21世纪

计算机应用技术系列规划教材

C语言程序设计教程

◎ 李丽娟 主编 ◎
◎ 吴蓉晖 副主编 ◎

人民邮电出版社
POSTS & TELECOM PRESS

21世纪计算机应用技术系列规划教材

C语言程序设计教程

李丽娟　主　编

吴蓉晖　副主编

人民邮电出版社

图书在版编目（CIP）数据

C 语言程序设计教程/李丽娟主编．—北京：人民邮电出版社，2006.2
（21 世纪计算机应用技术系列规划教材）
ISBN 7-115-14516-4

Ⅰ．C… Ⅱ．李… Ⅲ．C 语言—程序设计—教材 Ⅳ．TP312

中国版本图书馆 CIP 数据核字（2006）第 007949 号

内 容 简 介

本书内容分为三部分，共 11 章。第 1、2 章为第一部分，为初学者的入门知识，简单介绍 C 语言编写程序的步骤、方法和程序结构。第 3、4、5 章为第二部分，是程序设计的基础部分，详细描述程序算法的流程图及 C 语言的控制结构。掌握了第一、二部分的内容，读者可以完成简单的程序设计。第 6、7、8、9、10、11 章为第三部分，讲述模块化程序设计的概念和实现的方法，以及图形方式的程序设计方法和对文件的操作。

本书每一章都有详细的程序范例，以说明程序语句的概念、作用、含义和使用方法。全书语言通俗易懂，讲解由浅入深，注重程序语句本身的功能与作用，强调对 C 语言语句的掌握。

本书既适合作大学本科和专科院校的教材，也可作一般工程技术人员的参考书。

21 世纪计算机应用技术系列规划教材

C 语言程序设计教程

◆ 主　　编　李丽娟
副 主 编　吴蓉晖
责任编辑　邹文波

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：18.5
字数：446 千字　　2006 年 2 月第 1 版
印数：1–4 000 册　　2006 年 2 月北京第 1 次印刷

ISBN 7-115-14516-4/TP・5240

定价：29.50 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223

前　言

C 程序设计语言是许多大学计算机基础课程中一门重要的计算机基础必修课程，该课程的目的是培养学生的逻辑思维和抽象思维的能力，学会提出问题、解决问题，掌握综合分析问题的方法，培养一种科学的思考问题和解决问题的能力，并通过 C 程序设计语言这门课程，为今后进一步学习计算机基础知识打下基础。

本书作者多年来从事 C 程序设计语言课程的教学，并从事计算机应用、模式识别等领域的研究，具有较丰富的科研实际经验和程序开发能力，对程序设计有较好的研究，同时在 C 程序设计语言的教学过程中有较好的经验积累，书中的很多地方是作者总结了教学中间的经验。全书将 C 语言的知识分成三大板块：入门、程序设计的基本结构和模块化程序设计的结构。使学习过程有一个台阶，在入门这个板块中，目的是使学生清楚地了解 C 语言程序的编辑、编译、运行及基本程序语句；在程序设计的基本结构板块中介绍了制定程序算法的流程图和两种重要的程序语句结构：分支结构和循环结构，目的是使学生掌握分析问题的方法，学会用 C 语言程序的结构来思考问题和解决问题；在模块化程序设计的结构板块中，目的是要使学生了解模块化程序设计的思想，掌握怎样通过程序功能来划分程序模块，并进一步通过指针、结构、文件、位运算等全面了解 C 语言程序设计的手段和方法，培养和锻炼开放性思考问题和解决问题的能力。

本书的一个特点是书中有许多程序范例留有充分的可变性，既给教师教学留下了丰富的启发空间，也给读者留下了可独立深入研究和探索的空间，满足了不同层次的需求。针对学生学习过程中可能遇到的问题，本书每一章的后面都配备了大量的程序范例和习题，帮助读者理解基本概念，通过书中的习题，进一步熟练掌握 C 语言的语法结构和应用，提高程序设计能力。

C 语言本身并不难学，学习用 C 语言进行程序设计是一种综合思维训练。本书可以为那些还没有任何程序设计经验的学生提供一个良好的入门教程。通过系统的学习，正确认识什么是程序，并学会 C 语言程序的编写，掌握思考问题和解决问题的方法，并初步掌握用 C 语言来进行程序设计。

C 语言是一种非常有效的编程语言，它简练的语法结构和灵活的机制是开发系统软件和应用软件的首选语言。同时，C 语言又是 C++程序设计语言的基础，学会了 C 语言，也就为将来学习 C++程序设计语言打下了一个坚实的基础。为了让读者既能够学会 C 语言的语法规则，又能够解决一些实际问题，本书提供了丰富的程序范例和习题，这些程序范例的难度适中，希望读者通过本书中的程序范例，体会到程序设计的过程，并且可以改写程序，使程序完善。

本书精选了 140 多个程序，每一个程序都是经过了上机验证的，并且对程序的结构、函数的设计、变量的设置进行了有效地说明，同时还引申出一些有深度的问题供读者思考，借以提高程序设计的能力。

与本书配套的《C 语言程序设计教程实验指导与习题解答》给出了本书中习题的全部参考答案和学生

上机实验的内容。通过完成实验、上机编写程序，达到掌握知识的目的。还可以将C语言概念方面的一些问题，编写成程序，然后编译、运行，查看程序的运行结果。从程序的运行结果可以讨论程序的正确与否，从而真正掌握C语言程序设计的基本方法和基本技能。

本书的特点是通俗易懂，由浅入深，内容安排紧凑，可操作性强。系统讲述了C语言的基本语法结构，有一些地方是吸取了C语言原创作者的本意。希望能带给读者最真实的一面。

C语言程序设计课程建议学时数为88，其中课堂教学时数为40学时，上机实验学时为48学时，各章节的教学时数可大致划分如下：

章　节	内　容	课堂教学时数	上机实验时数
1	引言	2	2
2	基本的程序语句	4	6
3	程序的简单算法制定	2	2
4	分支结构	2	4
5	循环结构	2	4
6	函数与宏定义	8	8
7	数组	4	4
8	指针	8	8
9	构造数据类型	4	4
10	文件操作	2	4
11	位运算	2	2

实际教学中可以根据具体情况予以调整，适当减少或增加学时数。

感谢湖南大学计算机与通信学院李仁发教授对本书的支持和关心，感谢湖南大学计算机与通信学院的龚如华老师和杨晓林老师对本书提出的宝贵意见。

由于作者水平有限，书中难免会有错误。希望各位读者和专家提出宝贵的意见，帮助作者不断改进和完善，本人将不胜感激。

如需要书中的原程序代码，请与作者联系：jt_lljh@hnu.cn。

李丽娟

2005年12月于岳麓山

目 录

第 1 章 引　言

1.1　C 语言的发展过程

C 语言是在 20 世纪 70 年代初问世的。1978 年美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时 B.W.Kernighan 和 D.M.Ritchit 合著了著名的“*THE C PROGRAMMING LANGUAGE*”一书，通常简称为“K&R”，也有人称之为“K&R”标准。但是，在“K&R”中并没有定义一个完整的标准 C 语言，后来美国国家标准协会（American National Standards Institute，ANSI）在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C。

1.2　C 语言的特点

1．C 语言简洁、紧凑，使用方便、灵活。程序书写自由，共有 9 种控制语句。ANSI C 规定 C 语言共有 32 个关键字。关键字就是已被编程语言本身使用的标识符，不能作变量名、函数名等其他用途。C 语言的关键字如表 1-1 所示。

表 1-1　　C 语言的关键字

auto	const	double	float	int	short	struct	Unsign
break	continue	else	for	long	signed	switch	Void
case	default	enum	goto	register	static	typedef	Volatile
char	do	extern	if	return	sizeof	union	While

不同的实现对 C 语言的关键字有不同的扩充，Turbo C 扩充了 11 个关键字，如表 1-2 所示。

表 1-2　　Turbo C 扩充的关键字

asm	_cs	_ds	_es	_ss	cdecl
far	huge	interrut	near	pascal	

一般而言，C 语言的关键字不允许用户将其用作其他用途，在 C 语言中，所有关键字都是小写的。

2．运算符丰富。运算符共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理，从而使 C 语言的运算类型极为丰富，可以实现其他高级语言难以实现的运算。

3．数据结构类型丰富。

4．具有结构化的控制语句。

5．语法限制不太严格，程序设计自由度大。

6．C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此，有人把它称为中级语言。

7．生成目标代码质量高，程序执行效率高。

8．与汇编语言相比，用 C 语言写的程序可移植性好。

但是，C 语言对程序员要求也高，程序员用 C 写程序会感到限制少、灵活性大，功能强，但在学习上较其他高级语言要困难一些。

1.3 简单的 C 语言程序

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由易到难，表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到组成一个 C 源程序的基本部分和书写格式。

【例 1-1】

```
#include<stdio.h>
main()
{
    printf("Hello,world! /n");
}
```

程序说明：

1．include 称为文件包含命令，扩展名为.h 的文件称为头文件，在这里包含的文件是 stdio.h，它表示在程序中要用到这个文件中的函数。‘#’是一个标志。

2．main 是主函数的函数名，表示这是一个主函数。

3．printf 是函数调用语句，printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是一个由系统定义的标准函数，在 stdio.h 中，可在程序中直接调用。

4．main()函数中的内容必须放在一对花括号“{}”中。

【例 1-2】

```
#include<stdio.h>
#include<math.h>
main()
{
    double x,s;
    prinft("input number: ");
    scanf(" %lf ", &x);
    s=cos(x);
    printf("cos(%lf)is %lf/n", x,s);
```

```
}
```

程序说明：

1．程序包含了两个头文件：stdio.h、math.h。

2．在 main 函数中定义了两个双精度实数型变量 x、s。

3．printf("input number:");用于显示提示信息。

4．scanf(" %lf ",&x)；用于从键盘获得一个实数 x。

5．s=cos(x);求 x 的余弦，并把它赋给变量 s。

6．printf ("cos(%lf)is%lf\n",x,s)；显示程序运算结果。双引号"…"中有两个格式字符%lf，分别对应着 x 和 s 两个输出变量。

程序的功能是从键盘输入一个数 x，求 x 的余弦值，然后输出结果。include 称为文件包含命令，其意义是把尖括号<>内指定的文件包含到本程序中，成为本程序的一部分。被包含的文件可以是由系统提供的，其扩展名为.h。因此，也称为头文件或首部文件。C 语言的头文件中包括了各种标准库函数的函数原型。因此，凡是在程序中调用一个库函数时，都必须包含该函数的原型所在的头文件。在本例中，使用了 3 个库函数：输入函数 scanf，余弦函数 cos，输出函数 printf。余弦函数 cos 是数学函数，其头文件为 math.h，因此在程序的主函数前用 include 命令包含了 math.h。scanf 和 printf 是标准输入/ 输出函数，其头文件为 stdio.h，在主函数前也用 include 命令包含了 stdio.h 文件。

需要说明的是，有些编译环境对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令。所以在例 1-1 和例 1-2 中可以省略文件包含命令#includ<stdio.h>。

但是，在任何情况下都写上#include<stdio.h>是一个良好的习惯，否则，会由于编译系统的不同而发生语法错误。

【例 1-3】

```
#include<stdio.h>
int add(int x, int y);
main()
{ int a, b, c;
   printf ("please input value of a and b:\n");
   scanf("%d %d", &a, &b);
   c=add(a,b);
   printf ("max=%d",c);
}
int add(int x, int y)
{
   return(x+y);
}
```

在例 1-3 中的主函数体又分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变量的类型说明。例 1-1 中未使用任何变量，因此无说明部分。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，说明部分是 C 源程序结构中很重要的组成部分。

运行本程序时，首先在显示屏幕上给出提示串 please input value of a and b:，这是由执行部分的第一行完成的。用户在提示下从键盘上键入两个数，如 5↵9↵，（或 5 9↵）接着在屏幕上会显示出计算结果：max＝14。

1.4 C 语言程序的结构

一个 C 语言程序可由下面不同的部分组合而成：

① 文件包含部分；

② 预处理部分；

③ 变量说明部分；

④ 函数原型声明部分；

⑤ 主函数部分；

⑥ 函数定义部分。

关于程序的结构的说明：

1．并不是所有的 C 程序都必须包含上面的 6 个部分，一个最简单的 C 程序可以只包含文件包含部分和主函数部分两部分。

2．每一个 C 程序文件的后缀为.c，并且每一个 C 源程序都必须有且仅有一个主函数，主函数的组成形式如下所示：

```
mian()
{
    变量说明部分
    程序语句部分
}
```

每一个 C 语言的语句由分号结束。在以后的章节里，会详细地讲述 C 程序中各部分的作用和使用方法。

1.5 C 程序设计语言的执行

用程序语言编写的程序称为源程序（Source Program），实际上计算机本身并不能直接理解这样的语言，必须将程序语言翻译成机器语言，计算机才能理解程序。将源程序翻译成机器语言的过程称为编译，编译的结果是得到源程序的目标代码（Object Program）；最后还要将目标代码与系统提供的函数和自定义的过程（或函数）链接起来，就可得到机器可执行的程序。机器可执行的程序称为可执行程序或执行文件。

1.5.1 源程序翻译

适合 C 语言的编译器不止一种。一般地，对 C 源程序的翻译过程如图 1-1 所示。它由以下 3 个部分组成：

1．词法分析器（Lexical）；

2．语法分析器（Parser）；

3．代码生成器（Code Generator）。

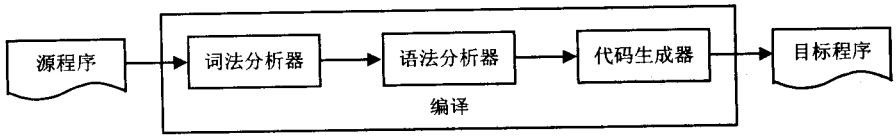

图 1-1 程序语言的翻译过程

词法分析器主要是对源程序进行词法分析，它是按单个字符的方式阅读源程序，并且识别出哪些符号的组合可以代表单一的单元，并根据他们是否是数字值、单词(标识符)、运算符等，将这些单词分类。词法分析器将词法分析结果保存在一个结构单元里，这个结构单元称为记号（Token），并将这个记号交给语法分析器，词法分析会忽略源程序中的所有注释。

语法分析器直接对记号进行分析，并识别每个成分所扮演的角色。这些语法规则也就是程序设计语言的语法规则。

代码生成器是将经过语法分析后，没有语法错误的程序指令转换成机器语言指令。

例如，假定编写了一个名为 mytest 的程序，源程序的全名为 mytest.c，用 microsoft C 编译器，在命令方式下，可以采用下面这样的方式对 mytest.c 进行编译：

```
cl –c mytest.c
```

如果源程序没有错误，就会生成一个名为 mytest.obj 目标代码程序。其他程序语言也会有类似的命令将源程序翻译成目标代码，具体的命令与每种程序语言的编译器有关。

1.5.2 链接目标程序

通过翻译产生的目标代码尽管是机器语言的形式，但却不是机器可以执行的方式，这是因为为了支持软件的模块化，允许程序语言在不同的时期开发出具有独立功能的软件模块作为一个单元，一个可执行的程序中有可能包含一个或多个这样的程序单元，这样可以降低程序开发的低水平重复所带来的低效率。因此，目标程序只是一些松散的机器语言，要获得可执行的程序，还需将它们链接起来。

程序的链接工作由链接器（Linker）来完成。链接器的任务就是将目标程序链接成可执行的程序（或称载入模块），这种可执行的程序是一种可存储在磁盘存储器上的文件。

例如，假设已对源程序 mytest.c 进行编译后生成了目标代码程序 mytest.obj，于是，我们可以利用链接器生成可执行代码，在命令方式下，将用下面这样的方式来链接程序：

```
link /out:mytest.exe mytest.obj
```

如果不发生错误，就会生成一个名为 mytest.exe 的载入模块，也就是可执行的代码程序。最后，可以通过操作系统将这个加载模块加载入内存，执行程序的进程。

上面对程序进行编译，连接都只针对了一个源程序文件，实际上，可以将多个源程序文件通过编译，链接成一个可执行文件。

例如，假定有 3 个源程序：file1.c，file2.c 和 file3.c，每一个源程序都包含有不同的函数或过程，在命令方式下可先用编译器对 3 个源程序进行编译：

```
cl  -c file1.c
cl  -c file2.c
cl  -c file3.c
```

分别得到 3 个目标程序：file1.obj，file2.obj 和 file3.obj。接下来可用链接器将 3 个目标程序进行链接。

```
Link /out:mytest.exe file1.obj file2.obj file3.obj
```

可得到一个可执行的程序：mytest.exe。

对于程序的编译、链接，有必要强调以下几点：

1. 并不是任何目标程序都可以连接成可执行程序。

2. 被链接成可执行程序的目标程序中，只允许在一个程序中有且仅有一个可被加载的入口点，即只允许在一个源程序中包含一个 main()函数。在上面的范例中这个可被加载的入口点在源程序 file1.c 中。

3. 对于具体的程序语言，翻译、链接程序的方法会有所不同，针对某一种程序语言的编译器，不可以用于对另一种源程序语言的编译。

4. 上面对 C 语言进行编译、链接的方式并不是惟一的，它允许有一些其他的变化，具体可参考各编译器的使用说明。

总之，完成一个 C 语言程序的完整过程主要包括 4 个部分：编辑、编译、链接和加载，如图 1-2 所示。

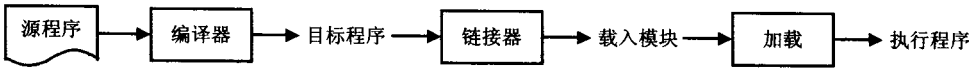

图 1-2　完整的程序生成过程

一旦生成了可执行程序，就可以反复的被加载执行，而不需要重新编译、链接，如果修改了源程序，也不会影响到已生成的可执行程序，除非对修改后的源程序重新编译和链接，重新生成一个新的可执行程序。

1.5.3　集成开发环境

显然，用命令方式来生成可执行的程序，并不是很方便，尤其是源程序的编辑，一般地，纯文本编辑器都可以输入源程序，如果编译时有错误，就必须回到编辑器修改程序。如此反复，使得程序开发效率不高。

程序的集成开发环境是一个经过整合的软件系统，它将编辑器、编译器、链接器和其他软件单元集合在一起，在这个环境里，程序员可以很方便地对程序进行编辑、编译、链接以及跟踪程序的执行过程，以便寻找程序中的问题。

适合 C 语言的集成开发环境有许多，如：Turbo C，Borland C，gcc，Microsoft c 等等。许多软件开发包已使用了图形用户界面，使软件的工作变得更为简单、快捷。以适应于不同的操作系统，如 DOS、Windows，UNIX 等。

1.6　本章小结

本章简要介绍了 C 语言的特点和发展过程，并介绍了 C 语言程序的基本结构，使读者对 C 语言程序有一个大致的初步了解，还介绍了 C 语言程序的开发过程：编辑源程序、对源程序进行编译以生成目标代码、连接目标代码生成可执行的程序，运行可执行的程序。有不同的集成开发环境可以用来开发 C 语言程序，为学习好 C 程序设计语言奠定了一个良好的认知基础。

习 题

【题 1.1】 ANSI C 是如何形成的？

【题 1.2】 由 B.W.Kernighan 和 D.M.Ritchit 合著的“The PROGRAMMING LANGUGE”是否定义了完整的标准 C 语言？

【题 1.3】 “K&R”标准指的是什么？

【题 1.4】 标准 C 和扩展 C 是什么关系？

【题 1.5】 标准 C 语言中的关键字共有多少个？是否可以用在大写字母表示？

【题 1.6】 怎样在程序中使用系统提供的函数？

【题 1.7】 C 语言程序可由哪些不同的部分组合而成？

【题 1.8】 实践性练习

（1）查阅资料，了解 C 语言的发展过程。

（2）熟悉一种 C 语言的集成开发环境：安装、运行。

（3）运行本章的 3 个范例程序，掌握程序的运行步骤和方法。