



普通高等教育“十三五”规划教材

计算机程序设计

案例教程

赵秀岩 于晓强 ◎ 主编

非
外
借



科学出版社

普通高等教育“十三五”规划教材

计算机程序设计案例教程

赵秀岩 于晓强 主编

王美航 邵利 阎丕涛 副主编



科学出版社

内 容 简 介

本书以案例为主线,由案例引导知识点的讲解形式,重点介绍程序设计的过程、方法及计算思维的基本思想。

本书以 C 语言作为程序设计的实现语言,共分 11 章,详细介绍了 C 语言的数据类型与运算、程序设计基本结构、数组、函数、指针、其他构造数据类型、文件等内容。每章围绕相关知识点来设计典型案例,让学生在编程实践中更加直观、深刻地理解知识点。同时,案例中的“案例分析与思考”和“程序流程图”注重学生计算思维能力的培养,“拓展训练”和“技能提高”注重学生举一反三和实践能力的培养。

本书适合作为高等学校本科生计算机程序设计课程的教材,也可作为程序设计初学者的自学参考书,还可作为培训机构的培训教材。

图书在版编目(CIP)数据

计算机程序设计案例教程/赵秀岩,于晓强主编. —北京:科学出版社,2017
(普通高等教育“十三五”规划教材)

ISBN 978-7-03-054672-2

I. ①计… II. ①赵… ②于… III. ①程序设计—高等学校—教材
IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2017)第 242104 号

责任编辑:宋丽 王惠 / 责任校对:刘玉靖
责任印制:吕春珉 / 封面设计:东方人华平面设计部

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

铭浩彩色印装有限公司印刷

科学出版社发行 各地新华书店经销

*

2017 年 10 月第 一 版 开本:787×1092 1/16

2017 年 10 月第一次印刷 印张:29

字数:672 000

定价:72.00 元

(如有印装质量问题,我社负责调换〈骏杰〉)

销售部电话 010-62136230 编辑部电话 010-62135397-2052

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

随着信息技术的发展, 计算机应用技能及计算思维能力已成为当代大学生必备的基本素质。多数高校选择开设计算机程序设计这样的课程, 旨在培养学生利用计算机解决专业领域研究与应用问题的能力。

当前市场上多为传统教材, 存在两方面不足。一方面, 传统的程序设计教材注重编程语言的语法及规则介绍, 学生依照教材完成学习过程, 虽然可以掌握高级编程语言的语法规则, 但是对利用计算机解决问题的思路及问题求解过程不能完全理解, 不具备应用计算机独立解决实际问题的思维能力。另一方面, 传统的程序设计教材在结构设计上分为理论与实验两部分内容, 理论部分的知识介绍及案例与实验部分的内容之间没有进行紧密的关联设计, 使得学生既对理论知识的认识和理解感到抽象, 又对实验内容感到无从下手, 实验只能“照猫画虎”、一知半解地完成, 往往造成对学生的实践动手能力培养事倍功半。

本书是在总结传统教材存在的不足基础上, 结合编者多年的教学经验编写而成的, 具有如下特点:

(1) 在内容设计上, 改变了传统教材以语法为主线的内容组织方式, 而采取以案例为核心的组织方式, 以“提出问题—分析问题—解决问题—总结问题—应用”为主线, 旨在引导和帮助学生建立解决问题的常规思路, 掌握程序设计的思维过程。

(2) 在结构形式上, 实现了“理论实践一体化”的无缝设计。每个知识单元都设计成“案例组”的形式。每一组案例涉及 2~3 个程序, 第一个是理解程序, 主要由教师讲解, 包括案例设计目的、案例分析与思考(部分简单案例略)、案例主要代码、案例运行结果、案例小结; 第二个是拓展训练: 模仿程序, 问题难度、算法都与第一个程序十分类似, 由学生模仿练习; 第三个是技能提高(部分案例略): 提高程序, 在前面两个程序的基础上, 难度和算法略有变化和提升, 由学生独立完成。这样就会在潜移默化中培养学生在面对复杂问题时, 采用从简到繁的解决方法和思维方式。

(3) 本书突破了传统教材理论与实践分别组织的特点, 将理论教学和实践完美地结合在一起。这也是本书的创新之处。如果程序设计课程能够在机房授课, 即保证学生人手一机, 那么本书将是教学的不二选择。即使是在多媒体教室授课, 也可以选择本书, 同样可以取得良好的教学效果。

(4) 书后附录 I 给出了大量经典习题, 并附有答案, 可供学生进行同步练习, 也可供参加计算机等级考试的学生参考。

本书由大连工业大学计算机基础教研室教师组织编写, 由赵秀岩、于晓强任主编, 王美航、邵利、阎丕涛任副主编。具体分工如下: 第 1、2 章由康丽编写, 第 3、4 章由

邵利编写，第 5 章由刘英编写，第 6 章由房媛编写，第 7 章由赵秀岩编写，第 8 章由于晓强编写，第 9 章由王美航编写，第 10、11 章由贺晓阳编写，全书由赵秀岩、阎丕涛统稿。

由于编写时间紧张，加之编者水平有限，书中难免存在疏漏与不足之处，敬请广大读者和同行批评指正。

编 者
2017 年 7 月

目 录

第 1 章 概述	1
1.1 基本概念	1
1.2 初识 C 程序	2
1.3 C 程序的基本结构	4
1.4 运行 C 程序	5
1.4.1 Turbo C 2.0	5
1.4.2 VC++ 6.0	8
1.4.3 Win-TC	13
本章小结	15
第 2 章 程序设计基础	16
2.1 算法简介	16
2.2 数据类型	19
2.2.1 常量和变量	19
2.2.2 整型数据	22
2.2.3 实型数据	26
2.2.4 字符型数据	28
2.3 运算符和表达式	33
2.3.1 算术运算符和算术表达式	34
2.3.2 自增、自减运算符	35
2.3.3 关系运算符和关系表达式	37
2.3.4 逻辑运算符和逻辑表达式	39
2.3.5 赋值运算符和赋值表达式	41
2.3.6 逗号运算符和逗号表达式	44
2.3.7 条件运算符	45
2.3.8 位运算符	47
2.3.9 sizeof 运算符	49
2.3.10 各类型数据间的混合运算	51

2.4 宏常量与宏替换	53
本章小结	55
第 3 章 顺序结构与输入/输出	56
3.1 字符输入/输出函数	56
3.1.1 getchar()函数	56
3.1.2 putchar()函数	57
3.1.3 字符输入/输出函数的应用	58
3.2 格式输入/输出函数	60
3.2.1 scanf()函数概述	61
3.2.2 printf()函数概述	63
3.2.3 scanf()函数和 printf()函数的特殊格式控制	64
3.2.4 格式输入/输出函数的综合应用	66
本章小结	69
第 4 章 选择结构	71
4.1 if 语句	71
4.1.1 if 语句的单分支形式	71
4.1.2 if 语句的双分支形式	74
4.1.3 if 语句的多分支形式	77
4.2 if 语句的嵌套使用	83
4.2.1 if 语句嵌套的一般形式	83
4.2.2 if 语句嵌套的应用	84
4.3 switch 语句	86
本章小结	91
第 5 章 循环结构	92
5.1 while 循环语句	92
5.2 do...while 循环语句	95
5.3 for 循环语句	98
5.4 for 循环语句的特殊格式	100
5.5 用 goto 语句和 if 语句构成循环语句	104
5.6 循环的嵌套	106
5.7 break 语句和 continue 语句	108
5.7.1 break 语句	108
5.7.2 continue 语句	110

5.8 经典程序举例	112
本章小结	116
第6章 数组	118
6.1 数组的本质	118
6.2 一维数组	120
6.2.1 一维数组的定义	120
6.2.2 一维数组的引用	121
6.2.3 一维数组的初始化	123
6.2.4 一维数组的输入与输出	125
6.2.5 一维数组的应用	126
6.3 二维数组	132
6.3.1 二维数组的定义	132
6.3.2 二维数组的引用和初始化	134
6.3.3 二维数组的应用	137
6.4 字符数组和字符串	140
6.4.1 字符数组的定义、引用和初始化	140
6.4.2 字符串处理函数	143
6.4.3 字符数组的应用	149
本章小结	152
第7章 函数	153
7.1 函数的定义	154
7.2 函数的声明	160
7.2.1 库函数的声明	160
7.2.2 自定义函数的声明	161
7.3 函数的参数	161
7.4 函数的返回值	168
7.5 函数的调用关系	171
7.6 全局变量与局部变量	175
7.6.1 局部变量	175
7.6.2 全局变量	176
7.6.3 变量的存储类型与生存期	180
7.7 函数综合应用	186
本章小结	190

第 8 章 指针	191
8.1 指针的概念及指针变量	192
8.1.1 指针的概念	192
8.1.2 指针变量的定义	193
8.1.3 指针变量的赋值	194
8.1.4 指针运算符	195
8.2 指针的运算	199
8.2.1 指针的赋值运算	199
8.2.2 指针的算术运算	199
8.2.3 指针的关系运算	200
8.3 指针与数组	200
8.3.1 指向一维数组元素的指针变量	200
8.3.2 通过指针引用数组元素	201
8.3.3 指针与字符串	204
8.3.4 指向多维数组元素的指针变量	209
8.4 指针与函数	210
8.4.1 指针变量作为函数参数	210
8.4.2 数组名作为函数参数	213
8.4.3 返回指针值的函数	217
8.4.4 指向函数的指针	218
8.5 指向指针的指针	220
8.6 指针与二维数组	221
8.6.1 二维数组的结构	221
8.6.2 数组指针	223
8.6.3 指针数组	225
8.6.4 指针与二维字符数组	227
8.7 main()函数的参数	231
本章小结	233
第 9 章 结构体	235
9.1 结构体类型的定义	236
9.2 结构体变量定义	237
9.3 结构体数组	241
9.4 结构体与函数	247
9.5 结构体与指针	252

9.6 结构体成员变量为结构体类型	257
9.7 链表	260
9.7.1 链表的定义	261
9.7.2 链表的基本操作	264
第 10 章 共用体与枚举	280
10.1 共用体类型	281
10.1.1 共用体类型的定义	281
10.1.2 共用体变量的定义	281
10.2 枚举类型	285
10.2.1 枚举类型的定义	285
10.2.2 枚举变量的定义	286
本章小结	288
第 11 章 文件	289
11.1 流	289
11.1.1 文件指针	289
11.1.2 标准流和重定向	290
11.1.3 文件	290
11.2 标准 I/O	292
11.2.1 打开文件	292
11.2.2 模式	293
11.2.3 关闭文件	293
11.2.4 文件结束检测函数	295
11.3 文件 I/O	295
11.3.1 格式化 I/O	295
11.3.2 字符的 I/O	298
11.3.3 字符串的 I/O	302
11.3.4 数据块的 I/O	304
11.4 文件的随机读写	306
11.5 文件应用案例	307
本章小结	310
附录 I 经典习题及答案	311
题型一 单项选择题	311
题型二 填空题	332



题型三 程序改错题	340
题型四 程序填空题	385
题型五 程序设计题	418
附录 II 常用运算符优先级和结合性	446
附录 III 常用库函数	447
附录 IV 常用字符 ASCII 码值对照表	450
参考文献	451

第1章 概述

当今世界计算机已无处不在，从生产、生活到军事、科研，各行各业都离不开计算机；从学习、工作到休闲娱乐，计算机已以各种形式成为人们不可缺少的“伴侣”。当你使用计算机进行数据计算或者游戏娱乐的时候，是否想过这些功能是如何实现的呢？显然是通过运行程序实现的，若要处理数据，就要运行数据处理程序；若要玩游戏，就要运行游戏程序。那么，这些程序是如何编写出来的呢？是否想自己动手编写程序让计算机运行呢？让我们开始神奇的编程之旅吧！

本章主要介绍 C 程序的基本概念和运行方法，主要内容如下：

- 基本概念；
- C 程序基本结构；
- C 程序运行方法。

1.1 基本概念

1. 程序

在编写程序之前，首先要知道什么是程序。程序（Program）是为了实现特定任务而用计算机语言编写的命令序列的集合。

2. 计算机语言

编程需要使用计算机语言。计算机语言是人与计算机进行交流的工具，主要分为三大类：机器语言、汇编语言和高级语言。

1) 机器语言：完全由“0”和“1”构成机器指令，计算机可直接执行，但含义不清，难学、难记、难理解，同一条指令，不同型号的机器使用的“0”“1”串可能不同。

2) 汇编语言：为了方便理解和记忆，减轻编程人员的痛苦和负担，汇编语言出现了。汇编语言用一些含义相对清晰的英文字母、符号来替代相应功能的二进制串指令，如 ADD 代表相加。汇编语言程序不能直接运行，需要由翻译程序将其转换成机器语言程序才能够运行。汇编语言依旧依赖计算机硬件，不同型号的硬件其汇编指令可能不同。这种针对特定硬件的汇编指令，执行效率会比较高，但可移植性不好，因为更换硬件可能要更改程序甚至重新编程。

3) 高级语言：汇编语言程序的可读性和可移植性都不够好，人们期待使用更接近人类自然语言习惯、在不同型号机器上通用的编程语言，因而高级语言应运而生。高级语言是和人类自然语言（英语）较为接近的编程语言，具有良好的可读性，不依赖具体

的计算机型号和类型，通用性较强，是目前大多数编程者的选择。高级语言有很多种，如 C、C++、Java、C#等。高级语言程序也不能直接运行，需要借助翻译程序转换为机器语言程序。高级语言的出现，使得编程不再为极少数专业人士专属。大量编程爱好者的加入，极大地推动了计算机技术的飞速发展。

3. C 语言

C 语言是一种高级语言，诞生于世界著名的贝尔实验室。自 1972 年问世以来，经过多年的发展，C 语言已成为一种功能强大、应用广泛的高级语言。它既具有高级语言的基本特征，又具有汇编语言的部分特征，因而成为经久不衰的经典编程工具。同其他语言相比，C 语言具有代码简洁紧凑、运算符及数据结构丰富、有结构化的控制语句、语法限制不很严格、可以进行位操作、生成目标代码质量高、可移植性好等特点。

与所有高级语言一样，C 语言需要经过翻译程序转换为机器语言才能够运行。C 程序的执行过程如图 1-1 所示。

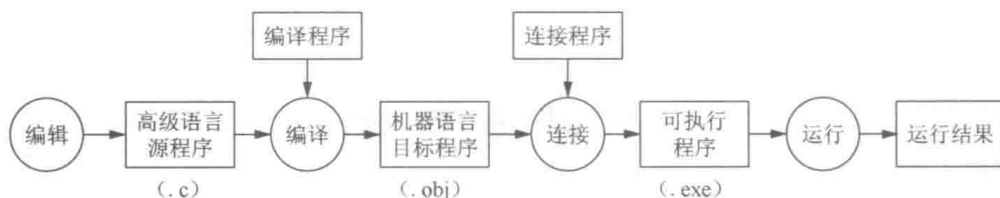


图 1-1 C 程序的执行过程

- 1) 编辑：使用编辑器录入程序代码，并保存成扩展名为.c 的源程序文件。
- 2) 编译：用编译程序将源程序翻译成机器语言程序，即生成扩展名为.obj 的目标程序文件。
- 3) 连接和运行：通过连接程序将引用的所有函数连接到目标程序中，生成扩展名为.exe 的可执行程序文件，运行即可得到运行结果。

1.2 初识 C 程序

了解了基本概念后，本节通过两个案例来初步认识 C 程序。

【案例 1-1】输出文字。

1. 案例设计目的

初步认识 C 程序。

2. 案例主要代码

```
void main()
{
    printf("我的编程之旅开始啦! \n");
}
```

3. 案例运行结果

本案例运行结果如图 1-2 所示。



图 1-2 案例 1-1 运行结果

因为 C 语言的语法和语句更接近自然语言的表述习惯，其程序可读性很好。请阅读下面的程序，猜一猜它的功能。

【案例 1-2】请猜程序功能。

1. 案例设计目的

猜测 C 程序功能，体会 C 程序的可读性。

2. 案例主要代码

```
void main()
{
    int a,b,c;
    a=300; b=18;
    c=a+b;
    printf("a+b=%d\n",c);
}
```

3. 拓展训练

观察案例 1-1、案例 1-2 两个程序，总结两者的异同。模仿案例 1-2，编写实现两个数相减（如 300 减 18）的程序。

4. 技能提高

模仿案例 1-2，编写实现 3 个数相加的程序。

1.3 C 程序的基本结构

对比案例 1-1、案例 1-2 两个程序，不难总结出程序的相同之处是都有 `main( ){}。` 模仿实现两个数相加功能的案例 1-2，可写出如下程序。

【案例 1-3】 求两个数的平均值。

1. 案例设计目的

- 1) 掌握 C 程序的基本构成。
- 2) 理解 C 程序的编写规则。

2. 案例主要代码

(注：为方便分析说明，本书后续内容的部分程序添加了行标号，运行时不需要)

```

1:  /*求两个数的平均值*/
2:  #include <stdio.h>
3:  void main()                      //主函数首部
4:  {                                // “{” 表示函数体开始
5:      int x,y,p;
6:      x=60; y=40;
7:      p=(x+y)/2;
8:      printf("%d\n",p);
9:  }                                // “}” 表示函数体结束
    
```

3. 案例运行结果

案例 1-3 的运行结果如图 1-3 所示。



```

50
Press any key to continue
    
```

图 1-3 案例 1-3 运行结果

4. 案例小结

(1) 主函数 `main()`

C 程序由函数构成，一个 C 程序可以有多个函数，但必须有且只能有一个 `main()` (主函数)；程序从 `main()` 函数开始执行。

(2) 函数

函数由函数首部和函数体组成，完整形式如下：

函数类型 函数名称 (函数参数类型 函数参数 1, 函数参数类型 函数参数 2, ...)
{ 函数体 }

本案例中 `main()` 函数的首部就是第 3 行 “`void main()`”, “`main`” 是函数名称, 函数名称前面的 “`void`” 是函数类型, 没有函数参数, 所以 “`()`” 内为空。紧跟在函数首部后的左大括号 “`{`” 表示函数体开始, 相应的右大括号 “`}`” 表示函数体结束, 即该函数结束。有关函数的详细介绍见后续的函数章节。

(3) 语句

函数体内为语句, 包括说明性语句和可执行语句, 所有语句必须以分号 “`;`” 结束。

(4) 注释

注释是不参与程序执行的内容, 可以出现在程序的任何位置。注释部分对程序执行没有任何影响, 仅供编程者查看。常用的注释方式有以下两种:

1) “`/* ..... */`” 形式: 用 “`/*`” 和 “`*/`” 括起来, 可用于行中部分注释、单行或多行注释, 如案例 1-3 中 “`/*求两个数的平均值*/`” 为注释。

2) “`//.....`” 形式: 用 “`//`” 实现单行注释 (个别编译环境不支持, 如 Win-TC), 表示该行从 “`//`” 开始的内容均为注释部分, 如案例 1-3 中的 “`//主函数首部`” 等。

(5) 书写格式

C 语言书写格式自由, 一条语句可以占一行或多行, 一行也可以有一条或多条语句。C 语言区分大小写, 多用小写字母。程序书写格式虽未规定, 但多采用按层次缩进格式。所有标点符号必须为英文格式, 否则编译程序无法识别。

以上部分内容, 读者也许暂时不能完全理解。在以后的学习过程中, 读者会不断有所体会。

1.4 运行 C 程序

前几节介绍了 3 个 C 程序, 并分析程序的运行结果。本节将介绍如何在计算机上运行 C 程序。C 程序的开发主要包括编辑、编译、连接和运行几个步骤。常用的高级语言程序开发工具集成了编辑需要的编辑器、编译需要的编译程序、连接需要的连接程序。下面简要介绍常见的 C 程序开发工具 Turbo C 2.0、VC++ 6.0 和 Win-TC。

1.4.1 Turbo C 2.0

打开 `tc2` 文件夹, 找到并运行 `TC.exe` 文件, 弹出软件版本信息窗口后按【Enter】键, 即进入 Turbo C 2.0 开发环境。其主界面包括菜单栏、编辑区、信息区、提示栏, 如图 1-4 所示。



图 1-4 Turbo C 2.0 主界面

菜单栏包括 File、Edit、Run、Compile、Project 等菜单项。Edit 菜单项的功能是激活编辑区进入编辑状态，除了此项外其他各项都有下拉菜单。在 Turbo C 2.0 环境下鼠标不可用，只支持键盘。要选择菜单项，可使用【Alt】键加上菜单项首字母，结合上下左右方向键切换菜单项。

以案例 1-1 的执行为例，在 Turbo C 2.0 中运行程序的主要步骤如下：

1. 编辑

按【Alt+F】组合键或在 File 菜单项选中的状态下按【Enter】键，在下拉菜单中通过向下箭头键选择 New 命令并按【Enter】键，光标将切换到编辑区域，即可开始编辑程序，如图 1-5 所示。如果需要打开已有 C 程序，则选择 Load 命令并按【Enter】键，输入要打开文件的路径名称后按【Enter】键即可。



图 1-5 新建文件

在编辑区域输入案例 1-1 程序代码，因为 Turbo C 2.0 不支持中文，所以需要修改汉字为英文。编辑完毕应保存程序，可使用 File→Save 命令或按快捷键【F2】。同其他软件一样，首次保存程序时会弹出保存路径窗口，输入需要保存的位置和文件名按【Enter】键，即可完成保存，如图 1-6 所示。