

计算机基础课程学习辅导丛书



C语言程序设计

——精选范例解析与习题

丛书主编 胡维华

主 编 颜 晖 王云武



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

计算机基础课程学习辅导丛书

C 语言程序设计

——精选范例解析与习题

主编 颜 晖 王云武



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计:精选范例解析与习题 / 颜晖,王云武主编.
—杭州:浙江大学出版社, 2010. 8
ISBN 978-7-308-07882-5

I. ①C… II. ①颜… ②王… III. ①C 语言—
程序设计—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 157591 号

C 语言程序设计——精选范例解析与习题

主编 颜 晖 王云武

策 划 希 言 黄娟琴

责任编辑 黄娟琴 jqhuang@zju.edu.cn

封面设计 卢 涛

出版发行 浙江大学出版社

(杭州市天目山路 148 号 邮政编码 310007)

(网址: <http://www.zjupress.com>)

排 版 杭州中大图文设计有限公司

印 刷 浙江全能印务有限公司

开 本 787mm×1092mm 1/16

印 张 14

字 数 332 千

版 印 次 2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷

书 号 ISBN 978-7-308-07882-5

定 价 25.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话 (0571)88925591

序 言

21 世纪是一个信息化的时代,计算机技术在各领域得到了越来越广泛的应用,社会要求当代大学生必须具备相当的计算机知识和应用能力。为此,各高校都不同程度地开设了一系列计算机类相关课程。由于该类课程一般课时不多,但技术性强且更新速度快,很多学生反映光靠课堂教学和一本主教材无法很好地掌握相关知识,希望有一套专门的辅助教材供课后加深理解与练习提高。为了满足同学们的这一要求,增强其自主学习和应用计算机的能力,我们邀请多位在计算机基础教学领域具有丰富经验的教师,编写了这套“计算机基础课程学习辅导丛书”。

本系列丛书包括《大学计算机基础——精选范例解析与习题》、《大学计算机基础(Linux)——精选范例解析与习题》、《C 语言程序设计——精选范例解析与习题》、《Visual Basic 程序设计——精选范例解析与习题》、《Java 语言程序设计——精选范例解析与习题》、《Visual FoxPro 程序设计——精选范例解析与习题》、《数据库技术与应用——精选范例解析与习题》、《计算机网络技术与应用——精选范例解析与习题》等。

此丛书中,每一本书都简明扼要地归纳了相应课程的知识要点与难点,详述了各精选范例的解题思路及过程,设计了大量习题,同时给出了 3~5 套模拟试卷,以及所有习题与试卷的参考答案。本书的编写凝结了长期从事计算机课程教学与计算机考试辅导教师们的宝贵经验,是一本难得的教学和自学指导用书。

丛书针对目前绝大多数高校已经开设的计算机基础课程进行设计和编写。每本书的知识范围与能力要求不是依据现有的某一套教材,而是根据教育部计算机专业教学指导委员会、教育部计算机基础课程教学指导委员会颁发的有关教学要求,综合考虑了一些省市计算机等级考试的考试大纲而确定的。丛书可作为相应课程的辅助教材(教师教学范例选取、学生自学自测的辅助用书),也可作为读者参加教育部或省市计算机等级考试的考前复习用书。希望本系列丛书的推出,能对教师的教学、学生的学习带来较大的帮助,促进高校计算机基础教学水平与质量的进一步提高。

胡维华

2009 年 5 月

前 言

程序设计是高校重要的计算机基础课程,它以编程语言为平台,介绍程序设计的思想和方法。通过该课程的学习,学生不仅要掌握高级程序设计语言的知识,更重要的是要在实践中逐步掌握程序设计的思想和方法,培养问题求解和程序设计语言的应用能力。因此,这是一门以培养学生探索创新精神和实践能力的课程。

C语言是得到广泛使用的程序设计语言之一,它既具备高级语言的特性,又具有直接操纵计算机硬件的能力,并以其丰富灵活的控制和数据结构、简洁而高效的语句表达、清晰的程序结构和良好的可移植性而拥有大量的使用者。目前,C语言被许多高校列为程序设计课程的首选语言。

本书是配合“C语言程序设计”课程而编写的教学参考书,可供该课程的教师及自学者进行理论学习、理解与实践时使用。本书简明扼要地阐述、归纳了“C语言程序设计”课程中相关知识的要点与难点,特别是配合大量的精选范例解析,将课程中的基本概念及技术要点、难点有机地结合起来并较好地进行呈现,是一本不错的教学或自学指导用书。

全书共分为10章:第1章C语言程序设计概述;第2章程序设计基础;第3章选择结构;第4章循环结构;第5章数组;第6章函数与预处理;第7章指针;第8章结构体;第9章文件;第10章综合程序设计。

每章(1~9章)又分为4节:第一节“知识要点”,对本章中的知识要点与难点作简单的归纳与扼要概括;第二节“范例解析”,列举了大量精选的实例,详尽地分析和讲解例题的分析过程和习题代码,并阐述相关的概念与涉及的知识点,非常有利于读者掌握程序设计的技术要领,加快对要点、难点内容的理解;第三节“常见错误”,列举了初学者在学习过程中容易犯错的地方,详细分析错误的原因和改正的方法,有利于读者对知识点的理解和巩固,这节也是本书的特色之一;第四节“精选习题”,精选一些具有内容延伸性与应用性的附加习题,供读者进一步思考、自测和提高,附录中提供所有习题的参考答案。在第10章,特别精选编写一些侧重于趣味性和实用性的综合应用设计范例,集中全书各章的内容,将分散的知识有机地串联起来,给出详细的分析过程和解题思路。

此外,为了方便教师组卷和读者自学、自测和总结,我们在10章内容之

后,还编制了7套内容搭配合理、符合教学考试大纲的考试模拟试卷,并在附录2中给出每套模拟试卷的参考答案。

本书由浙江大学城市学院的颜晖老师和王云武老师联合担任主编,王云武老师执笔编写了第1章、第3章、第5章、第6章、第8章、第10章,浙江大学计算学院的白洪欢老师执笔编写了第4章、第7章,浙江大学城市学院计算学院的肖艳萍老师执笔编写了第2章、第9章,全书由颜晖教授和王云武老师统稿。另外,参与本书编写工作的还有俞雪永、李晨光等。

计算机科学和技术在不断发展,计算机教学的研究和改革也从未停顿。希望在从事计算机基础教学的各位同仁的共同努力下,能不断提高C语言程序设计课程的教学质量和水平。

由于作者水平所限,对书中存在的谬误之处,敬请读者指正。作者邮箱: wangyw@zucc.edu.cn。

编 者

2010年7月

目 录

第 1 章 C 语言程序设计概述	1
1.1 知识要点	1
1.1.1 程序的性质	1
1.1.2 C 语言源程序的框架	1
1.1.3 程序的编译和执行	1
1.2 范例解析	2
1.3 常见错误	4
1.3.1 语法错误	4
1.3.2 运行时错误	4
1.3.3 逻辑错误	4
1.4 精选习题	4
第 2 章 程序设计基础	6
2.1 知识要点	6
2.1.1 C 的数据类型	6
2.1.2 常量与变量	6
2.1.3 类型转换	6
2.1.4 运算符与表达式	6
2.1.5 数据的输入和输出	7
2.2 范例解析	8
2.3 常见错误	11
2.4 精选习题	13
2.4.1 单项选择题	13
2.4.2 程序阅读题	14
2.4.3 程序设计题	15
第 3 章 选择结构	16
3.1 知识要点	16
3.1.1 if 语句	16



3.1.2	switch 语句	17
3.2	范例解析	17
3.3	常见错误	22
3.4	精选习题	23
3.4.1	单项选择题	23
3.4.2	程序阅读题	24
3.4.3	程序设计题	25
第 4 章	循环结构	26
4.1	知识要点	26
4.2	范例解析	27
4.3	常见错误	34
4.4	精选习题	37
4.4.1	单项选择题	37
4.4.2	程序阅读题	38
4.4.3	程序设计题	39
第 5 章	数 组	41
5.1	知识要点	41
5.1.1	一维数组	41
5.1.2	二维数组	41
5.1.3	数组的引用	42
5.1.4	字符数组	42
5.1.5	常用的字符串处理函数	42
5.2	范例解析	42
5.3	常见错误	48
5.4	精选习题	50
5.4.1	单项选择题	50
5.4.2	程序设计题	50
第 6 章	函数与预处理	52
6.1	知识要点	52
6.1.1	函数概述	52
6.1.2	函数定义	52
6.1.3	函数的调用及返回	53
6.1.4	变量及其作用范围	53
6.1.5	函数的作用范围	54
6.1.6	预处理	54



6.2 范例解析	55
6.3 常见错误	61
6.4 精选习题	63
6.4.1 单项选择题	63
6.4.2 程序阅读题	64
6.4.3 程序填空题	65
6.4.4 程序设计题	66
第7章 指 针	67
7.1 知识要点	67
7.2 范例解析	68
7.3 常见错误	77
7.4 精选习题	79
7.4.1 单项选择题	79
7.4.2 程序阅读题	81
7.4.3 程序设计题	82
第8章 结构体	84
8.1 知识要点	84
8.1.1 结构体类型定义	84
8.1.2 结构体变量的定义	84
8.1.3 结构体变量的引用	84
8.1.4 结构体与数组	85
8.1.5 结构体与指针	85
8.1.6 链表	85
8.1.7 自定义类型说明符	86
8.2 范例解析	86
8.3 常见错误	90
8.4 精选习题	91
8.4.1 单项选择题	91
8.4.2 程序阅读题	91
8.4.3 程序设计题	92
第9章 文 件	93
9.1 知识要点	93
9.1.1 文件的基本概念	93
9.1.2 常用的文件操作函数	93



9.2 范例解析	94
9.3 常见错误	98
9.4 精选习题	99
第 10 章 综合程序设计	100
附录 1 模拟试卷	112
模拟试卷 1	112
模拟试卷 2	120
模拟试卷 3	128
模拟试卷 4	135
模拟试卷 5	143
模拟试卷 6	149
模拟试卷 7	155
附录 2 参考答案	164



第 1 章

C 语言程序设计概述

1.1 知识要点

C 语言是一种结构化的程序设计语言。它集高级语言和低级语言的功能于一体,既适用于系统软件的开发,又适合于应用软件的编写。由于 C 语言强大的功能和各方面的特点,因此广泛应用于专业的程序设计领域。

1.1.1 程序的性质

- (1)指令顺序地执行,一般情况下,从第一条指令开始,依次执行每一条指令直至结束;
- (2)程序运行至少有一个结果,常常以打印或者显示出的符号组成输出的形式出现;
- (3)程序总是操作于某些对象(数据);
- (4)有时指令前面有一个操作对象的说明(说明数据的属性);
- (5)有时指令要求由执行者作出判定;
- (6)一条或者一组指令可能需要执行一次以上。

1.1.2 C 语言源程序的框架

一个完整的 C 语言源程序一般是由一个或多个函数组成的。其中必须有一个且只能有一个名为 `main()` 的主函数,可以在程序的任何合法位置。C 语言程序可以有若干个自定义函数。整个程序从 `main()` 函数开始执行,执行时 `main()` 函数可以调用其他标准库函数或自定义函数。函数名后必须有小括号,函数体放在大括号内。

1.1.3 程序的编译和执行

编辑完成的 C 语言源程序必须经过编译、链接生成可执行文件后,才能在操作系统的控制下被执行。调试程序是一项深入细致的工作,需要在实践的基础上逐步积累经验,提高自身程序调试的能力。



1.2 范例解析

【例 1-1】 C 程序是_____程序设计语言。

- A. 低级及结构化
- B. 高级及面向对象
- C. 高级及结构化
- D. 高级及非结构化

【分析】 目前使用的计算机程序设计语言有上百种,有些语言是面向机器的,如二进制语言,而多数语言是面向问题的,被称为计算机的“高级语言”,如 C 语言和面向对象的 C++ 语言。

程序设计一般可分为非结构化程序设计、结构化程序设计和面向对象的程序设计。C 语言是结构化程序设计语言,它由顺序、选择和循环 3 种结构组成。

【答案】 C

【例 1-2】 组成 C 程序的基本单位是_____。

- A. 表达式与语句
- B. 若干文件
- C. 函数
- D. main 函数

【分析】 C 语言是结构化程序设计语言。C 程序由一个或多个文件组成,而一个文件可由一个或多个函数组成。函数是组成 C 程序的基本单位,函数由语句构成。C 程序的执行总是从 main 函数开始的。

【答案】 C

【例 1-3】 C 程序中能让机器直接执行的文件是_____。

- A. 编辑后的.c 源文件
- B. 通过编译的.obj 文件
- C. 通过连接的.exe 文件
- D. 正在编辑的源文件存盘为后缀为.exe 的文件

【分析】 我们编辑的是源程序,其扩展名为.c 或.cpp,它是文本文件,机器不能直接执行。C 语言源程序经过编译(Compile)生成一个后缀为.obj 的二进制文件后,再由连接程序(link)把.obj 文件与 C 语言提供的各种库函数连接起来生成一个.exe 文件,它就是可执行文件,机器可以直接执行。可执行文件是通过编译和连接生成的,并非通过改变扩展名而得到。

【答案】 C

【例 1-4】 找出下面程序的错误。

```
main
{
    int a,b;
    a = 10;
    b = a + 50;
    print ("%d,%d\n",A,B);
```

【分析】 程序存在 5 个错误:一是主函数名后面的圆括号忘记写了;二是声明语句



int a;b;中的两个变量之间应该用逗号隔开;三是输出函数 print 书写错误,正确应该是 printf;四是输出变量的值时,把变量 a 和 b 写成了大写字母 A 和 B,在 C 语言中,应该区分大小写字母;五是最后没有右花括号,使函数体的花括号不匹配。

【例 1-5】 请编写一个程序,在屏幕上显示一个文字菜单模样的图案,如图 1.1 所示。

```
*****
          1. 输入数据      2. 修改数据
          3. 查询数据      4. 打印数据
*****
```

图 1.1 文字菜单模样的图案

【分析】 这是我们所涉及的第一个程序设计题,它是一个打印输出格式问题。

【答案】

```
#include<stdio.h>
void main()
{
    printf("*****\n");
    printf("1. 输入数据      2. 修改数据\n");
    printf("3. 查询数据      4. 打印数据\n");
    printf("*****\n");
}
```

【例 1-6】 判断从键盘输入的一个整数是奇数还是偶数。

【分析】 这是一个典型的分支结构。判断奇数还是偶数的常用方法是:将该整数对 2 取余数,余数为 0 为偶数,为 1 即为奇数(如图 1.2 所示)。

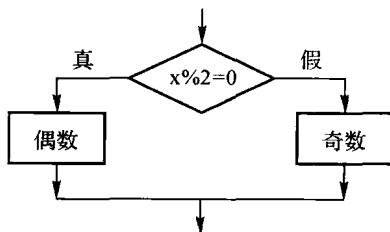


图 1.2 判断奇数和偶数的流程

【答案】

```
#include<stdio.h>
void main()
{
    int x;
    scanf("%d", &x);
    if (x % 2 == 0)
```



```
printf ("%d 为偶数\n", x);  
else  
    printf ("%d 为奇数\n", x);
```

1.3 常见错误

程序设计初学者会发现,一个程序很少能够一次性运行通过。实际上,在程序设计过程中,错误是非常常见的,改正错误的过程称为调试。

当编译系统检测到一个错误时,就会显示一条出错信息,表示程序有错误及可能的错误原因。遗憾的是,这些出错信息有时很难被理解甚至使人误解。随着经验的积累,我们将慢慢地成为查找并纠正错误的高手。

下面分别介绍三种类型的错误:语法错误、运行时错误和逻辑错误。

1.3.1 语法错误

编程时若违反了C语言的语法规则,就会产生语法错误,编译系统在编译源程序时,就能够发现这种类型的错误。若程序源代码含有语法错误,就不能被正确地编译生成可执行文件,程序也就不能运行。常见的错误:例如初学者没有定义变量而直接使用变量、语句后面缺少“;”等。需要知道的是,在程序编译时,程序中的一处错误可能会导致出现多条错误信息。比较理想的纠错过程是首先集中修改声明语句中的错误,然后在修改其他错误之前,重新编译程序。在初学过程中,常常是一旦声明语句正确,其他很多错误信息也就随之消失。

1.3.2 运行时错误

运行时错误就是程序在运行期间发生了超出程序允许程度的故障,使程序无法正常运行。当程序指挥计算机去执行一个非法操作时,比如5/0就会发生这种类型的错误。这种类型的错误发生时,计算机将会停止程序的运行,并显示诊断信息,指出错误的所在行。

1.3.3 逻辑错误

若程序的算法错误就会产生逻辑错误。因为这类错误通常不会产生运行时的错误,也不会显示出错信息,因此很难检测到。只能通过对程序的全面调试,将程序输出与计算结果进行比较,才有可能找到逻辑错误。

1.4 精选习题

1. 用高级语言编写的程序称之为 ()
- A. 源程序 B. 目标程序 C. 汇编程序 D. 可执行文件



2. C 语言总是从_____函数开始执行 ()

A. 处于最前的 B. main C. 处于最后的 D. 随机选一个

3. 对 C 语言来说,下列标识符中哪些是合法的,哪些是不合法的?

total, _debug, Large&Tall, Counter1, begin_

4. 画出流程图,求 1~100 中能被 6 整除的所有整数的和。

5. 对于给定的整数 $n(n>1)$,请设计一个流程图判别 n 是否为一个素数(只能被 1 和自己整除的整数),并分析该流程图中哪些是顺序结构,哪些是分支结构与循环结构(如果有的话)。



第 2 章

程序设计基础

2.1 知识要点

程序设计基础主要包括 C 语言数据类型、常量与变量、运算符、表达式、printf 和 scanf 语句等内容。

2.1.1 C 的数据类型

数据是程序处理的对象。数据类型决定了数据对象的存储形式、取值范围和能进行的运算。在 C 语言中,数据类型一般包括基本数据类型、构造数据类型、指针类型和空类型 4 大类。

2.1.2 常量与变量

在 C 语言中,数据有常量和变量之分。在程序运行过程中,其值不能改变的量称为常量;其值可以改变的量称为变量。

1. 常量

在基本数据类型中,常量可分为:整型常量、浮点型常量和字符型常量。

2. 变量

变量用于存储数据,在程序中使用变量之前必须先将其声明为某种数据类型。变量定义的形式为:

数据类型名 变量名表;

2.1.3 类型转换

当一个表达式中有不同数据类型的数据参加运算时,就要进行类型转换,包括自动转换和强制转换。

强制类型转换的一般形式为:

(类型名)(表达式)

2.1.4 运算符与表达式

表达式就是由运算符和运算对象(操作数)组成的有意义的运算式子,它的值和类型



由参加运算的运算符和运算对象决定,其中运算符就是具有运算功能的符号,运算对象指常量、变量、函数等表达式。表达式按其所含运算符和运算对象的不同,可分为算术表达式、赋值表达式、逗号表达式、关系表达式、逻辑表达式等。

1. 算术运算符与算术表达式

算术运算符用于各类数值运算,包括 $+$ 、 $-$ 、 $*$ 、 $/$ 、 $\%$ 、 $++$ 、 $--$ 。

2. 赋值运算符及赋值表达式

赋值表达式的一般形式为:

变量 = 表达式;

其作用是把“=”右边的表达式值赋给“=”左边的变量。

C语言中,在赋值运算符“=”前加上算术运算符($+$ 、 $-$ 、 $*$ 、 $/$ 、 $\%$)就构成复合赋值运算符,例如: $+=$ 、 $-=$ 、 $*=$ 、 $/=$ 、 $\%=$ 等。

3. 关系运算符与关系表达式

关系运算符也称为比较运算符,包括 $>$ 、 $<$ 、 $==$ 、 $>=$ 、 $<=$ 、 $!=$ 6种,均为双目运算符,用于比较两边的表达式是否满足条件,运算结果为1代表真,0代表假。

4. 逻辑运算符与逻辑表达式

常用的逻辑运算符有3种:与($\&$)、或($\|$)、非(!)。逻辑运算对象可以是关系表达式或逻辑量,用整数1代表真,0代表假。

5. 条件运算符与条件表达式

条件表达式的一般形式为:

表达式1? 表达式2: 表达式3;

其运算过程是:先计算表达式1的值,如果它的值为非0(真),那么就将表达式2的值作为条件表达式的值;否则,将表达式3的值作为条件表达式的值。

6. 逗号运算符与逗号表达式

逗号表达式的一般形式为:

表达式1, 表达式2, ..., 表达式 n

它的值为表达式 n 的值。

7. 运算符的优先级和结合性

C语言的运算符具有优先级。多种运算符混合运算的优先级关系为:先算术运算符,次关系运算符,后逻辑运算符。

2.1.5 数据的输入和输出

C语言中,数据的输入和输出是通过调用函数来实现的。在调用输入输出函数时,在源程序开始时要使用编译预处理命令`#include <stdio.h>`。

