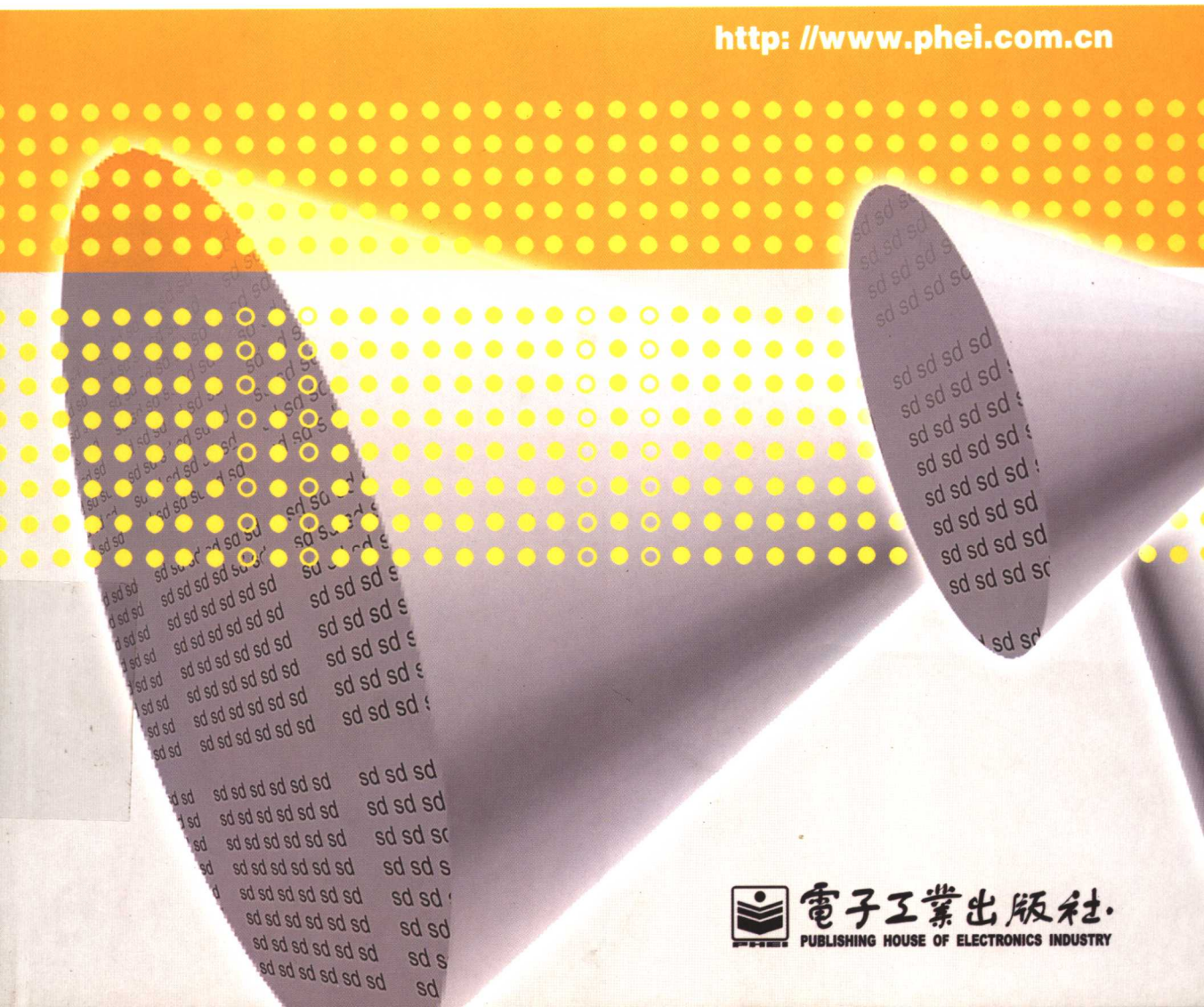
研究型教学模式系列教材

C语言程序设计

刘明军　韩玫瑰　**主编**　　魏东平　**主审**

http: //www.phei.com.cn



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

研究型教学模式系列教材

C语言程序设计

刘明军　韩玫瑰　主编
王卫峰　黄艺美　蒋　彦　潘玉奇　编
魏东平　主审

電子工業出版社
Publishing House of Elecronics Industry
北京 · BEIJING

内 容 简 介

本书以任务驱动的方式，通过实例讲授程序设计的基本概念和基本方法，把重点放在解题思路上，试图贯穿以程序编写带动语法教学的模式，引导读者掌握C语言的核心编程方法，提高应用能力。全书分为理论部分和实验部分。理论部分主要介绍C语言的基本结构、语法成分、调试方式、输入/输出语句；C语言程序的基本结构；模块分解、构造数据类型的程序设计；磁盘数据存储程序的设计方法；实用程序设计的一般方法。实验部分共设计了8个实验，通过详细的上机实验，使读者深入理解语法，培养程序设计的能力。本书每章后均附有大量习题，并提供程序源代码、课后习题指导与参考答案、多媒体电子课件。

本书可作为高等学校计算机专业和非计算机专业公共基础课的教材，也可供相关领域的工程技术人员学习参考。

图书在版编目（CIP）数据

C语言程序设计 / 刘明军，韩玫瑰主编. —北京：电子工业出版社，2007.3

研究型教学模式系列教材

ISBN 978-7-121-03893-8

I. C… II. ①刘… ②韩… III. C语言－程序设计－高等学校－教材 IV.TP312

中国版本图书馆CIP数据核字（2007）第023236号

责任编辑：王羽佳

印　　刷：北京市通州大中印刷厂

装　　订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路173信箱　邮编　100036

开　　本：787×1092　1/16　印张：20.5　字数：524.8千字

印　　次：2007年3月第1次印刷

印　　数：5 000册　　定价：28.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系电话：（010）68279077；邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

出 版 说 明

随着科教兴国战略的实施和社会信息化进程的加快，我国高等教育事业的发展迎来了新的机遇，高等学校的计算机基础教育也得到了蓬勃发展。经过多年教学实践的不断探索，我们总结出适合高等学校非计算机专业学生计算机教育的研究型教学模式。

研究型教学模式的基本形式为：精讲多练，以学生在课题研究中探索式的学习为主，以网络教学平台答疑讨论为辅，以试题库在线测验为补充的教学模式。

研究型教学模式的操作，重点突出以下三个方面：

① 加强自学和实践。课堂教学主要精讲重点内容，而不是面面俱到。在教师的指导下，学生通过自学教材，并借助网络教学平台上的多媒体课件或其他多种学习资料进行学习。同时增加上机实验教学的学时比例，充分利用上机练习掌握所学的内容。

② 以实际训练提高教学效果。在上课前给每个学生（或几个学生为一组）布置一项实际操作或软件开发课题。课题力求既结合实际，又能涵盖课程教学的内容，明确具体要求和进度。学生结合课程进度在规定时间内完成该课题后，由教师进行考核。

③ 充分重视辅助教学手段在课程教学中的作用。建设在线考试环境，学生可以随时登录进行在线测试。根据教学进度的安排，每个重要学习单元都组织学生在线测试。另外，在教学平台的辅导答疑论坛，安排专人主持，负责解答学生提出的各种问题，根据学生在答疑论坛发表见解的次数和深度，评定答疑讨论分，并计入平时成绩。

总之，研究型教学模式在重视教学过程的每个环节的同时，把调动学生学习的积极性放到了重要位置，把培养学生数字化学习的能力、自主学习的意识和培养学生创新思维的意识有机地融合到平时的教学过程之中。

为了更好地探索研究型教学模式，我们组织编写了这套系列教材，主要包括《信息技术基础》、《C 语言程序设计》、《数据库技术及应用》和《计算机网络技术与应用》等。同时开发了与本套教材相配合的网络化教学平台软件，并首先在济南大学的非计算机专业学生中试用，收到了较好的教学效果。本套教材还配有习题解答和上机指导及教学用多媒体电子课件，以利于教师备课和学生自学，请通过华信教育资源网下载，网址 www.hxedu.com.cn 或 www.huaxin.edu.cn。

非计算机专业学生的计算机教育，在教学目的、教学内容和教学方法等方面都不同于计算机专业教育。对非计算机专业的学生，计算机教育的重点应该是计算机应用能力的培养。为此，本套教材从应用出发，以应用为目的，更强调实用性，在确保概念严谨的同时，做到通俗易懂、例题丰富、便于自学。我们希望这套教材能使广大非计算机专业的学生受益，并通过研究型教学模式的应用使他们能更好地灵活掌握信息技术的相关知识和技能。

这套教材得到了济南大学教材建设委员会及各方人士的指导、支持和帮助，在此我们表示衷心的感谢。

教材中还可能存在不足之处，竭诚欢迎广大读者和同行批评指正。

研究型教学模式系列教材编写组

前　言

C 语言是一种被广泛学习、普遍使用的计算机程序设计语言。它的高级语言形式、低级语言功能的特点具有特殊的魅力，因而被大多数高等学校采用作为典型的计算机语言教学课程，也被选为计算机等级考试、全国计算机应用证书考试等多种计算机技能考试的考试内容。此外，C 语言作为一门实用且功能强大的程序设计语言，被程序设计人员广泛使用。因此，C 语言是一门十分重要的程序设计语言。

学习程序设计语言的目的在于使用该语言编写程序解决实际工作中的问题，而不仅仅是掌握该语言的语法。读者在 C 语言程序设计的学习中，很容易陷入语法的泥潭中，一开始就学习大量的语法知识，难以记忆，更难以理解，难免产生为难情绪，感到 C 语言难学，更不知道如何使用。作者在建设“省级精品课程”的同时，对课程的教学理念和教学方法进行了认真思考，集聚多年的教学经验，从而完成了本书——一本真正注重培养读者程序设计能力的教材。

本教材具有以下特色：

① 进行了基于任务驱动的实例教学法的尝试。以任务驱动的方式，通过实例讲授程序设计的基本概念、基本方法，把重点放在解题思路上。

② 从开始学习就使读者将注意力集中在所解决的问题领域，从具体实例理解 C 语言的开发特点和程序总体框架，通过实例本身学习某一类问题的解决方法和算法设计，贯穿以程序编写带动语法教学的模式。

③ 在 C 语言的环境下，针对实际问题进行分析，构建数学模型，设计算法，最后编程实现。

④ 将原来的被动填鸭式的灌输语言知识，变为自主的学习和探索，读者不再陷入语言的语法规则，而重在程序设计方法的学习和探究，通过编程掌握语言。

⑤ 在学习的不同阶段设计不同的针对性的实例，从而得到较好的教学效果。例如，开始阶段设计的实例是将学生的注意力吸引在 C 语言的总体功能和程序的总体框架上；在学习中间阶段设计针对某些数据类型或应用特点的实例、针对模块分解和组合的实例、针对算法分析与设计的实例等；在学习的后期进行综合课程设计，将所学知识融会贯通。

⑥ 引导读者掌握 C 语言的核心编程方法，提高应用能力。引导学生在解题编程的实践中探索其中的规律，将感性认识升华到理性高度。

全书分为理论部分和实验部分。理论部分共分 7 章。第 1 章介绍 C 语言的基本结构、语法成分、调试方法，简单 C 语言程序的设计、输入/输出语句等；第 2 章结合实际问题介绍 C 语言程序的基本结构；第 3 章介绍如何将复杂问题简单化处理的编程方法；第 4、5 章介绍如何编写具有构造数据类型的程序；第 6 章介绍磁盘数据存储程序的设计方法；第 7 章介绍实用程序设计的一般方法。

通过学习本书，你可以：

- 以任务驱动的方式了解 C 语言程序设计的基础知识
- 掌握 C 语言的核心编程方法
- 以程序编写带动语法的学习
- 建立程序设计的思想
- 通过上机实验提高程序设计能力
- 小试身手——利用 C 语言进行程序设计

本书的编写工作由刘明军主持。第 1、2 章由刘明军编写，第 3 章由王卫峰编写，第 4、5 章由黄艺美编写，第 6、7 章由韩玫瑰编写，实验部分由蒋彦和韩玫瑰老师编写，潘玉奇老师参加了部分内容及部分习题的编写。全书由刘明军统稿。中国石油大学（华东）的魏东平教授在百忙之中审阅了全书，济南大学的董吉文教授对本书的编写提出了宝贵意见。本书的编写参考了近年来出版的大量书籍及相关技术资料，吸取了许多同仁和专家的宝贵经验，在此一并表示衷心地感谢！

在琳琅满目的书海中，编写一本有特色并能使读者感兴趣的教材绝非易事。尽管我们付出了很大的努力，但由于水平有限，书中难免出现错误或不妥之处，我们诚恳地欢迎读者和同行批评指正。

联系地址：济南市济微路 106 号，济南大学信息科学与工程学院　刘明军

邮政编码：250022

电话：0531-82767505

E-mail：lmj@ujn.edu.cn

作　者

2007 年 1 月

目　录

第 1 章　C 语言程序基础 …… (1)

1.1　C 语言程序的基本结构 …… (2)

1.1.1　认识 C 语言程序 …… (2)

1.1.2　基本结构 …… (4)

1.1.3　C 语言的表达式和语句 …… (6)

1.2　C 语言程序运行过程 …… (7)

1.2.1　源程序、目标程序和可执行程序 …… (7)

1.2.2　C 语言程序上机步骤 …… (8)

1.3　编写简单的 C 语言程序 …… (9)

1.4　C 语言基本语法成分 …… (12)

1.5　C 语言数据类型 …… (14)

1.5.1　整型数据 …… (15)

1.5.2　实型数据 …… (16)

1.5.3　字符型数据 …… (16)

1.6　数据的输入/输出 …… (18)

1.6.1　printf( )函数 …… (19)

1.6.2　scanf( )函数 …… (20)

1.7　算法 …… (23)

1.7.1　算法的概念及特性 …… (23)

1.7.2　算法的表示方法 …… (23)

1.8　C 语言的产生、发展及特点 …… (25)

1.8.1　C 语言的产生及发展 …… (25)

1.8.2　C 语言的特点 …… (26)

习题 1 …… (28)

第 2 章　C 语言程序的基本结构 …… (31)

2.1　分支结构 …… (32)

2.1.1　单分支结构 …… (32)

2.1.2　双分支结构 …… (32)

2.1.3　多分支结构 …… (34)

2.1.4　if 语句的嵌套 …… (35)

2.1.5　条件运算符 …… (38)

2.1.6　switch 语句 …… (39)

2.2 关系运算和逻辑运算 …… (42)
2.2.1 关系运算符和关系表达式 …… (43)
2.2.2 逻辑运算符和逻辑表达式 …… (43)
2.3 循环结构 …… (45)
2.3.1 概述 …… (45)
2.3.2 当型循环 while …… (46)
2.3.3 直到型循环 do-while …… (48)
2.3.4 当型循环 for …… (50)
2.3.5 几种循环的比较 …… (53)
2.4 break 语句和 continue 语句 …… (54)
2.4.1 break 语句 …… (54)
2.4.2 continue 语句 …… (56)
2.5 goto 语句 …… (57)
2.6 经典算法举例 …… (58)
习题 2 …… (65)

第 3 章 模块化程序设计 …… (69)

3.1 模块化程序设计的方法和特点 …… (70)
3.2 函数的定义 …… (70)
3.3 无返回值函数的定义与调用 …… (73)
3.4 有返回值函数的定义与调用 …… (76)
3.5 函数嵌套调用和函数声明 …… (82)
3.6 函数的递归调用 …… (84)
3.7 库函数的使用 …… (86)
3.8 全局变量和局部变量 …… (89)
3.9 指针和指针作为函数参数 …… (93)
3.10 返回指针值的函数 …… (98)
3.11 函数的指针 …… (100)
3.12 典型例题 …… (102)
习题 3 …… (108)

第 4 章 简单构造数据类型 …… (113)

4.1 一维数组的引出及使用 …… (114)
4.1.1 一维数组的引出 …… (114)
4.1.2 一维数组的特点及引用 …… (115)
4.2 二维数组的引出及使用 …… (117)
4.2.1 二维数组的引出 …… (117)
4.2.2 二维数组的特点及引用 …… (119)
4.2.3 数组程序举例 …… (121)

4.3 字符数组 …… (125)
4.3.1 字符数组的引出 …… (125)
4.3.2 字符数组的定义和使用 …… (127)
4.3.3 字符串的定义和使用 …… (128)
4.3.4 字符数组程序举例 …… (134)
4.4 数组与函数 …… (137)
4.4.1 数组元素作为函数参数 …… (137)
4.4.2 数组名作为函数参数 …… (138)
4.4.3 多维数组作为函数参数 …… (140)
4.5 数组与指针 …… (141)
4.5.1 一维数组与指针 …… (141)
4.5.2 多维数组与指针 …… (144)
4.5.3 数组名作为函数参数 …… (144)
4.6 字符串与指针 …… (147)
4.6.1 字符串的指针和指针变量 …… (147)
4.6.2 用字符指针访问字符串 …… (148)
4.6.3 字符指针和字符数组的区别 …… (150)
4.6.4 字符指针作为函数参数 …… (152)
4.7 典型例题 …… (155)
习题 4 …… (166)

第 5 章 复杂构造数据类型 …… (177)

5.1 结构体 …… (178)
5.1.1 结构体的引出及使用 …… (178)
5.1.2 结构体数组的引出及使用 …… (183)
5.1.3 结构体程序举例 …… (186)
5.1.4 结构体与指针 …… (188)
5.2 共用体 …… (192)
5.2.1 共用体的引出 …… (193)
5.2.2 共用体的定义及使用 …… (195)
5.3 枚举类型 …… (196)
5.3.1 枚举类型的引出 …… (196)
5.3.2 枚举类型的定义及使用 …… (198)
5.4 链表 …… (199)
5.4.1 概述 …… (199)
5.4.2 简单链表 …… (200)
5.4.3 动态链表 …… (201)
5.4.4 输出链表及链表结构的实现和使用 …… (203)

习题5 …… (208)

第6章　磁盘数据存储 …… (213)

6.1　将数据写入文件 …… (214)
6.2　文件/读写分类函数 …… (217)
6.3　文件定位函数 …… (222)
习题6 …… (225)

第7章　实用程序设计技巧 …… (229)

7.1　程序的模块化结构 …… (230)
7.1.1　软件工程的思想 …… (230)
7.1.2　模块设计 …… (230)
7.1.3　使用模块化方法开发程序的好处 …… (231)
7.2　模块的组装 …… (232)
7.2.1　文件包含与头文件的使用 …… (233)
7.2.2　模块间的连接 …… (235)
7.2.3　标识符的一致性 …… (236)
7.2.4　条件编译 …… (237)
7.3　模块设计风格简述 …… (239)
7.3.1　数据风格 …… (239)
7.3.2　标识符风格 …… (239)
7.3.3　算法风格 …… (240)
7.3.4　输入/输出风格 …… (240)
7.3.5　书写风格 …… (241)
7.4　大型程序开发的项目管理 …… (242)
7.4.1　项目管理器 …… (242)
7.4.2　用项目管理器开发程序项目的步骤 …… (242)
7.4.3　项目管理器的使用技巧 …… (243)
7.5　几个应用程序设计实例 …… (243)
习题7 …… (254)

第8章　实验 …… (257)

实验1　C语言运行环境与C语言程序初步 …… (258)
实验2　顺序结构程序设计 …… (265)
实验3　选择结构程序设计 …… (268)
实验4　循环结构程序设计 …… (269)
实验5　函数编程 …… (272)
实验6　构造数据类型 …… (276)
实验7　磁盘数据存储 …… (280)

实验 8　项目开发与模块设计……（281）

附录……（283）

附录 A　Turbo C 2.0 集成开发环境的使用……（284）
附录 B　常用功能键简表……（289）
附录 C　Turbo C 编译出错信息……（291）
附录 D　常用 C 语言库函数……（303）

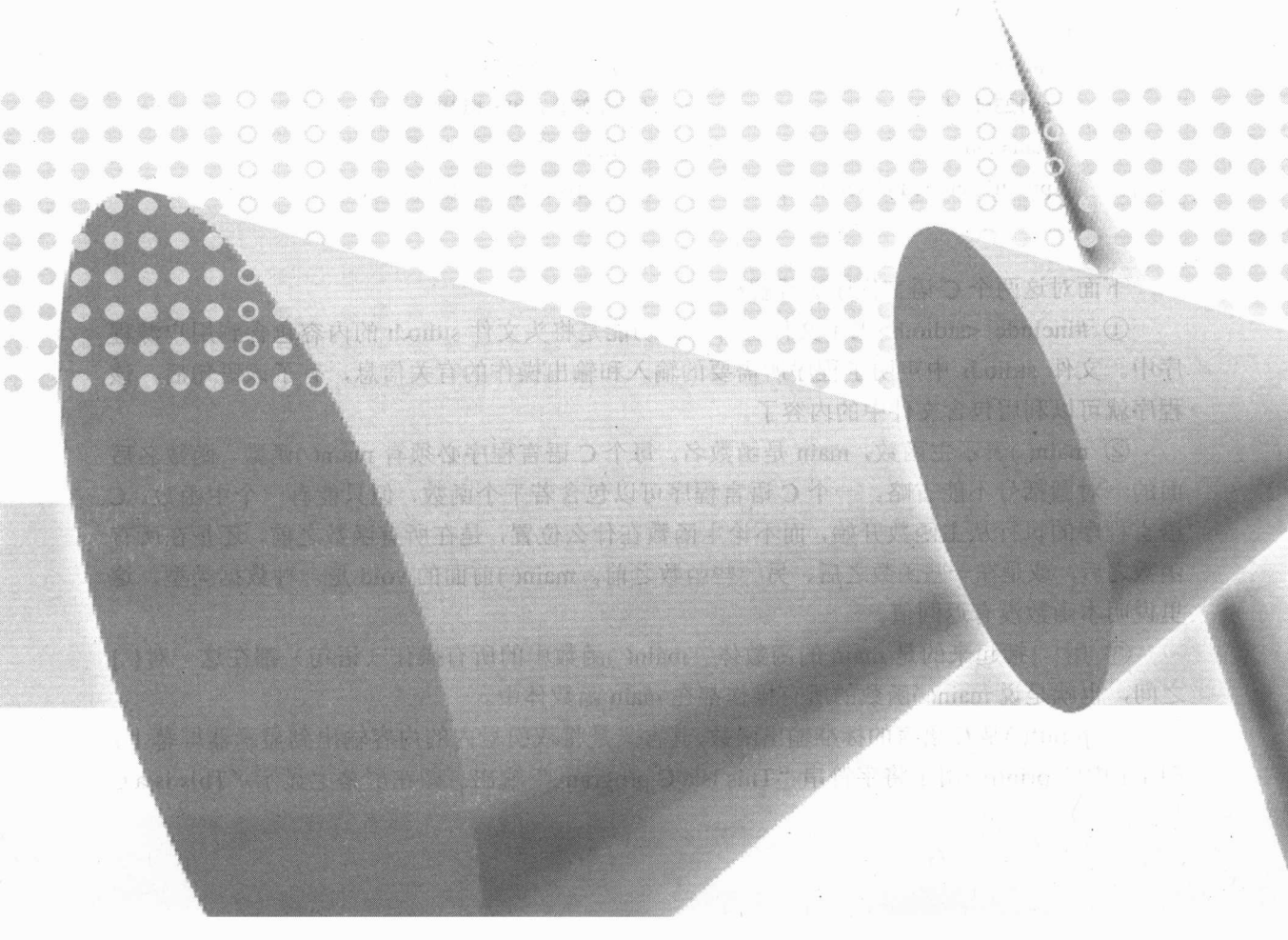
参考文献……（315）

第 1 章　C 语言程序基础

我们用计算机解决问题，一般利用某种软件。例如，使用 Word 编辑文件，用 Excel 处理电子表格，用 IE 浏览网络资源，用 QQ 上网聊天等。这些软件都是按照一定的算法编制的计算机程序，而算法是为了解决一个问题而采取的方法与步骤。

从广义的角度说，算法早就融入了我们的生活中。比如，早晨上学，就开始了上学的算法——走哪条路，坐什么车，如果堵车怎么办等，一直到到达学校，这个算法就完成了。用计算机解决问题也是按照相应的步骤（算法）一步一步完成的。这些步骤的实现用的是计算机语言，也就是编写计算机程序的语言。C 语言是一种计算机语言。

下面我们要学习的就是如何用 C 语言编写计算机程序。



1.1　C 语言程序的基本结构

1.1.1　认识 C 语言程序

首先来看两个简单的 C 语言程序。

【例 1.1】 在屏幕上输出一行信息“This is a C program.”。

C 语言程序如下：

```
#include <stdio.h>                    /* 预处理命令 */
void main( )                          /* 定义主函数 */
{
  printf("This is a C program.\n");   /* 调用库函数，输出信息 */
}
```

【例 1.2】 计算两数之和，并输出结果。

C 语言程序如下：

```
#include <stdio.h>                    /* 预处理命令 */
void main( )                          /* 定义主函数，计算两数之和 */
{
  int a,b,sum;                        /* 定义 3 个整型变量 */
  a=123; b=456;                       /* 为变量 a 和 b 赋初值 */
  sum=a+b;                            /* 让 sum 等于 a+b 的值 */
  printf("sum=%d\n",sum);             /* 调用库函数，输出结果 */
}
```

下面对这两个 C 语言程序进行解析。

① #include <stdio.h>表示文件包含，其功能是将头文件 stdio.h 的内容包含到用户源程序中。文件 stdio.h 中声明了程序所需要的输入和输出操作的有关信息，有了该语句后，该程序就可以利用包含文件中的内容了。

② main( )表示主函数，main 是函数名。每个 C 语言程序必须有 main( )函数。函数名后面的一对圆括号不能省略。一个 C 语言程序可以包含若干个函数，但只能有一个主函数。C 语言程序的执行从主函数开始，而不论主函数在什么位置，是在所有函数之前，还是在所有函数之后，或是在一些函数之后、另一些函数之前。main( )前面的 void 是一种数据类型，这里说明本函数没有返回值。

③ 用{ }括起来的是 main 的函数体。main( )函数中的所有操作（语句）都在这一对{ }之间，也就是说 main( )函数的所有操作都在 main 函数体中。

④ printf( )是 C 语言的标准输出函数，其含义是将双引号内的内容输出到显示器屏幕上。例 1.1 中的 printf( )用于将字符串“This is a C program.”输出。即在屏幕上显示“This is a C

program."。\n 是换行符，即使光标跳到下一行行首（第一列）。例 1.2 中的%d 为格式控制符，表示在此位置将用一个十进制整数替代，该整数由逗号后面的变量 sum 提供。程序运行后，将在屏幕上输出“sum=579”。

⑤ 分号“;”是 C 语言的语句结束符，表示该语句结束。

⑥“/* */”括起来的部分是一段注释，注释只是为了改善程序的可读性，在编译、运行时不起作用（事实上编译时会跳过注释，目标代码中不会包含注释）。注释可以放在程序的任何位置，并允许占用多行，只是需要注意“/*”与“*/”匹配，不能嵌套注释。

⑦“int a,b,sum;”是变量声明。声明了 3 个具有整数类型的变量 a、b 和 sum。C 语言的变量必须先声明再使用。

⑧“a=123;b=456;”是两条赋值语句。将 123 赋给变量 a，将 456 赋给变量 b。执行上述两条语句后，a 和 b 两个变量的值分别为 123 和 456。

⑨“sum=a+b;”是将 a 和 b 两个变量的内容相加，然后将结果赋值给整型变量 sum。此时，sum 的内容为 579。

【例 1.3】 输入两个整数，计算并输出两者中较大的数。

C 语言程序如下：

```
#include <stdio.h>
void main( )                          /* 主函数 */
{                                     /* main 函数体开始 */
  int a,b,c;                          /* 声明部分，定义变量 */
  scanf("%d%d",&a,&b);                /* 调用库函数，输入变量 a 和 b 的值 */
  c=max(a,b);                         /* 调用 max 函数，将调用结果赋给 c */
  printf("max=%d\n",c);
}                                     /* main 函数体结束 */
int max(int x,int y)                  /* max 函数，用于计算两数中较大的数 */
{                                     /* max 函数体开始 */
  int z;                              /* 声明部分，定义变量 */
  if (x>y) z=x;
  else z=y;
  return z;                           /* 将 z 值返回，通过 max 返回调用处 */
}                                     /* max 函数体结束 */
```

说明：

① 本程序包括两个函数。其中，主函数 main( )是整个程序执行的起点。函数 max( )的功能是计算两个数中较大的数。

② scanf( )是 C 语言的标准输入函数，用于从键盘输入若干数据给指定的变量。%d 表示输入十进制整数。主函数 main( )调用 scanf( )函数获得两个整数，存入 a 和 b 两个变量，然后调用函数 max( )获得两个数中较大的值，并赋给变量 c，最后输出变量 c 的值（结果）。

③“int max(int x,int y)”是函数 max( )的函数头，函数 max( )的函数头表明此函数获得两个整数，返回一个整数。

④ 用{ }括起来的部分是 max( )函数的函数体。max( )的函数体是函数 max( )的具体实现。它从参数表获得数据，处理后得到结果 z，然后将 z 返回调用函数 main( )。

⑤ 本例表明除了可以调用库函数外，还可以调用用户自己定义和编写的函数。

1.1.2 基本结构

综合上述 3 个例子，可以对 C 语言程序的基本组成和形式（程序结构）做以下说明。

C 语言程序的结构如下：

```
预处理命令
void main( )                    /* 主函数 */
{                               /* 函数体开始 */
    声明部分
    执行部分
}                               /* 函数体结束 */
其他函数
{
    声明部分
    执行部分
}
```

下面对 C 语言程序的结构进行简单说明。

1. 函数是 C 语言程序的基本单位

一个 C 语言源程序必须包含一个 main( )函数，也可以包含其他函数。函数是 C 语言程序的基本单位。C 语言是函数式的语言，程序的全部工作都是由各个函数完成的。编写 C 语言程序就是编写一个个函数。

程序中的函数有些是 C 语言的标准库提供的，称为标准函数（或库函数）。例如，printf( )函数和 scanf( )函数。编写程序时，如果有标准函数，就使用标准函数，如果没有标准函数，则需要用户自己编写。其他函数可以是系统提供的库函数，也可以是用户根据需要自己编写的函数。

C 语言函数库非常丰富，ANSI C 提供 100 多个库函数，Turbo C 提供 300 多个库函数。

2. main( )函数

主函数是每个程序执行的起点。一个 C 语言程序总是从 main( )函数开始执行，而不论 main( )函数在程序中的位置。可以将 main( )函数放在整个程序的最前面，也可以放在整个程序的最后，或者放在一些函数之后另一些函数之前。一个程序只能有一个 main( )函数。

3. 函数的结构

一个函数由函数首部和函数体两部分组成。

（1）函数首部

函数首部在一个函数的第一行，格式如下：

返回值类型 函数名([函数参数类型 1 函数参数名 1],……,[函数参数类型 *n*，函数参数名 *n*])

例如：

```
int max(int x,int y)
```

注意：一个函数可以没有参数，但是后面的一对( )不能省略，这是格式的规定，如 main( )。

（2）函数体

函数体是函数首部下面用一对{ }括起来的部分。如果函数体内有多对{ }，则最外层是函数体的范围。函数体一般包括声明部分和执行部分。

- 声明部分：定义本函数所使用的变量，并为变量分配相应大小的内存单元。变量名是内存单元的符号地址。
- 执行部分：由若干条语句组成的命令序列（可以在其中调用其他函数）。

4．书写风格

C 语言程序书写格式自由，一行可以写几个语句，一个语句也可以写在多行上。每条语句的最后必须有一个分号“;”，表示语句的结束。

5．输入/输出

C 语言本身不提供输入/输出语句，输入/输出操作是通过调用库函数（如 scanf( )和 printf( )）完成的。

由于输入/输出操作涉及具体的计算机硬件，因此把输入/输出操作放在函数中处理可以简化 C 语言和 C 语言的编译系统，便于 C 语言在各种计算机上实现。

不同的编译系统除了提供函数库中的标准函数外，还按照硬件的情况提供一些专门的函数。因此，不同编译系统提供的函数的数量和功能会有一定差异。

6．注释

注释部分要放在符号“/*”和“*/”之间。为了便于阅读，程序中应适当地加入注释。注释只起帮助阅读和理解程序的作用，不参加程序的编译，也不影响程序的运行。使用注释是编程人员的良好习惯。

在实践中，编写程序往往需要不断地修改、完善，事实上没有一个应用系统是不需要修改、完善的。很多人会发现自己编写的程序在经历了一些时间以后，由于缺乏必要的文档和必要的注释，最后连自己都很难再读懂。需要花费大量时间重新思考、理解原来的程序，从而浪费了大量的时间。如果开始编程就对程序进行注释，刚开始麻烦一些，但日后可以节省大量的时间。

7．预处理命令

预处理命令能够改进程序的设计环境，提高编程效率。C 语言的预处理功能主要包括宏

定义、文件包含、条件编译，分别用宏定义命令（#define）、文件包含命令（#include）、条件编译命令（#ifdef-#else-#endif）实现，为了与一般语句区别，这些命令以“#”开头。

1.1.3　C 语言的表达式和语句

用表达式构成语句，表示一种运算或操作。表达式语句是在表达式的最后加上一个“;”构成。一个表达式语句必须在最后出现分号，分号是语句不可缺少的一部分。C 语言程序中大多数语句是表达式语句（包括函数调用语句）。表达式与表达式语句举例见表 1-1。

表 1.1　表达式与表达式语句举例

表 达 式	表达式语句	说　明
a=3（赋值表达式）	a=3;（赋值语句）	将 3 赋给变量 a
getch()（函数调用表达式，函数调用也属于表达式）	getch(); （函数调用语句）	“getch();”合法且有意义，只关心是否有调用函数，不关心具体的函数值
i++ （自增表达式）	i++;（一般表达式语句）	变量 i 的值加 1
ch=getch()（函数调用表达式，赋值表达式）	ch=getch(); （一般表达式语句）	将函数的返回值赋给变量 ch
x+y（算术表达式）	x+y;（一般表达式语句）	“x+y;”是一个语句，其作用是完成 x+y 操作，是合法的，但是并不将结果赋给另外的变量，所以并无实际意义

表达式语句的一般形式如下：

表达式;

表达式语句常见的形式有：赋值语句、函数调用语句和空语句。

1．赋值语句

赋值语句由赋值表达式加上一个分号构成。

C 语言的赋值语句先计算赋值运算符（=）右边子表达式的值，然后将此值赋值给左边的变量。

对变量的赋值的一般格式如下：

<变量>=<表达式>

例如：

```
b=30.0;                    /* 将表达式的值 30.0 赋给变量 b */
a=sin(b*3.14159/180);      /* 将表达式（正弦函数）的值赋给变量 a */
```

变量赋值的特点如下。

① 变量必须先定义，后使用。例如“int d,e,f;”定义 3 个变量为整数类型。如果未定义，则在编译时认为非法，提示错误。

② 变量被赋值前，值不确定。