高等院校计算机教育系列教材

C语言程序设计基础与应用

刘 丽 朱俊东 张 航 编著

- 知识点新，突出实践教学，强化能力培养
- 理论知识+感性认识+动手实践，完美结合
- 内容简明扼要，突出知识要点
- 以实用为宗旨，实例丰富，用实例引导读者模仿学习

赠送电子课件

清华大学出版社

高等院校计算机教育系列教材

C语言程序设计基础与应用

刘 丽 朱俊东 张 航 编著

清华大学出版社
北京

内 容 简 介

本书是在作者多年讲授C语言程序设计的基础上，总结多年的教学经验和实践体会编写而成的。本书采用由浅入深、循序渐进的原则，系统地介绍了C语言的基本语法知识，通过大量实例描述C语言的程序设计方法，并针对实例给出了算法分析，注重培养学生程序设计的思维方法和程序设计能力。

本书共分为13章，第1～3章介绍C语言的概念、数据类型和基础知识；第4～6章介绍C语言的三种基本结构；第7～13章介绍C语言中的数组、函数、结构体、文件和编译预处理的基础知识和编程技巧。综观全书，既有基础知识的介绍，也有各种算法的分析；既有生动的实例讲解，也有典型经验的分享。

本书既可以作为高等学校各专业的正式教材，也适合自学使用。另外，在本书中，兼顾了全国计算机等级考试二级C语言程序设计考试大纲的相关内容，也可以作为考试辅导教材使用。

图书在版编目(CIP)数据

C语言程序设计基础与应用/刘丽，朱俊东，张航编著. --北京：清华大学出版社，2012.12
(高等院校计算机教育系列教材)
ISBN 978-7-302-30035-9

Ⅰ. ①C…　Ⅱ. ①刘… ②朱… ③张…　Ⅲ. ①C语言—程序设计—高等学校—教材　Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第211751号

责任编辑：汤涌涛　魏　莹
装帧设计：刘孝琼
责任校对：李玉萍
责任印制：张雪娇

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62791865

印 刷 者：清华大学印刷厂
装 订 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：25.5　　**字　　数**：618千字
版　　次：2012年11月第1版　　**印　　次**：2012年11月第1次印刷
印　　数：1～4000
定　　价：45.00元

产品编号：046616-01

前　言

为何编写本书

由于 2006 年教育部高等学校计算机科学与技术教学指导委员会关于“关于进一步加强高等学校计算机基础教学的意见”中专门提到了要加强大学生在计算机程序设计及信息素养方面的培养，C 语言是国际上广泛流行的一种计算机语言，而 C 语言因数据类型丰富、表达能力强、使用灵活方便、生成目标代码质量高，并且可直接对硬件进行操作，在国内外得到广泛应用。因此目前国内的很多高校都将 C 语言作为本科学生学习程序设计的入门课程。通过学习该课程，学生不仅可以掌握一门编程语言，而且还可以掌握结构化程序设计的基本理论、编程思想、编程方法、常用算法和实现技术。另外，还可以锻炼学生的逻辑思维能力，培养学生的创新精神，进而提高学生的综合素质。

本书是作者在讲授 C 语言程序设计的基础上，总结多年的教学经验和实践体会编写而成的。本书既可以作为高等学校各专业的正式教材，也适合自学使用。另外，在本书中，兼顾了全国计算机等级考试二级 C 语言程序设计考试大纲的相关内容，也可以作为考试辅导教材使用。

本书内容特色

1. 由浅入深、循序渐进

本书内容由浅入深、循序渐进，介绍 C 语言的基本语法知识和程序设计等内容，同时兼顾了全国计算机等级考试二级 C 语言程序设计考试大纲的内容。书中结合大量实例描述 C 语言的重要特性，并对很多代码给出逐步的分析，以这种独特的教学方法向读者解释 C 语言的编程元素及方法。

2. 案例新颖、趣味性强

针对初学者的特点，作者精心策划了书的框架及内容，书中概念清晰，逻辑性强，内容通俗易懂，以丰富的例题深入浅出地阐述知识点。书中的每个案例都由作者精心设计，趣味性较强，通过这些案例，不仅可以提高读者学习的兴趣，也可以引导读者对所学知识点举一反三，从而使读者更深刻地理解所学习的知识点。

3. 通俗易懂，针对性强

本书面向初学者，强调应用性，注重实例的分析。作者在编写每个案例时，都先对案例进行分析，提高读者分析问题的能力；然后编写代码，并在代码中给出大量注释；最后在案例之后提出一些思考问题，提高读者独立思考问题的能力。

适用读者群

(1) 初学编程的自学者。
(2) 编程爱好者。
(3) 大、中专院校的老师和学生。
(4) 相关培训机构的老师和学员。
(5) 初、中级程序开发人员和程序测试及维护人员。

参加本书编写的人员

参加本书编写工作的有刘丽、朱俊东、张航等。其中本书的第 1 章和第 2 章由张航编写，第 3～5 章由朱俊东编写，第 6～13 章由刘丽编写。

由于时间仓促，书中难免有遗漏和不足之处，恳请广大读者提出宝贵意见。联系方式为：tsliuli@126.com。

编　者

目　录

第1章

C语言程序设计概述

随着计算机技术的不断发展，计算机已经应用到社会的各个领域，如文字处理 Word、表格计算 Excel、各种数据库管理软件等。这些软件都是由专业软件开发人员设计的。一般在日常生活中遇到的需要用计算机处理的大多数问题都可以使用现成的应用软件完成，但有时我们仍需为遇到的具体问题自行开发软件。如某些大型计算、工程应用等，使用通用软件可能无法完成任务。这种情况下，自行编写完成指定功能的软件非常有必要。

目前，计算机程序设计语言的种类非常多，各有特点。有的适合数据库开发、有的适合科学计算、有的适合图形制作等。在本书中，将为读者介绍C语言。

1.1 程序与程序设计语言

一个应用软件是由哪些内容构成的？软件=程序+数据+文档。软件最重要的一部分就是程序，那么程序是什么？

1.1.1 程序的基本概念

为了让计算机能够贯彻执行人的意图，需要人能与计算机“沟通”，而“沟通”所需要的语言就是程序设计语言，使用程序设计语言设计的指令的集合就是程序。

程序是使用程序设计语言解决某一问题的解题步骤，是符合一定语法规则的语句和指令的集合。

人们借助程序设计语言告诉计算机要处理的原始数据、什么步骤来处理、以什么样的形式输出，这个过程就是程序设计。程序设计的过程一般由4个步骤组成。

(1) 分析问题：在解决问题前，应充分分析要解决的问解，明确需要处理的数据是什么，怎样对数据进行处理，以及最后输出的结果的数据及形式等。

(2) 设计算法：算法是为了解决一个问题所采用的方法与步骤。为解决一个问题所采用的算法不是唯一的。程序员需要设计一个最适合的算法，然后设计算法的总体规划，之后自顶向下，逐步细化过程，最终把抽象的问题具体化为可以用程序语句表达的算法。

(3) 编码：利用某种程序设计语言实现算法的过程称为编码。

(4) 程序调试：编码步骤完成后，程序要进行调试，调试包括编译和连接等操作。编译是对源程序进行语法检查的过程，程序员根据编译过程中的出错提示信息，修改源程序，并重新编译，直到没有语法错误为止，编译程序会将源程序编译成目标文件，大多数程序设计语言往往还要使用连接程序把目标程序系统提供的库文件连接形成可执行文件，在连接过程中由于函数名不正确等，也会引起连接错误。连接成功后的文件才会顺利执行。程序员需要对程序执行结果进行分析，只有结果正确的程序才是正确的程序。

1.1.2 程序设计语言

程序设计语言一般可分为机器语言、汇编语言和高级语言三大类。

(1) 机器语言：面向某种特定机器的语言，以二进制代码表示的指令集合，是唯一的计算机能直接识别并执行的语言。机器语言的优点是占用内存少，执行效率高。但由于它是面向机器的语言，不具备可移植性和通用性，非常难于记忆和识别，所以人们很少用机器语言编程。

(2) 汇编语言：用助记符来表示机器指令的语言，也称为符号语言。汇编语言较机器语言容易读写、记忆与维护，同时它也具备机器语言的全部优点，如：执行速度快、占用内存少、可直接访问和控制计算机的各种硬件设备等。但它仍然是面向机器的语言，所以不具备通用性和可移植性。

(3) 高级语言：是最接近于人类自然语言的语言，又称作算法语言，是面向问题、实现算法的语言。用高级语言编写的源程序短小精炼、便于阅读、易于查找错误和修改。高

级语言容易学习、具有可移植性。但用高级语言编写的程序，计算机不能直接识别和执行，所以需要编译程序对高级语言程序进行编译、连接后才可执行，但是高级语言编译生成的目标代码比汇编语言的程序代码要长，执行速度也要慢一些。

高级语言一般可分为结构化程序设计语言和面向对象的程序设计语言两大类。

① 结构化程序设计语言：结构化程序设计的中心是模块化。其过程针对要开发的软件采用“自顶向下，逐步分解”的方法，将其划分为若干个相互独立的模块。每一个模块实现相对独立的功能，由于每一个模块相对独立，在设计时不会受其他模块设计的影响，因而将一个复杂的大型软件的设计转变为小型的、简单的模块设计。结构化程序设计规定所有程序都是由“顺序结构、分支结构、循环结构”这三个基本结构之一或组合形成的。

结构化程序规定每一个结构都只能具有唯一的入口和出口，程序不会形成死循环，所以结构化程序整体思路清楚，易于诊断错误及维护。

常见的结构化程序设计语言有：C 语言、Fortran 语言、Pascal 语言等。

② 面向对象的程序设计语言：面向对象的程序设计语言在 20 世纪 90 年代以后才兴起，其代表语言有：C++、Java 及 small Talk 等。

结构化程序设计是一种面向解题过程的编程思想，它需要程序设计人员把重点放在设计解题步骤和过程上。而面向对象的程序设计思想则是人们把对现实社会中的现实对象的思维方式映射到编程思想中。

现实世界中各种实体可称为对象，程序就是要解决人们在现实世界中的问题。面向对象的编程就是针对现实事物(对象)设计程序，这样的编程是非常直观的。所以面向对象的程序设计思想是人们分析、设计和实现一个系统的方法尽可能接近人们认识现实世界的思维方式。

面向对象的程序设计中把数据以及对数据的操作看成一个整体，称之为对象。而对象又是某个类的实例，所有的类通过继承关系、消息传递构成一个系统。

面向对象程序设计不是完全拒绝结构化程序设计思想。在面向对象的程序编制中也要采用结构化程序设计的思想来解决问题。

面向对象程序设计的基本要素有抽象、封装、继承、多态等。

1.2 算　　法

要想解决某个问题，首先确定解决该问题的方案，如一个厨师学一种新菜式，要照着菜谱来做一样，菜谱就是该菜式的解决方案。一个程序的解决方案就是该程序的算法。

1.2.1 算法的概念

算法在程序设计中是非常重要的，瑞士计算机科学家、Pascal 之父——尼克劳斯·威茨提出了程序定义的著名公式：

算法+数据结构=程序

这个公式表明了算法和数据结构对程序的重要性。那么什么是算法呢？算法指解题方案的准确而完整的描述，是一系列解决问题的清晰指令。通俗地说就是解决一个问题的方

法和步骤，一个算法应当具有以下几个方面的特点。

(1) 有穷性。

算法的有穷性是指一个算法必须在执行有限个步骤后终止。

(2) 确切性。

算法的每一个步骤必须有确切的定义。

(3) 有 0 个或多个输入。

一个完整的程序应该包含数据输入、数据处理、数据输出三个部分，要对数据进行加工必须有原始数据，这个原始数据可以由用户输入，也可以在程序中获得。

(4) 有一个或多个输出。

数据处理后的结果必须输出，用户才可以分析该程序是否正确。

(5) 可行性。

算法的每一步都必须是可以执行的，比如除零操作就是不可执行的。

1.2.2 算法的描述方法

描述算法的方法有很多，如：自然语言描述算法、流程图、伪代码等。但这些只是描述算法的工具并不能被计算机所执行，还得用某种计算机语言在计算机上实现，这就是程序。本节将介绍常见的算法描述方法。

下面将用一个例题来说明这几种算法的描述方法。

【例 1】 输入若干个学生的成绩，以-1 作为输入的结束，计算平均成绩。

(1) 自然语言。

自然语言就是人们在日常生活中使用的语言，用自然语言描述算法，通俗易懂、容易掌握，同时也便于用户之间交流。但自然语言的表达往往不太严谨，有时在描述时可能会出现歧义，因此，自然语言表述算法往往只用于描述简单的问题。下面我们试着用自然语言描述一下例 1 的算法：

S1：设 average 为平均值，因为人数不确定，所以设置一个变量 n 统计人数
S2：输入一个成绩 score
S3：判断 score 是否等于-1，如果等于-1，则转到 S5，否则执行 S4
S4：将 score 加到变量 average 中，同时，令统计人数的变量 n 加 1，转向 S2 继续执行
S5：将累加计算得到的成绩和 average 除以人数 n，得到平均成绩
S6：输出平均成绩

从上例可以看出，自然语言描述算法虽然便于理解，但书写麻烦，而且对复杂的问题难以表述准确。

(2) 流程图。

流程图兴起于二十世纪五六十年代，它用规定的图形、连线及文字描述算法，较之自然语言，流程图形象、直观，容易掌握。因而现在已经成为程序设计及算法描述的必用工具。常用流程图标准化符号如图 1-1 所示。

仍以例 1 为例画出其流程图，如图 1-2 所示。

相对于自然语言描述算法，流程图更清楚，而且每一步非常明确，不会产生歧义。

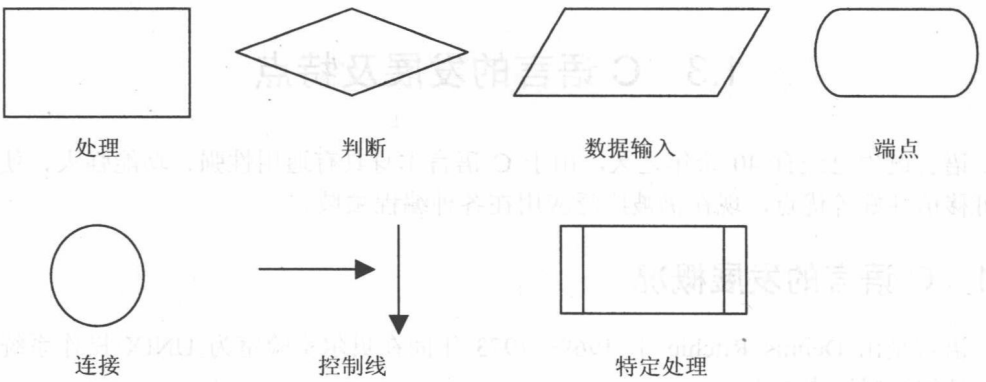

图 1-1　常用的流程图标准化符号

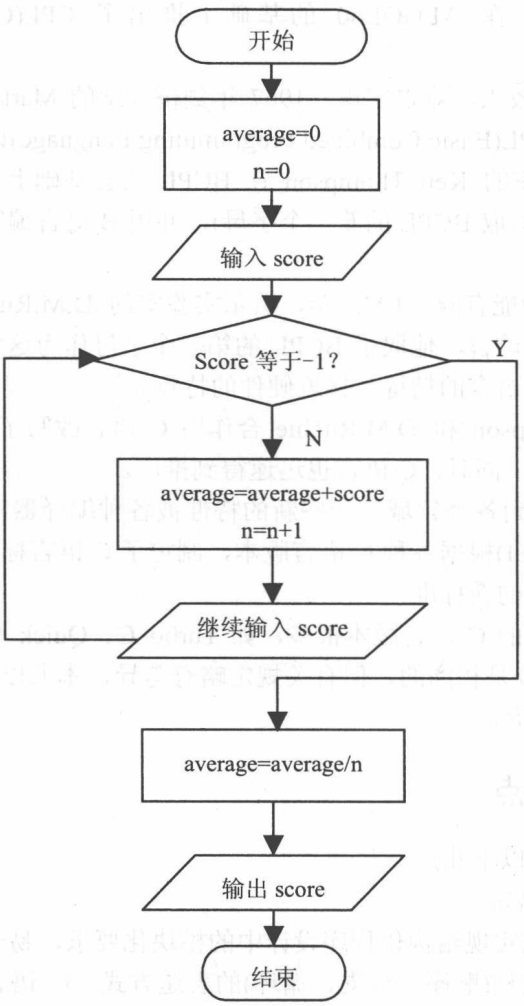

图 1-2　例 1 算法的流程图

1.3 C语言的发展及特点

C语言诞生已经有40余年之久，由于C语言本身具有通用性强、功能强大、使用灵活、可移植性好等优点，现在仍被广泛应用在各种编程领域。

1.3.1 C语言的发展概况

C语言是由Dennis Ritchie于1969—1973年间在贝尔实验室为UNIX操作系统开发的。其具体发展历史如下。

C语言的原型是1960年出现的一种面向问题的过程式高级语言ALGOL60。

1963年，剑桥大学在ALGOL60的基础上推出了CPL(Combined Programming Language)语言。

由于CPL语言规模较大，难以实现，1967年剑桥大学的Martin Richards将CPL语言简化，进而形成了BCPL(Basic Combined Programming Language)语言。

1970年，贝尔实验室的Ken Thompson在BCPL语言基础上再进行简化，设计出简单且很接近硬件的B语言(取BCPL的第一个字母)，并用B语言编写了第一个UNIX操作系统。

B语言过于简单且功能有限，1972年，贝尔实验室的D.M.Ritchie在B语言的基础上最终设计出了一种新的语言，他取了BCPL的第二个字母作为这种语言的名字，这就是C语言。C语言保留了B语言的精炼、接近硬件的特点。

1973年，Ken Thompson和D.M.Ritchie合作用C语言改写了UNIX系统，1977年后，UNIX得到广泛应用，同时，C语言也迅速得到推广。

随着C语言被推广到各个领域，一些新的特性被各种编译器实现并添加进来，1983年，美国标准化协会(ANSI)根据各种C语言版本，制定了C语言标准，称为ANSI C。后来ANSI又制定了C语言的新标准。

目前在微型机上使用的C语言版本很多，如Turbo C、Quick C、Visual C++等。这些C编译系统的基本部分是相同的，但有关规定略有差异，本书以Microsoft Visual C++ 6.0环境对C语言进行介绍。

1.3.2 C语言的特点

C语言的主要特点有以下几点。

(1) C语言是结构化语言。

C语言以函数的形式实现结构化程序设计中的模块化要求，易于实现程序间的共享。C语言提供了三种基本结构(顺序、分支、循环)的表达方式。C语言结构清晰，易于设计和维护。

(2) C语言的运算符丰富。

C语言提供了34个运算符，把括号、赋值都当成运算符处理，而其中的很多运算，如逗号、指针、求字节等也是C语言所特有的。

(3) 语言简洁、紧凑、程序书写灵活。

C 语言共有 32 个关键字、9 种控制语句，程序书写非常自由，减少了对程序员的束缚。但同时，由于其灵活、语法定义不严格，使得很多初学者掌控 C 语言比较困难。

(4) C 语言可以直接操纵硬件。

通常有一种说法是：C 语言是中级语言。因为与高级语言相比，C 语言接近硬件，而与低级语言相比，C 语言又具有面向用户、容易书写和理解等特点，因而很多大型软件都是用 C 语言编写，功能非常强大。

(5) C 语言的可移植性好。

C 语言本身不依赖于计算机硬件和操作系统，因此，C 语言编写的代码可以适用各种机型及操作系统，如 DOS、UNIX、Windows 等。

1.4　C 语言程序的基本结构

1.4.1　简单的 C 语言程序示例

虽然 C 语言程序书写灵活，但程序设计人员必须按照 C 语言规定的格式进行书写。本节将以 3 个小程序为例，简单分析 C 语言程序书写的特点。

【例 2】 编写一个程序，输出指定的信息。程序如下：

```
#include <stdio.h>
main()
{
    printf("This is the first C program!\n");
}
```

程序运行结果：

```
This is the first C program!
```

程序的第一行为一个编译预处理命令，表示包含标准输入输出头文件 stdio.h。

第二行的 main()是 C 语言的主函数，每个 C 语言程序必须有且仅有一个 main()函数。main()函数之后跟着一对大括号{…}，大括号中为函数体。函数体中包含了函数需要执行的语句，以实现函数的功能。这个 main()函数中只有一条语句，调用 printf()函数输出指定的信息，printf()函数为标准输入输出头文件 stdio.h 中的函数，所以程序第一行#include <stdio.h>的意思是告知编译器，本程序需要使用 stdio.h 中的部分函数。所以在程序书写时，只要用到输入输出这些函数，就应该在程序上方包含输入输出头文件，不包含这个头文件而使用这些函数是一个非常不好的编程习惯，会导致程序不可移植。

【例 3】 输入一个任意整数，输出其立方值。程序如下：

```
#include <stdio.h>
main()
{
    int x,s;                    /*定义变量*/
    scanf("%d",&x);             /*调用 scanf()函数输入 x 的值*/
    s=x*x*x;                    /*计算*/
    printf("x*x*x=%d\n",s);/*输出结果*/
}
```

程序运行结果：

当从键盘上输入5时，屏幕上输出：

```
x*x*x=125
```

本程序的主函数体是由多条语句组成的。main()函数体中的第一行定义了两个整型变量x和s，x用来接收输入的数据，s则用来计算x的立方值。第二行调用scanf()函数输入x的值，scanf()函数同样包含在stdio.h头文件中。第三行为数据计算，计算x的立方值，并将其赋给变量s。第四行为结果的输出。每行后面/*…*/为注释部分，注释部分不会被执行。

【例4】 输入两个两位数，如a=34，b=56，取其各自的个位数组成一个新数c，规则为取a的个位数当做c的十位数，取b的个位数当做c的个位数结果，则c=46。程序如下：

```
#include<stdio.h>
int fun(int a,int b)
{
    int m;
    m=a%10*10+b%10;
    return m;
}

main()
{
    int a,b,c;
    printf("please input a & b:\n");
    scanf("%d%d",&a,&b);
    c=fun(a,b);
    printf("c=%d\n",c);
}
```

当输入45 76时，程序运行结果：

```
c=56
```

这个实例是由两个函数main()、fun()组成的，主函数完成数据输入及结果输出部分，而fun()函数实现对数据处理的功能。fun()函数的具体功能为：分别拆出两个数a、b的个位，并将它们组合成一个新的数返回main()函数。

1.4.2 C语言程序基本结构

根据上节的三个例题来分析C程序的基本结构。

(1) C语言的程序是由函数组成的。

C语言的程序是由函数组成的，每个程序中有且仅有一个main()函数，在稍大的程序中，往往把整个程序分成若干个函数来完成，以实现模块化功能。不管程序文件中有多少个函数，程序的执行永远从main()函数处开始，以main()函数的结束为结束。在执行过程中，main()函数可以调用其他函数，函数之间也可以相互调用。

每个C语言函数都由函数首部和函数体构成。例如上一小节例4中的fun函数中，int fun(int a,int b)为函数首部，函数首部代表定义一个函数的开始，其中包括函数类型、函数名、函数形参等。