浙江省“十一五”重点建设教材

高等院校信息技术应用型规划教材

C语言程序设计

江宝钏 主 编
陈叶芳 贾晓雯 裘姝平 副主编

清华大学出版社

浙江省“十一五”重点建设教材

高等院校信息技术应用型规划教材

C语言程序设计

江宝钏 主 编
陈叶芳 贾晓雯 裘姝平 副主编

清华大学出版社
北 京

内 容 简 介

本书以实例程序和知识点相结合的方式组织内容，通过示例程序来引入知识点。全书共分 10 章，包括 C 语言程序设计概述，基本数据类型和表达式，顺序结构程序设计，选择结构程序设计，循环，函数，数组，指针，结构体与共用体，文件。本书每章都有详细的程序范例和运行结果。

本书适合作为各类高等院校计算机专业及理工类非计算机专业的 C 语言程序设计课程的教材，也可作为程序设计爱好者的自学用书。

图书在版编目（CIP）数据

C 语言程序设计/江宝钏主编. --北京：清华大学出版社，2015
高等院校信息技术应用型规划教材
ISBN 978-7-302-39009-1

Ⅰ. ①C… Ⅱ. ①江… 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(2015)第 013572 号

责任编辑：孟毅新
封面设计：傅瑞学
责任校对：刘　静
责任印制：王静怡

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，010-62795764
印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：17.5　　**字　　数**：419 千字
版　　次：2015 年 3 月第 1 版　　**印　　次**：2015 年 3 月第 1 次印刷
印　　数：1～2500
定　　价：36.00 元

产品编号：063056-01

前　言

C语言程序设计是高校计算机专业和理工类非计算机专业重要的计算机基础的必修课程，一般是作为计算机程序设计的第一门语言课程。通过本课程的学习，学生能够学会提出问题、解决问题并提高综合分析问题的能力；还可掌握程序设计的思想和方法，提高实际编程能力。

本书采取的是以程序案例为驱动方式和以知识点为主线相结合的编排方式，以循序渐进的方法，从编程主线、知识点和应用面3个层次逐步提高学生的编程能力。

教材内容按照“示例问题的提出与解决→引出相关知识→归纳总结→举例说明→修改与扩展→实践训练”来编排。重点讲解与分析程序设计的过程，辅以语言知识的介绍。

本书淡化了C语言的复杂语法知识和数值计算方面的应用，举例以通俗易懂、贴近生活和富有趣味性为主。在讲解C语言语法知识的过程中，给出重要的知识点和经典的例题分析，并让学生对例题进行修改和扩展，指导学生如何分析问题、组织数据、设计算法和编写程序以及调试、运行与测试，强调学生编程能力的培养。对于难度较大或复杂的语法叙述的内容作“*”标注，以便教师根据学生情况和教学时数作适当调整，具有较大的灵活性，为差异性教学做好教材方面的准备工作，以便学生能较自主地选择想学并且能够学的东西，能够为学生在一个更高的层次上运用计算机解决专业问题打下基础。

本书具有如下特色。

(1) 每章开头都设有内容要点提示，并通过问题的提出与示例程序分析来引入内容，让学生知道为什么要学习本章内容，通过示例程序分析对本章的内容有个大概的了解，先把学生引进门。

(2) 尽量淡化语法规则和复杂的叙述。对于前面4章，语法规则和输入/输出仅作较简单叙述，考虑首先不能让学生陷入烦琐的语法规则中，如自增、自减问题，不能让学生一开始学习就产生畏难情绪。

(3) 让学生多模仿和修改程序。在编程的开始阶段有意识地编排一些简单程序让学生模仿并修改原来的程序和数据，尽量让学生尽快进入编程状态，让学生有一定的成功感，从而获得进一步学下去的愿望和动力。与本书配套的实验教材中还将加强示例程序的调试和分析，示例程序的模仿和修改。

(4) 给出“示例程序运行结果截屏图”。对每个示例程序都给出“运行结果截屏图”，不同的输入参数，不同的输出结果，这样让学生感觉有真实的运行结果。大部分示例程序提出了相关的思考或问题，改变一些参数、条件或扩展一些解题思路，让学生试一试。教材中所有的例题都在Visual C++ 6.0中上机调试通过，可直接下载使用，学生在此基础上直接修改即可。

(5) 单独设有“提高部分”。考虑到学生的差异性和教学时数的限制，在多数章的最后一节设“深入探讨”或作“*”标注，放入一些相对复杂的内容。前面几章主要是一些复杂的、不太常用的语法规则，后面几章是一些不太常用或较难学的内容。学生在开始学习阶段完

全可以抛开这些细节，教师可以根据学生的情况作灵活的调整。

本书由宁波大学江宝钏担任主编，其他编者都是具有丰富计算机程序设计教学经验的一线教师。第1～3章、第7章、第9章、附录由江宝钏编写，第8章由宁波大学陈叶芳编写，第5章由浙江传媒学院贾晓雯编写，第6章由浙江传媒学院裘姝平编写，第4章和第10章由台州学院王爱冬和陈金飚编写。

本书得到了浙江省重点教材建设项目经费的资助，杨任尔、宋宝安、李纲、董一鸿、石守东等多位任课老师对本书的大纲和内容提出了许多修改意见和建议，在此表示衷心的感谢。

本书经过多次修改讨论而定稿，限于编者的水平，书中难免存在不足之处，恳请读者批评指正，以便进一步修订。

编者联系地址：浙江省宁波大学信息科学与工程学院，邮编 315211；E-mail：jiangbaochuan@nbu.edu.cn。

编　者

2015年1月

目　录

第1章　C语言程序设计概述

内容要点提示

(1) C语言程序的基本组成结构是什么?

(2) 运行C语言程序需要哪些步骤?

要让计算机为人们完成各种各样的工作,就必须让它执行相应的程序,这些程序都是利用程序设计语言编写出来的。

所谓程序,实际上是用计算机语言描述的解决某一问题的步骤,是符合一定语法规则的有序的符号序列。人们借助程序设计语言告诉计算机要做什么以及如何做。通过在计算机上运行程序,向计算机发出一系列指令,让计算机按人们的要求解决问题。本章主要简述程序设计语言的发展概况、C语言程序的基本结构、如何运行简单的C语言程序。

1.1　程序设计语言的发展概况

不同的问题可以用不同的程序设计语言来解决,即为了解决某一个问题所编写的程序并不是唯一的,不同的程序有不同的解决方法,有不同的效率。程序设计语言的发展经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

直接使用二进制表示的指令来编程的语言即机器语言。由"0"和"1"组成的一个二进制编码,表示一条机器指令,使计算机完成一个简单的操作。用机器指令编写的程序即机器语言程序,是计算机唯一能够直接识别并执行的程序。

用机器语言编写程序,编程人员要首先熟记所用计算机的全部指令代码和代码的含义,还得记住编程过程中每步所使用的工作单元处于何种状态,这是一件十分烦琐的工作。编写程序花费的时间往往是实际运行时间的几十倍甚至几百倍。而且,编出的程序全是些0和1的指令代码,直观性差,容易出错,且不同的机型有不同的机器指令。目前很少有人直接用机器语言编程。

2. 汇编语言

汇编语言是指用比较容易识别、记忆的助记符代替特定的二进制串。通过助记符,人们较容易读懂程序,调试和维护也较为方便。但这些助记符号计算机无法识别,需要一个专门的程序将其翻译成机器语言,这种翻译程序称为汇编程序。汇编语言与机器语言性质上是一样的,只是表示方式做了改进,但可移植性与通用性仍然很差。

3. 高级语言

汇编语言和机器语言是面向机器的,同属于低级语言。高级语言是一种用能表达各种意义的"词"和"数学公式"按一定的"语法规则"编写程序的语言,也称为高级程序设计语言或算法语言,这类语言接近于人的自然语言。半个多世纪以来,有几百种高级语言问世,影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、PL/1、Pascal、C、

PROLOG、Ada、C++、Visual C++、Visual Basic、Delphi、Java 等。高级语言的发展也经历了从早期语言到结构化程序设计语言、再到面向对象程序设计语言的过程。高级语言的使用，大大提高了程序编写的效率和程序的可读性。

对于用高级语言写成的源程序，通常每一条语句对应一组机器指令，它必须经过翻译程序，翻译成机器语言指令形式才能执行。翻译程序有编译程序和解释程序两种。每种高级语言都有自己的翻译程序，用 C 语言编写的源程序，需通过 C 语言编译程序，翻译成目标程序，再通过连接程序生成可执行程序，才可在机器上运行。

C 语言是 20 世纪 70 年代初由美国贝尔实验室在 B 语言的基础上发展起来的，它保持了 B 语言精练、接近硬件的特点，又改进了 B 语言过于简单的缺点。早期的 C 语言主要是在贝尔实验室内部使用。1978 年，以 C 语言编译程序为基础，B. W. Kernighan 和 D. M. Ritchie 合著了著名的 *The C Programming Language* 一书。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础。但是，在这本书中并没有定义一个完整的标准 C 语言，后来由美国国家标准协会(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，通常称为 ANSI C。

早期的 C 语言主要是应用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识，到了 80 年代，C 语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

在 C 语言的基础上，贝尔实验室在 1983 年推出了 C++。C++ 进一步扩展和完善了 C 语言，成为一种面向对象的程序设计语言。

1.2 简单的 C 语言程序

1.2.1 简单 C 语言程序示例

先看几个简单的 C 语言程序实例，然后从中分析 C 语言程序的特点。

【例 1-1】 在屏幕上输出“Hello，World!”。

```
/* 源程序：exp1_1.cpp */
#include<stdio.h>                    /* 编译预处理命令 */
void main()                          /* 定义主函数 main(),void 表示没有函数返回值 */
{   printf("这是一个最简单的屏幕输出程序\n");      /* 输出双引号中的文字到屏幕 */
    printf("Hello,World!\n");                    /* 输出"Hello, World!"到屏幕 */
}
```

运行结果如图 1-1 所示。

图 1-1 例 1-1 程序运行结果

该程序的功能是在屏幕上输出“Hello，World!”。下面分析本程序的构成。

(1) #include <stdio.h>是编译预处理命令，其目的是使输入、输出能正常执行。

(2) main 是主函数的函数名，且只能使用这个名字。每个 C 语言源程序都必须有且有一个主函数，函数后面必须有一对圆括号。main 前面的 void 表示函数返回值为“空类型”，即执行本函数后没有函数返回值。

(3) 一对花括号内的部分称为函数体。

(4) 程序的执行总是从 main()函数的左花括号“{”开始到 main()函数的右花括号“}”结束。

(5) C 语言的关键字和特定字使用小写字母。如 main、include 是特定字，必须用小写字母。在 C 语言中是要区分大小写的。

(6) 使用 printf()函数在屏幕上输出内容。

【例 1-2】 已知圆的半径，求圆的面积。

分析：这是一个简单的数学计算问题，利用公式 πr^2 求解，其中 π 有固定的值。如果已知半径 r 的值，就可以求出圆的面积。

```
/* 源程序: exp1_2.cpp */
#include <stdio.h>
void main()
{
    float r,s;                          /* 定义变量 */
    printf("请输入圆的半径: \n");        /* 在屏幕上显示提示信息 */
    scanf("%f", &r);                    /* 从键盘输入半径值赋给变量 r */
    s=3.1415*r*r;                       /* 计算面积 */
    printf("s=%f\n", s);                /* 输出面积 */
}
```

运行结果如图 1-2 所示。

图 1-2　例 1-2 程序运行结果

为了能执行输入函数 scanf()和输出函数 printf()，必须在程序中包含 #include <stdio.h>这条语句，一般是放在程序的开头。

程序由一个主函数 main()组成。花括号内的函数体部分由 5 条语句组成。

“float r,s;”是变量定义语句。变量是内存中的存储单元，能够存储程序使用的数据，变量必须先定义后使用。

“scanf("%f",&r);”语句接收用户从键盘上输入圆的半径并赋给变量 r。“&”是地址符号，不能省略。

“s=3.1415*r*r;”语句利用 r 计算圆面积，并把结果存放到变量 s 中。C 语言中“*”表示数学中的乘号。

scanf()、printf()是 C 语言中最常用的输入/输出函数，用来输入/输出数据。

“/*...*/”是注释，不是程序部分，在程序执行中不起任何作用，只为增加程序的可读性。

【例 1-3】 输入两个整数，计算并在屏幕上输出他们的和。

如何输入两个整数到变量中？这里是通过键盘输入。

```
/* 源程序: exp1_3.cpp */
//这是一个计算两数之和的程序
#include <stdio.h>
void main()
{   int a,b,sum;                        //变量定义
    printf("请输入两个整数:\n");        //提示用户输入数据
    scanf("%d%d",&a,&b);                //调用输入函数,要求用户从键盘输入两个整数
```

```
    sum=a+b;
    printf("sum=%d\n",sum);          //输出和值
}
```

运行结果如图 1-3 所示。

本程序的作用是求两个整数之和。

图 1-3 例 1-3 程序运行结果

“int a,b,sum”是声明部分,声明变量 a、b 和 sum 为整型(int)变量。

printf()函数在屏幕上输出提示信息。

scanf()函数接收用户从键盘输入两个整型数据。

“sum=a+b;”语句把 a、b 两个变量的值求和后赋给变量 sum。

最后是在屏幕上输出和值。

注释也可用//。程序中从//开始到行尾都为注释文字。注释对程序的编译和运行不起作用,注释可以放在程序中的任何位置。

1.2.2 C 语言程序的组成结构

从上面的几个例子中可以看出,C 语言程序由编译预处理命令、函数及注释三部分组成。

一个完整的 C 语言程序可以由一个或多个函数组成,其中主函数 main()必不可少,且只能是唯一的。C 语言程序总是从主函数 main()开始执行,与 main()函数在整个程序中的位置无关。

C 语言程序的较完整的结构形式如下。

```
#include <stdio.h>                    /* 编译预处理命令 */
void main()
{
    定义部分                          /* 这是程序的定义部分 */
    语句序列                          //这是程序的语句部分
}
f1() {
    定义部分
    语句序列
}
f2() {
    定义部分
    语句序列
}
…
fn() {
    定义部分
    语句序列
}
```

(1) 编译预处理命令。程序中每个以“#”号开头的命令行都是编译预处理命令,用于包含头文件,一般放在程序的最前面。头文件的扩展名一般为.h,Visual C++ 6.0 提供了多个头文件。不同的编译预处理命令完成不同功能,包含了各类标准函数的原型说明。如#include<stdio.h>命令的作用是将特定目录下的 stdio.h 文件,嵌入到源程序中,输入/

输出函数必须用到这个头文件。

(2) 函数。

① 函数首行，即函数的第一行，如 void main()，包括函数类型、函数名、函数参数、参数类型等。

② 函数体，是函数首行下面花括号对中的内容。函数体由各类语句组成，执行时按语句的先后次序依次执行，各语句间用分号";"结束。f1()～fn()代表用户定义的函数。

(3) 注释。注释在程序执行时不起任何作用，只是增加程序可读性，方便别人的阅读或自己以后的回顾。C 语言的两种注释方法。

①"/* 注释内容 */"：适用于注释多行，"/*"和"*/"之间的内容即为注释。

②"//注释内容"：适用于注释单行，"//"后面的部分(行)即为注释。

其中"注释内容"可以是汉字或英文字符。

1.2.3 标识符与关键字

前面例子出现了很多字符组合，它们分别为关键字、函数库中预定义的标识符和用户自定义的标识符。

标识符是程序中实体如变量、常量、函数、数组、结构体以及文件等的名称。C 语言中的标识符分为系统定义标识符和用户定义标识符。

1. 系统定义的标识符

系统定义的标识符是指具有固定名称和特定含义的标识符，如 int、for、while 等。系统定义的标识符分为关键字和预定义标识符两种类型。

2. 关键字

关键字又称保留字，是 C 语言规定的具有特定含义的标识符。关键字必须用小写字母表示。C 语言共有 32 个关键字，分为以下几类。

(1) 数据类型

```
int  long  short  char  float  double  singed  unsigned  struct
union  enum  void  volatile  const  typedf
```

(2) 存储类别

```
auto  static  register  extern
```

(3) 语句命令字

```
goto  return  break  continue  if  else  while  do  for  switch  case  default
```

(4) 运算符

```
sizeof
```

3. 预定义标识符

预定义标识符也是一类具有特殊含义的标识符，包括系统标准库函数名和编译预处理命令。系统允许用户对预定义标识符重新定义，但这将使这些标识符失去系统规定的原意。例如，"int include=5"中的 include 不再表示文件包含命令，而是作为一个变量名使用。因此为了避免误解，建议不要将这些预定义标识符另作他用。