

普通高校系列教材·信息技术

# C语言程序设计

普通高校系列教材(信息技术)编委会组编

蒋劲柏 编

## 学习参考



2C

南京大学出版社

普通高校系列教材·信息技术

# 《C 语言程序设计》学习参考

蒋劲柏 编

南京大学出版社

内  
容  
简  
介

本书是《C 语言程序设计》的配套辅导材料,全书共分为五个部分:自学方法,习题解答,典型例题,模拟测试,实验指导。

习题解答和典型例题部分主要以习题讲解的形式针对教材中的不同知识点进行阐述,可以帮助读者提高解题能力。这两部分是本书的重点。

通过使用本书,读者可以对教材有进一步的掌握,同时也能提高自己的动手能力。

图书在版编目(CIP)数据

《C 语言程序设计》学习参考/蒋劲柏编.—南京:南京大学出版社,2003.5

ISBN 7-305-02594-1

I . C . . . II . 蒋 . . . III . C 语言-程序设计-高等学校-教学参考资料 IV . TP312

中国版本图书馆 CIP 数据核字(2003)第 038036 号

书 名 《C 语言程序设计》学习参考  
编 者 蒋劲柏  
出版发行 南京大学出版社  
社 址 南京市汉口路 22 号 邮编 210093  
电 话 025-3596923 025-3592317 传真 025-3303347  
网 址 www.njupress.com  
电子函件 njupress1@public1.ptt.js.cn  
经 销 全国新华书店  
印 刷 合肥学苑印刷厂印刷  
开 本 787×1092 1/16 印张 9.5 字数 228 千  
版 次 2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷  
书 号 ISBN 7-305-02594-1/TP·94  
定 价 14.00 元

\* 版权所有,侵权必究

\* 凡购买南大版图书,如有印装质量问题,请与所购  
图书销售部门联系调换

## 出版前言

近些年来我国的高等教育事业有了长足的发展,高校招生人数年年递增,越来越多的年轻人有机会接受正规的高等教育。这一举措无疑对我国的社会进步和经济发展有着重要的意义。但是人们也深刻地认识到,高等教育质量的好坏是一个不容忽视的关键性问题,而保证教育质量的一个重要环节就是抓好教材建设。但是教材内容陈旧、教学手段落后的现象一直存在着。尤其像计算机技术这样的新兴领域发展迅猛,知识更新日新月异,教学内容落后于新技术新知识的矛盾显得尤为突出。基于上述两方面考虑,在南京大学出版社的鼎力相助下,一个以组编高校信息、电子类专业教材为主要任务的教材编委会成立了。

针对我国高等教育的现状和信息、电子技术的发展趋势,编委会组织部分高校的专家教授进行了深入的专题研讨。大家一致认为,在当前情况下组编一套紧跟新技术发展、符合高校教学需要、满足大学生求知欲望的系列教材势在必行,这将有助于规范教学体系、更新教学内容、把握教学质量,培养合格人才。专家们还对教材的结构、内容、体例及配套服务等提出了具体要求。为了能使这套教材逐步完善,并促进全国各地高校教学质量的提高,编委会决定在教材之外认真做好三件事:第一,为每本教材配备一本供学生使用的学习参考书,其主体内容为学习方法指导、习题分析与解答、典型题解或课程设计、模拟测试卷及解答、实验指导书;第二,定期对教材内容进行修订,及时补充新技术新知识,并根据具体情况组编新的教材;第三,有计划地组织各地高校教师进行教学交流与研讨,通过这种途径来提高偏远地区的师资水平。我们相信,通过各方面的大力支持和大家的不懈努力,这套教材会逐步被广大师生所接受,并在使用过程中得到完善、充实。

大家都知道,组编这样一套系列教材是个牵涉面很广的大工程。这个工程不仅在起步阶段需要得到各级教育主管部门、各高等院校、出版社的大力支持和协助,而且在使用过程中也离不开各位专家、教授、学生的热心呵护和指导。因此,殷切期待所有的能人志士关心我们、帮助我们,向我们提出好的建议或意见,为我们指出教材中的不足之处。

最后,感谢所有为本套系列教材出版付出辛勤劳动的同志们。

普通高校教材(信息技术)编委会

2003年5月

## 编者的话

C语言是一门易学难精的语言,语法比较简单,容易入门;但要说学好C语言可不是件容易的事,C语言中的指针、数组等概念更是如此。为了帮助大家更好地学习C语言,提高大家的解题能力和上机实践能力,我们特编写了此书。

本书是《C语言程序设计》的配套辅导教材,全书共分为五个部分:自学方法,习题解答,典型例题,模拟测试,实验指导。具体编写内容为:

自学方法部分包括三部分内容:一是针对教材,整理出一条有关本课程的学习主线条;二是结合课程特点总结出一些学习C语言的方法和思路;三是对C语言中一些重点难点内容进行专门的讲述,以求加深读者的印象。

习题解答部分主要是对教材每章后的习题进行解答分析。

典型题解部分精选了一些有代表性的题目,通过分析解答,可以让读者对C语言的一些重点难点有更深刻的理解。

模拟测试部分给出了两套模拟试卷,读者可以自己进行测试。

实验指导部分给出了几个需要进行上机实践的练习,并进行了指导,以求在掌握基本知识的同时提高读者的实际动手能力。

结合使用本书,读者可以对教材有进一步的掌握,同时提高自己的动手能力。

编者

2003年5月

# 普通高校系列教材(信息技术)编委会

主 任:孙钟秀 中国科学院院士

副主任:张福炎 南京大学教授

陈国良 中国科学技术大学教授

钱洲胜 中国计算机函授学院教授

## 委 员(按姓氏笔画排序):

|     |              |      |         |
|-----|--------------|------|---------|
| 于学锋 | 中国计算机函授学院    | 王佩珠  | 西安交通大学  |
| 王文兰 | 桂林电子工业学院     | 王蔚韬  | 重庆建筑大学  |
| 伍良富 | 成都电子科技大学     | 成松林  | 东南大学    |
| 刘存书 | 郑州信息工程大学     | 朱大奇  | 安徽工业大学  |
| 朱宝长 | 西安电子科技大学     | 孙德文  | 上海交通大学  |
| 杜象元 | 上海交通大学       | 李茂青  | 厦门大学    |
| 李学干 | 西安电子科技大学     | 杨来利  | 兰州大学    |
| 何淑兰 | 北京科技大学       | 张凤祥  | 华中科技大学  |
| 张绍林 | 河北行政学院       | 张维勇  | 合肥工业大学  |
| 张民坤 | 云南工业大学       | 张景书  | 哈尔滨工程大学 |
| 赵良全 | 新疆大学         | 皇甫正贤 | 东南大学    |
| 洪志全 | 成都理工学院       | 姚君遗  | 合肥工业大学  |
| 高 平 | 浙江大学         | 陶世群  | 山西大学    |
| 曹翊旺 | 湖南省计算机高等专科学校 | 梁文康  | 山东大学    |
| 韩国强 | 华南理工大学       | 舒 洪  | 南昌大学    |
| 葛 燕 | 中国科学技术大学     | 解世耀  | 辽宁大学    |
| 谭耀铭 | 南京大学         | 黎庆国  | 合肥工业大学  |

# 目 录

|                |       |
|----------------|-------|
| 第一部分 学习方法指导    | (1)   |
| 一、教学目的         | (1)   |
| 二、学习方法指导       | (2)   |
| 三、课程的基本内容      | (3)   |
| 四、重点、难点解析      | (9)   |
| 第二部分 习题解答      | (25)  |
| 第1章            | (25)  |
| 第2章            | (26)  |
| 第3章            | (30)  |
| 第4章            | (32)  |
| 第5章            | (35)  |
| 第6章            | (36)  |
| 第7章            | (38)  |
| 第8章            | (41)  |
| 第9章            | (43)  |
| 第10章           | (44)  |
| 第三部分 典型例题分析与解答 | (46)  |
| 第四部分 模拟测试      | (90)  |
| 模拟试卷(一)        | (90)  |
| 模拟试卷(一)参考答案    | (98)  |
| 模拟试卷(二)        | (105) |
| 模拟试卷(二)参考答案    | (114) |
| 第五部分 实验指导      | (120) |
| 实验一 顺序和分支程序设计  | (120) |
| 实验二 数组和循环程序设计  | (125) |
| 实验三 指针和函数的应用   | (129) |
| 实验四 结构型的应用     | (134) |
| 实验五 文件的应用      | (139) |

# 第一部分 学习方法指导

## 一、教学目的

### 1. 课程的性质与任务

C 语言是当前最受人们欢迎的一种程序设计语言之一,不管是在以前的 DOS 系统,还是现在的 Windows 系统,都有 C 语言家族的身影。C 语言具有功能强、语言简洁、结构紧凑高效等特点,这是它得到广泛应用的原因。

Turbo C 2.0 在 C 语言的发展史上具有非常重要的地位:在 DOS 系统上,它是一个比较成熟的产品,提供了比较合理的结构化程序设计方法;在 Windows 系统上,当前流行的 Visual C 等语言继承了 Turbo C 的一些优良特性,比如说指针系统等。所以学习本课程,一方面的目的是培养读者良好的编程风格和一定的程序设计能力,解决实际工作中遇到的各种问题,另一方面也可以为以后学习可视化的 C 语言打下一定的基础。

C 语言也是学好其他一些计算机专业课程的基础。在计算机专业教育中占有重要地位的数据结构课程以及操作系统等课程中,需要使用到 C 语言的知识。

### 2. 课程要求

C 语言本身语法并不复杂,入门比较容易,但是要真正掌握 C 语言的精髓,充分发挥它的潜力,做到应用自如却并非易事,这就要求深入领会课程内容,通过“阅读—体会—练习—上机”的步骤认真学习各个知识点,对一些重点难点,这个步骤更要反复进行。

教材中提供了大量的实例,认真阅读这些实例也将有助于 C 语言的学习。

具体来说,学习本课程需要掌握以下内容:

- 深入掌握 C 语言程序组织结构,领会模块化结构的思想;
- 深入掌握 C 语言的各种数据类型(基本类型、构造类型、指针类型、空类型);
- 熟练掌握 C 语言的基本输入输出函数;
- 掌握 C 程序设计的三种基本结构:顺序、选择和循环结构;
- 掌握 C 语言中常用的库函数;对不常用的库函数,要求会查表引用;
- 深入掌握函数定义及调用规则,特别是对函数调用数据传递方式中的数据复制方式和地址传递方式这两种不同机制,应深入理解内涵;
- 了解并掌握 C 语言的模块化设计方法和分别编译技术;
- 掌握指针类型的概念和使用规则;
- 掌握单向链表的建立、输出、插入与删除等算法及其程序设计;
- 掌握 C 语言的文件操作。



## 二、学习方法指导

### 1. 循序渐进, 不急不躁

C 语言的内容很丰富, 有的部分涉及到的知识较多, 如对于指针的学习, 就必须同时具备一定的硬件知识和软件知识才可以学好, 更有一些难点必须通过长期的实践才能熟练掌握。因此对于初学者来说, 一开始学习时不要强求面面俱到, 否则必然会顾此失彼, 反而抓不住主要矛盾, 不必在每一个细节上过于死抠, 而应当把主要精力放在最基本、最常用的那些部分, 待有一定基础后再注意一些非主要的细节或研究更深层次的问题。循序渐进, 不急不躁, 是学习 C 语言时需要注意的第一个问题。

### 2. 整体把握, 勤于复习

初学 C 语言时, 可能会遇到有些问题理解不透, 请不要气馁, 鼓足勇气向后面学习, 可能学完后面的章节知识, 再看看前面的问题, 会发现问题迎刃而解。学习 C 语言始终要记住“曙光在前头”和“千金难买回头望”, “千金难买回头望”是学习知识的重要方法, 就是说, 学习后面的知识, 不要忘了回头弄清遗留下的问题和加深理解前面的知识, 这是学习的人最不易做到的, 然而却又是最重要的。最典型的是关于结构化程序设计思想, 一开始我们就强调了这种方法, 这时也许你不能充分体会, 但是学到函数时, 再回过头来仔细体会, 就会对结构化程序设计的概念有更深层次的理解。学习 C 语言必须要经过几个反复, 才能前后贯穿, 积累应该掌握的 C 知识。即使学完这门课程之后, 在编写大型的或复杂的程序时也会遇到一些问题, 那时还要进一步学习深入的、提高的部分(例如利用 C 绘图或进行过程控制等), 查阅更精深的参考书。本书只是提供了一把打开 C 语言程序设计知识宝库的钥匙, 至于打开宝库后可以挖掘到什么样的宝贝, 则要看个人的努力了。

### 3. 深刻理解知识点

C 语言中的知识点较多, 对这些知识点要深刻理解。

C 语言中的有些知识点具有一定的迷惑性, 学习时一定要认真体会。例如, 对变量类型的理解, 既要考虑到变量的数据类型, 又要考虑变量的存储类型, 还要考虑变量定义的位置, 哪一方面都不能存在理解不清的问题, 只有三方面都认识地得较深刻, 才可以很好地理解某个具体变量的特性, 进而使用时才不会出错。

C 语言中还有一些知识点是具有一定联系的, 对这种联系要有清楚的认识, 既要知道相同的地方, 又要了解不同之处。如 C 语言中的指针和数组是密切相关的, 在很多场合下可以互换使用, 但两者仍有区别, 简单地讲就是: 数组名是一个指针常量, 它的值是不变的, 而指针变量的值可以变。

### 4. 从语句上把握各个知识点

对于 C 语言中的知识点, 如果一个一个地去死记硬背, 零散地学习, 是有一定困难的, 因此需要找到一条线索将这些零散的知识点串起来, 才可以收到事半功倍的效果。语句就

可以作为这样的线索。一门计算机语言,最终的目的是使用语句编写出程序,所有的知识点都是围绕这个目的而展开的,如果某个知识点不能在语句中得到使用,那么这个知识点就是多余的,完全可以不去学习。实际上也不存在这种知识点。因此,一开始学习时,可以把大部分的精力放在语句的掌握上,研究语句的结构形式、使用规则、注意事项等,随着编写程序的越来越多,越来越复杂,自然而然地会掌握各个知识点的基本内容。当然,对于一些复杂的知识点以及一些很少用到的知识点,使用这种方法可能效果不大,此时就需要专门的深入研究了。

5. 注重实践

实践出真知,这句话在 C 语言的学习中更是得到深刻的体现。C 语言有很多知识点,光看书本来理解很复杂,实际上机操作一下,可能会很简单。更有一些知识点,可能在书本上根本没有提到,必须在实践中摸索。

实践,也就是上机操作,在 C 语言的学习中占有很重要的地位,一定要花足够的时间用于上机!

三、课程的基本内容

一门课程中会有很多的知识点,掌握了这些知识点,也就意味着对课程的掌握。为了帮助读者更好地掌握 C 语言,下面按章节顺序列出 C 语言中需要掌握的各个知识点,对需要重点掌握的知识点特别指出。

第 1 章 概 述

| 章节号 | 知 识 点                                        | 重点、难点      |
|-----|----------------------------------------------|------------|
| 1.1 | C 语言的起源和发展<br>C 语言的标准化                       |            |
| 1.2 | C 语言的特点                                      | 结构化程序设计的概念 |
| 1.3 | C 语言程序设计概貌<br>C 语言程序设计本质<br>C 语言程序的书写格式      | 程序设计的本质    |
| 1.4 | C 语言程序上机环境的安装<br>C 语言程序上机环境的配置<br>C 语言集成开发环境 | C 语言集成开发环境 |

## 第2章 词法、数据和表达式

| 章节号 | 知 识 点                                                                                                                      | 重点、难点                                         |
|-----|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| 2.1 | C 语言的字符集<br>转义字符的概念<br>保留字<br>标识符的命名                                                                                       | 转移字符的概念<br>标识符的命名                             |
| 2.2 | 数据类型的概念<br>C 语言的数据类型<br>数据类型的长度                                                                                            | 数据类型的概念                                       |
| 2.3 | 常量的概念<br>C 语言中的常量<br>符号常量                                                                                                  |                                               |
| 2.4 | 变量的概念<br>变量的数据类型和定义<br>变量的类型转换规则<br>变量的存储类型和作用域<br>变量的作用域<br>变量的初始化                                                        | 变量的数据类型<br>变量的类型转换规则<br>变量的存储类型和作用域<br>变量的作用域 |
| 2.5 | 各种运算符的运算规则<br>运算符优先级和结合性<br>算术运算符<br>关系运算符<br>逻辑运算符<br>增 1, 减 1 运算符<br>赋值运算符<br>算术自反运算符<br>逗号运算符<br>条件运算符<br>长度运算符<br>位运算符 | 运算符优先级和结合性                                    |
| 2.6 | 表达式的概念<br>表达式的一般构成规则<br>算术表达式<br>关系表达式<br>逻辑表达式<br>条件表达式<br>赋值表达式<br>逗号表达式                                                 | 表达式的一般构成规则                                    |

### 第3章 简单C语言程序设计

| 章节号       | 知 识 点                                                  | 重点、难点                         |
|-----------|--------------------------------------------------------|-------------------------------|
| 3.1 ~ 3.5 | C 语言中的基本语句<br>表达式语句<br>空语句<br>复合语句<br>函数调用语句<br>流程控制语句 | 各种基本语句的特点和功能                  |
| 3.6       | 标准输出函数 printf<br>标准输入函数 scanf                          | 标准输出函数 printf<br>标准输入函数 scanf |

### 第4章 程序流程的控制

| 章节号 | 知 识 点                                                                                                                 | 重点、难点                    |
|-----|-----------------------------------------------------------------------------------------------------------------------|--------------------------|
| 4.1 | 结构化程序设计中的三种结构：<br>顺序结构<br>选择结构<br>循环结构                                                                                | 三种程序结构的特点                |
| 4.2 | 选择结构流程控制语句 if<br>选择结构流程控制语句 switch                                                                                    | if 语句的构成<br>switch 语句的使用 |
| 4.3 | 循环结构流程控制语句：<br>while 语句<br>for 语句<br>do...while 语句<br>三种循环语句的嵌套<br>改变循环结构的三种语句：<br>break 语句<br>continue 语句<br>goto 语句 | 三种循环语句的特点                |

## 第5章 数 组

| 章节号 | 知 识 点                                                                  | 重点、难点                  |
|-----|------------------------------------------------------------------------|------------------------|
| 5.1 | 数组的概念<br>一维数组的定义与初始化<br>一维数组的引用<br>一维数组的具体应用                           | 一维数组的建立与应用             |
| 5.2 | 字符数组的概念<br>字符数组的定义与初始化<br>字符数组的引用<br>字符数组与字符串的关系<br>字符串函数<br>字符数组的具体应用 | 字符数组与字符串的关系<br>字符数组的应用 |
| 5.3 | 多维数组的概念<br>多维数组的定义与初始化<br>多维数组的引用<br>多维数组的具体应用                         | 多维数组的应用                |

## 第6章 指 针

| 章节号 | 知 识 点                                                         | 重点、难点                  |
|-----|---------------------------------------------------------------|------------------------|
| 6.1 | 指针的概念<br>指针与指针变量的关系<br>指针变量的定义和初始化<br>指针运算符 * 和 &<br>动态内存分配函数 | 指针与指针变量的关系<br>指针运算符的使用 |
| 6.2 | 指针的运算：<br>赋值运算<br>加减算术运算<br>两个指针变量之间的运算                       | 指针运算的实际意义              |
| 6.3 | 指针与数组的关系<br>指向多维数组的指针<br>字符数组与指针                              | 指针与数组的关系               |

| 章节号 | 知 识 点                             | 重点、难点   |
|-----|-----------------------------------|---------|
| 6.4 | 指针数组的概念<br>指针数组的定义与初始化<br>指针数组的使用 | 指针数组的使用 |
| 6.5 | 多级指针的概念<br>多级指针的定义与初始化<br>多级指针的应用 | 多级指针的应用 |

## 第 7 章 函 数

| 章节号 | 知 识 点                                                                 | 重点、难点                   |
|-----|-----------------------------------------------------------------------|-------------------------|
| 7.1 | 函数的概念<br>函数的分类<br>函数的定义<br>函数的存储类型和数据类型<br>函数的调用                      | 函数的存储类型和数据类型            |
| 7.2 | 函数间的数据传递方式<br>行参与实参结合时的两种方式：<br>数据复制方式<br>地址传递方式<br>函数的返回值<br>全局变量的使用 | 行参与实参的结合<br>全局变量的使用     |
| 7.3 | 几种特定数据类型的函数参数<br>命令行参数的使用                                             | 命令行参数的使用                |
| 7.4 | 变量的作用域：<br>局部变量的概念<br>全局变量的概念<br>变量的生存期：<br>动态变量的概念<br>静态变量的概念        | C 语言对内存空间的使用<br>各种变量的特性 |
| 7.5 | 指向函数的指针                                                               | 指向函数的指针                 |
| 7.6 | 标准库函数的使用                                                              | 标准库函数的使用                |
| 7.7 | 递归函数的概念<br>函数的递归调用                                                    | 函数的递归调用                 |

## 第8章 结构、联合与枚举

| 章节号 | 知 识 点                                                                                                    | 重点、难点              |
|-----|----------------------------------------------------------------------------------------------------------|--------------------|
| 8.1 | 结构体类型的概念<br>结构体类型变量的定义与初始化<br>结构体变量的引用<br>结构体类型与数组类型的混合类型<br>结构体类型与指针类型的混合类型<br>结构体类型在函数中的应用<br>结构体类型的应用 | 结构体的概念<br>链表的构造与使用 |
| 8.2 | 联合体类型的概念<br>联合体类型变量的定义与使用<br>联合体类型变量的应用                                                                  | 联合体的概念             |
| 8.3 | 枚举型的概念<br>枚举型变量的定义与初始化<br>枚举型变量的应用                                                                       | 枚举型的概念             |
| 8.4 | 用户自定义类型                                                                                                  |                    |

## 第9章 编译预处理与分别编译

| 章节号 | 知 识 点                           | 重点、难点       |
|-----|---------------------------------|-------------|
| 9.1 | 三种编译预处理：<br>宏定义<br>文件包含<br>条件编译 | 宏定义的使用      |
| 9.2 | 分别编译的概念<br>extern 型变量或函数        | extern 型的使用 |

## 第10章 文 件

| 章节号  | 知 识 点                                                                    | 重点、难点         |
|------|--------------------------------------------------------------------------|---------------|
| 10.1 | 文件的概念<br>文件的分类<br>文件指针                                                   | 文件的概念<br>文件指针 |
| 10.2 | 标准设备文件函数：<br>getchar()函数和 putchar()函数<br>gets()函数和 puts()函数<br>标准设备文件的换向 | 标准设备文件的换向     |

| 章节号  | 知 识 点                                                                   | 重点、难点 |
|------|-------------------------------------------------------------------------|-------|
| 10.3 | 文件的打开和关闭<br>文件的字符输入输出函数<br>文件的字符串输入输出函数<br>文件的格式化输入输出函数<br>文件的数据块输入输出函数 |       |
| 10.4 | 文件的定位操作：<br>fseek()函数<br>fseek()函数<br>rewind()函数                        |       |
| 10.5 | 文件的错误检测                                                                 |       |

#### 四、重点、难点解析

下面针对 C 语言程序设计中的重点和难点,逐个谈谈具体的学习方法。

##### 1. 如何学习 C 语言的运算符和运算顺序

C 语言的运算功能十分强大,运算种类远多于其他程序设计语言。因此,当多种不同运算组成一个运算表达式,即一个运算式中出现多种运算符时,运算的优先顺序和结合规则显得十分重要。初学者往往对此感到非常困难,觉得 C 语言学习太繁杂,其实只要对运算符进行合理分类,找出它们与数学中所学到运算之间的不同点之后,记住这些运算也就不困难了。有些运算符在理解后更会牢记心中,将来用起来得心应手,而有些可暂时放弃不记,等到用时再记不迟。

对运算符的掌握应从三个方面着手:①和操作符相关的数据类型(包括操作数和操作结果的数据类型);②运算符的优先级;③运算符的结合次序。

教材中将所有的运算符按优先级不同分类列表(表 2-1)。表中列出了 15 种优先级的运算符,从高到低优先级分别为 1~15,表中最右边一栏是结合规则,除第 13 级和第 14 级为从右至左结合外,其他都是从左至右结合,它决定同级运算符的运算顺序。

为了帮助读者更好地记住表(2-1)中的内容,列出一些规则如下:

①在双目运算符中,算术运算符的优先级高于关系运算符,关系运算符的优先级高于逻辑运算符(取反运算符!例外);②条件运算符的优先级高于赋值运算符,赋值运算符的优先级高于逗号运算符,这三种运算符的优先级都低于其他所有的运算符;③除了单目运算符、条件运算符和赋值运算符(包括自反赋值)的结合方向是自右向左以外,其他运算符均为自左向右;④须注意字位运算符 &、^、| 以及逻辑运算符 &&、|| 的优先级低于 =、!= 等关系运算符。

下面通过几个例子来说明运算符的使用:

(1)  $3 * 20 / 4 \% 10$

这个表达式中出现 3 种运算符,是同级运算符,运算顺序按从左至右结合,因此先计算



$3 * 20 = 60$ , 然后被 4 除, 结果为 15, 最后是 % (求余数) 运算, 所以表达式的最终结果为  $15 \% 10 = 5$ 。

(2)  $a = 3; b = 5; c = ( ++a ) * b; d = ( a++ ) * b$

例子中的“;”是 C 语言中的语句分隔符, 执行顺序为从左到右。第 1 条语句执行后  $a$  的值为 3; 第 2 条语句执行后  $b$  的值为 5; 第 3 条语句中有两个运算符: 前置  $++$  和  $*$ , 按表中所示顺序,  $++$  先执行,  $*$  后执行, 所以  $++a$  执行后,  $a$  的值为 4, 由于  $++$  为前置运算, 所以  $a$  以值 4 参与运算,  $c$  的值为 20; 最后执行第 4 条语句, 由于  $a++$  为后置运算, 所以  $a$  以值 4 参与运算, 使得  $d$  的值仍为 20, 而  $a$  参与运算后其值加 1, 值为 5。这个例子执行后,  $a$  的值为 5,  $b$  的值为 5,  $c$  的值为 20,  $d$  的值也是 20。

(3)  $a = 3, b = 5, b + = a, c = b * 5$

例子中的“,”是逗号运算符, 上式称为逗号表达式, 自左向右结合, 最后一个表达式的结果值就是逗号表达式的结果, 所以上面的逗号表达式结果为 40,  $a$  的值为 3,  $b$  的值为 8,  $c$  的值为 40。

## 2. 如何学习 C 语言的三种程序结构

### (1) 顺序结构

这种结构的程序比较简单, 就是按照语句的排列顺序依次执行的机制。顺序结构的执行顺序如图 1-1 所示, 自上而下, 依次执行, 因此编写程序也必须遵守这一规定, 否则程序执行的结果就不对。

例如:  $a = 3, b = 5$ , 现交换  $a, b$  的值, 正确的程序为:

$c = a;$

$a = b;$

$b = c;$

执行结果是  $a = 5, b = c = 3$  如果改变其顺序, 写成:

$a = b;$

$c = a;$

$b = c;$

则执行结果就变成  $a = b = c = 5$ , 不能达到预期的目的, 这是初学者常犯的错误。

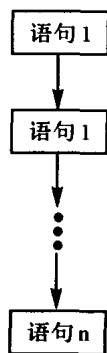


图 1-1 顺序结构执行顺序

顺序结构可以独立使用构成一个简单、完整的程序, 常见的输入、计算、输出三步曲的程序就是顺序结构。例如, 计算圆的面积, 其程序的语句顺序就是输入圆的半径  $R$ , 计算  $S = 3.14159 * R * R$ , 输出圆的面积  $S$ 。而大多数情况下顺序结构都是作为程序的一部分, 与其他结构一起构成一个复杂的程序, 如分支结构中的块体、循环结构中的循环体等。