据是如何进行表示的?如何进行存储的?如何进行计算的?这三个问题是学习一门编程语言的入门知识。在本篇中,主要对常量、变量、运算符、输入输出进行详细介绍,学好本篇内容是掌握编程语言的基础。

第 1 章

编程中必备的基础知识

学好编程语言,首先要掌握一些计算机的基础知识,主要包括计算机的工作原理、计算机的组成、数据的存储形式以及编写程序所遵循的算法等。只有清楚了这些知识,学习编程才能更加轻松和得心应手。

1.1 编程语言

对于计算机而言,并不会因为程序员想什么,它就做什么,它只能根据程序员编写的程序指令完成相应的操作。虽然这些程序指令是由程序员下达的,但是,需要计算机必须识别并可以根据指令表达的含义完成相应的操作。那么,程序员和计算机到底是如何交流的呢?

人与人之间交流需要语言,人和计算机之间交流同样也需要语言。只是这种语言不是单纯的汉语、英语,而是符合计算机语法的语言,这种计算机能够响应、程序员能够理解的语言被称为计算机程序设计语言,也叫编程语言。编程语言是人和计算机交流、沟通的桥梁。

当前可用的编程语言以各种形式和类型出现,这些不同形式和类型的语言使编程过程更容易,它们或者专门针对某个硬件的特别属性,或者满足某个应用程序的特殊要求。但是,在基本层面,所有的程序都必须最终转换成机器语言程序,这是能够实际操作计算机的唯一程序类型。事实上,可以把程序设计语言分为机器语言、汇编语言和高级语言三类。

1. 机器语言

机器语言是直接采用二进制代码表示指令的计算机语言。指令是由若干个 0 和 1 组成的一串代码,它们有一定的位数,并分成若干段,各段的编码表示不同的含义。例如,某台计算机字长为 16 位,即用 16 个二进制位组成一条指令或其他信息。16 个 0 和 1 可组成各种排列组合,通过线路变成电信号,进而使得计算机能够执行各种不同的操作。例如,一个简单的包含两条指令的机器语言程序是:

```
110000000000000000001000000000010  
111100000000000000001000000000011
```

每一个构成一条机器语言指令的二进制数字序列最少由两部分组成:指令部分和数据部分。指令部分称为操作码,告诉计算机要执行的操作,如加、减、乘等。二进制数的其余部分提供有关数据的信息。

机器语言是一种低级语言,它是最接近计算机硬件的一种语言,也是计算机唯一直接识别的语言。也就是说,无须翻译,计算机就能识别机器语言并可以执行相关的操作。因此,采用机器语言进行编码,计算机的执行效率较高。但是,由于每一条指令都是由若干个二进

制数来表示,所以,编写机器语言的程序难度大、直观性差、容易出错和不易调试,给程序员带来了诸多不便。

2. 汇编语言

汇编语言,又称助记符语言,就是采用单词风格的符号替代机器语言中的二进制操作码。如 ADD、SUB、MUL 等。虽然相对于机器语言来说,汇编语言减轻了程序员烦琐的编程压力,但是,汇编语言必须经过翻译才能被计算机识别,而这个翻译的过程被称为汇编过程。

汇编语言比机器语言易于读写、调试和修改,同时具有机器语言的全部优点。但在编写复杂程序时,汇编语言相对高级语言而言,其代码量较大,而且汇编语言依赖于具体的处理器体系结构,即不能在不同处理器体系结构之间进行移植。所以,汇编语言适合编写一些有速度要求的程序或直接控制硬件的程序。

3. 高级语言

机器语言和汇编语言一般都称为低级语言。

高级语言是较接近自然语言和数学公式的编程语言,基本上脱离了机器的硬件系统,能够用人们更易理解的方式编写程序。高级语言并不是特指某一种具体的语言,而是包括很多种编程语言,如目前流行的 BASIC、C、C++、Delphi、Java 等,这些语言的语法、命令格式都不相同。

高级语言与计算机的硬件结构及指令系统无关,它具有更强的表达能力,可方便地表示数据的运算和程序的控制结构,能更好地描述各种算法,而且容易学习掌握。但高级语言编译生成的程序代码一般比用汇编程序语言设计的程序代码要长,执行的速度也较慢。

三种程序设计语言各有特色,使用的时期和领域也有所不同。可以根据自己的实际情况,选择不同的编程语言。但是,使用比较多的还是高级语言,而本书要介绍的 C 语言就是高级语言中的一种。虽然 C 语言是高级语言,但它也具有低级语言的某些特点,能够处理一些硬件控制操作,这也是 C 语言应用广泛的一个最基本的原因。

1.2 计算机基础知识

计算机的发展经历了电子管时代、晶体管时代、中小规模集成电路时代、大规模和超大规模集成电路时代,未来将迎来光子和量子计算机时代。到目前为止,无论是哪一个计算机时代,计算机的设计始终遵循冯·诺依曼式体系结构,即:计算机都是由运算器、控制器、存储器、输入设备和输出设备 5 大部分组成,整个工作过程遵循着存储程序控制的原理,并在其内部都是以二进制的形式存储数据来实现的。

1.2.1 计算机的工作过程

有人认为计算机非常神奇,也非常强大,玩游戏、看视频、网购等什么事情都能完成。其实,它所有功能的体现就是在进行数据运算。比如说,在计算机中玩一个战斗游戏,玩家拿枪打中敌人,敌人中弹倒地。实际上,计算机就是在计算枪口射出的子弹和敌人的坐标数据是否相同,如果相同,则显示一张敌人中弹倒地的图片。也就是说,计算机做的任何事情,实现的任何功能都是在计算数据。

前面说过,计算机是由运算器、控制器、存储器、输入设备和输出设备 5 大部件组成。输入设备也就是向计算机进行输入数据的设备,如键盘;输出设备就是将计算的数据结果进行输出的设备,如显示器;那么,运算器、控制器、存储器三者之间又是如何工作的呢?

为了说明这个问题,现在假设有一个没有记忆能力的人,其身边有一套纸笔和计算器,该计算器只能进行两个数据的算术运算,即不能进行多个数据的连算。现让他计算出 $(5+7) \times (2+6)$ 的结果,其步骤如下。

Step1: 向计算器中分别输入 $5+7$ 。

Step2: 得出结果 12,将结果 12 写到纸上。

Step3: 向计算器中分别输入 $2+6$ 。

Step4: 得出结果 8,将结果 8 写到纸上。

Step5: 从纸上读出 12、8 的数据输入到计算器中,计算 12×8 。

Step6: 得出结果 96,将结果 96 写到纸上。

Step7: 从纸上读出数据 96,输出最终结果。

从以上解决问题的步骤来看,有人、计算器、纸三个主要元素,事实上,其工作分别对应了控制器、运算器、存储器。

这样也就不难理解计算机 5 大部件的工作过程是:首先,将程序和数据通过输入设备送入存储器;然后,计算机从存储器中依次取出程序指令送到控制器进行识别和分析该指令的功能;控制器根据指令的含义发出相应的命令(如加法、减法),将存储单元中存放的操作数取出送往运算器进行运算,再把运算结果送回存储器指定的单元中;最后,计算机可以根据指令将最终的运算结果通过输出设备进行输出。

了解了计算机的工作过程,实际上,也就清楚了计算机的“神经系统”。依照这个“神经系统”,对计算机整个系统的组成也就不难理解了。

1.2.2 计算机系统的组成

一个完整的计算机系统由硬件系统和软件系统两大部分组成。

1. 硬件系统

硬件系统主要包括运算器、控制器、存储器和输入、输出设备 5 大部件。其中,运算器和控制器构成中央处理器,也就是人们常说的 CPU(Central Processing Unit);存储器是存放数据的单元,可以分为内存和外存,内存主要指 RAM(Random Access Memory,随机存取存储器),外存主要指计算机中的硬盘;输入设备就是人们常说的键盘和鼠标,而输出设备就是显示器。计算机硬件系统组成如图 1-1 所示。

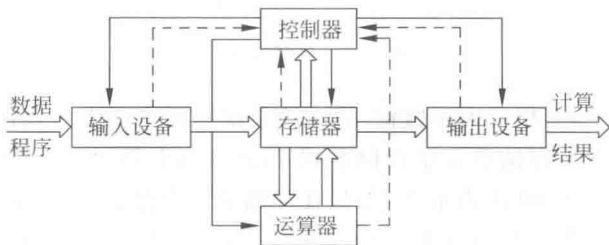


图 1-1 计算机硬件系统组成