高等学校计算机基础教育规划教材

C语言程序设计教程

张玉春 孙大元 主编

清华大学出版社

高等学校计算机基础教育规划教材

C语言程序设计教程

张玉春 孙大元 主编

孙元 黄玥 李晓峰 刘通 赵永华 副主编

清华大学出版社

北京

内 容 简 介

本书强调实用性，注重教材的理论与实践相结合，以培养学生程序设计的基本方法和基本技能为目标。全书分为11章，主要内容包括C语言与程序设计、基本类型数据及其运算、选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体与共用体、文件、位运算和C语言应用。本书以程序设计为中心，语法介绍精练，内容叙述深入浅出、循序渐进，程序案例生动易懂，具有很好的启发性，并且，每章均配有教学课件和精心设计的习题。

本书既可以作为高等学校本科及专科学生C语言程序设计的教材，又可以作为自学者的参考用书，同时可供各类考试人员复习参考。

对于本书所配电子教案及相关教学资源，用户可以从清华大学出版社教学资源网下载。使用本书的学校也可以与编者联系，索取更多相关教学资源。

图书在版编目(CIP)数据

C语言程序设计教程/张玉春，孙大元主编. --北京：清华大学出版社，2013
高等学校计算机基础教育规划教材
ISBN 978-7-302-32597-0

Ⅰ. ①C… Ⅱ. ①张… ②孙… 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(2013)第117682号

责任编辑：袁勤勇　王冰飞
封面设计：傅瑞学
责任校对：时翠兰
责任印制：李红英

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，010-62795954
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：17　　**字　　数**：395千字
版　　次：2013年9月第1版　　**印　　次**：2013年9月第1次印刷
印　　数：1～3500
定　　价：29.00元

产品编号：048933-01

前言

本书是根据教育部非计算机专业计算机课程教学指导分委员会制定的《非计算机专业计算机基础课程教学基本要求》和《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求(试行)》中提出的要求编写的。本书的特点是强调实用性，注重教材的理论与实践相结合，以培养学生程序设计的基本方法和基本技能为目标。

C语言是一种结构化程序设计语言，兼有高级语言和低级语言的功能，不仅可用于编写系统软件，也可用于编写各类应用程序以及工业控制程序。目前流行的面向对象程序设计语言，如C++、Java、C#等都是在C语言的基础上发展派生而来的。通过学习C语言，学生不仅能够掌握程序设计的基本思想，也可为今后学习Java、C++、VB等语言打下良好的基础。

本书共分11章，其中包括C语言与程序设计、基本类型数据及其运算、选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体与共用体、文件、位运算和C语言应用。

本书整体结构编排合理，大部分教学内容采用例题的形式进行组织，并对例题进行分析，有助于学生对知识的理解与掌握。通过本书的学习，学生能够掌握程序设计的基本思想和常见简单问题的算法，并可以编写程序加以实现。本书还介绍了C语言在硬件方面和在软件方面的应用，使学生能够了解C语言的用途，增加学习C语言的兴趣。

参加本书编写的教师及编写内容如下表所示：

编写内容	编者姓名	编写内容	编者姓名
第1章　C语言与程序设计	孙　元	第8章　结构体与共用体	张玉春
第2章　基本类型数据及其运算	张玉春	第9章　文件	刘　通
第3章　选择结构程序设计	李晓峰	第10章　位运算	赵永华
第4章　循环结构程序设计	李晓峰	第11章中的11.1节、11.2节	刘　通
第5章　数组	黄　玥	第11章中的11.3节	黄　玥
第6章　函数	孙　元	附录A、附录B、附录C	张玉春
第7章　指针	孙大元	附录D	孙　元

本书在编写过程中得到吉林大学公共计算机教学与研究中心领导的大力支持，在此表示感谢。本书在出版过程中得到清华大学出版社袁勤勇编辑的大力支持，在此表示感谢。本书也是所有参编教师辛勤努力的结果，在此向他们表示感谢。

由于编者水平有限，书中难免存在疏漏与不足之处，敬请读者指正。为方便教师的教学工作和读者的学习，本书有配套的源程序代码、习题答案和电子教案，需要者可通过出版社与编者联系获取。

编　者

2013 年 5 月

目录

第1章

C语言与程序设计

计算机的工作原理是执行程序。所谓程序，就是一组计算机能够识别和执行的指令。每个指令都具有不同的含义，计算机能够有效地区分并执行。程序设计就是给出解决特定问题程序的过程，也就是编写程序。程序设计往往以某种程序设计语言为工具，设计出这种语言环境下的程序。

1.1 程序设计语言及其发展

程序设计语言也称为计算机语言，它是人和计算机进行交流的语言，是用于书写计算机程序的语言。程序，其实就是把需要做的事情用程序设计语言描述出来，然后在计算机上执行，以此来解决实际问题。

自从世界上第一台计算机诞生以来，程序设计语言就不断发展。特别是近十几年来，随着计算机软、硬件的飞速发展，程序设计语言更加完善，种类也越来越多。目前，有多种不同类型的程序设计语言，主要分为低级语言和高级语言两大类。目前流行的程序中以面向对象的程序设计方法为主。

1.1.1 程序设计语言的发展历程

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

机器语言是直接用二进制代码指令表达的计算机语言，可以用0和1组成的一串代码表示指令，该串代码有一定的位数且分成若干段，每段代码表示不同的含义。目前，计算机全称仍然是“电子计算机”。由于电子元器件有“开”和“关”两种稳定的工作状态，所以从物理上决定了目前电子计算机采用二进制进行运算，因此计算机不能识别人所识别的文字，而只能识别机器语言，即由0和1构成的代码。例如，某台计算机的字长为16位，即一条指令或某个信息由16个二进制数组成。16个0和1可组成多种排列组合，从而形成计算机可以识别的不同操作。对于人来说，机器语言难以记忆和识别，出错的时

候也难以修改。

2. 汇编语言

在汇编语言中，用助记符代替操作码，用地址标号或符号代替地址。用符号代替二进制编码，可以把二进制编码的机器语言变成汇编语言，即汇编语言实际上就是机器语言的符号化。例如，指令“ADD”代表加，指令“ MOV”代表数据传送等。汇编程序的每一个指令都对应一个实际操作，类似的符号对于编程的人来说比机器语言更易懂，同时维护更方便。尽管如此，一般的汇编源程序还是相对冗长、复杂、易出错。由于汇编语言可以直接对硬件进行操作，所以其源程序经汇编生成的可执行文件不仅小，而且执行速度很快，缺点是由于与硬件紧密相关，所以可移植性差。

机器语言和汇编语言都是面向机器的语言，由于它们“贴近”计算机，所以被称为低级语言。低级语言与特定的机器有关，功效高，但使用复杂、烦琐、费时、易出错。

3. 高级语言

为了克服低级语言的缺点，20 世纪 50 年代中期出现了高级语言。高级语言主要是相对于低级语言而言的，这种语言接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，其特点是在一定程度上与具体机器无关，易学、易用、易维护。

1954 年，第一个高级语言——Fortran 问世了，到目前为止，共有几百种高级语言出现，其中，影响较大的是 Fortran、Algol、Cobol、Basic、Lisp、Pascal、C、Prolog、C++ 、VC、VB、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，从面向过程到面向对象程序语言的过程。20 世纪 60 年代中后期，软件的需求越来越多，软件的规模越来越大，但由于缺乏科学规范的系统规划、测试与评估标准，往往耗费巨资编写出来的软件系统，由于含有错误而无法使用，甚至带来巨大的损失。为了解决和避免这种情况，1969 年提出了结构化程序设计方法，1970 年，第一个结构化程序设计语言——Pascal 语言出现，标志着结构化程序设计时期的开始。在面向过程的程序设计方法中，数据与数据的处理过程是分开的，把处理过程按功能组成一个个独立的模块，在对数据进行处理的过程中，再分别调用各个独立的模块来实现功能。但是随着模块的程度越来越复杂，面向过程程序设计的缺点暴露出来，例如生产率低下、代码重用程度低、软件仍然很难维护等。针对这些缺点，面向对象的程序设计方法应运而生。面向对象的程序设计方法把数据和处理数据的过程当作一个整体，即对象。在分析过程中，把系统分解成一个个对象，同时把数据和相应数据的处理过程封装在对象中。在此之前的高级语言，几乎都是面向过程的，而 C++ 、VC、VB、Delphi、Java 等是典型的面向对象程序设计语言。

1.1.2 程序处理方式

计算机不能直接识别和执行用汇编语言或高级语言编写的程序，必须通过“翻译程序”将程序翻译成机器语言形式的目标程序，计算机才能识别和执行。这种“翻译”通常有

两种方式，即解释方式和编译方式。

1. 解释方式

解释方式是将程序的每条语句一边翻译一边执行，即程序一边由相应语言的解释器“翻译”成目标代码（即计算机可以识别的机器语言），一边执行。此类方式的典型程序语言是 Basic。这种翻译方式较灵活，可以动态地调整、修改程序。但该方式没有对整个程序优化的过程，所以效率较低；而且不能生成独立的可执行文件，即程序不能脱离其解释器。

2. 编译方式

编译方式是将程序源代码“翻译”成目标代码（二进制），再经过连接程序连接，形成可执行文件。可执行文件可以脱离其语言环境独立执行，因此和解释类语言相比，其使用比较方便、可移植性好、效率较高。但如果程序需要修改，那么必须先修改源代码，再经过编译、连接生成新的可执行文件才能执行。大多数高级语言程序都是编译型的，例如 C、Fortran、Pascal 等。这类高级语言是面向过程的，即程序设计语言都属于过程化语言，在编写程序时需要具体指定每一个过程的细节。在编写规模较小的程序时，使用这些编程语言比较方便，但在处理规模较大的程序时，就显得很复杂。

1.2 程序的基本结构及其表示

程序的基本结构主要有顺序结构、选择（也可称为分支）结构和循环结构 3 种。数学证明，这 3 种结构可以组成所有程序结构。

顺序结构就是按照程序语句出现的先后顺序一步一步进行。如图 1-1 所示，当执行完成 A 所指定的操作之后，顺序执行 B 指定的操作。

选择结构则需要条件判断，当条件成立才会执行相应条件下的语句，如果条件都不成立，则执行其他的语句或什么也不执行。如图 1-2 所示，当条件 P 成立时，执行 A 操作；如果条件 P 不成立，则执行 B 操作。如图 1-3 所示，当条件 P 成立时，执行 A 操作；如果条件 P 不成立，则什么操作也不执行。

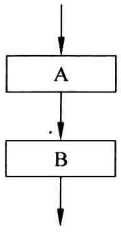

图 1-1 顺序结构

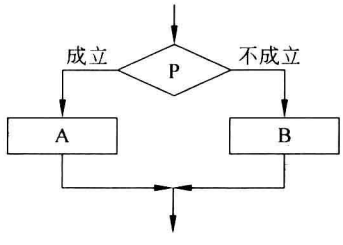

图 1-2 选择结构(1)

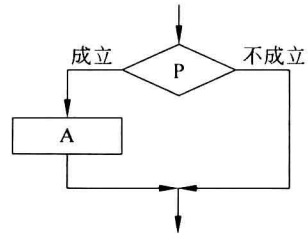

图 1-3 选择结构(2)

循环结构是在条件成立的前提下不断重复执行相同的语句，直至条件不成立为止，其循环分为直到型和当型两种类型。图 1-4 所示为直到型循环，首先执行 A 操作，然后判断条件 P 是否成立，如果不成立则结束循环。如果成立则继续执行 A 操作，循环往复，直到条件 P 不成立则结束循环；如图 1-5 所示为当型循环，首先判断条件 P 是否成立，如果不成立，则不执行任何操作；如果条件 P 成立，则执行 A 操作，循环往复，直至条件 P 不成立则结束循环。

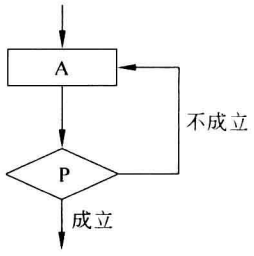

图 1-4　直到型循环

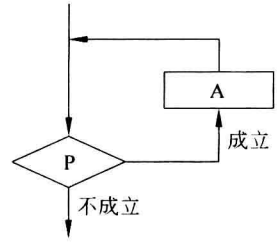

图 1-5　当型循环

1.3　C 语言概述

C 语言是一门通用的、模块化的程序设计语言，可以用于系统软件和应用软件的开发。由于 C 语言本身具有的优势，使得其应用广泛，所以不同的公司有各自不同的 C 语言版本，在 C 标准出台之后，各个 C 语言版本的特点及功能相对一致，使得用户应用起来更为方便、快捷。

1.3.1　C 语言发展简史

C 语言是在 20 世纪 70 年代初问世的，到 80 年代，C 语言被广泛应用。随着微型计算机以及高级语言编程的日益普及，出现了许多 C 语言版本。由于没有统一的标准，这些 C 语言之间或多或少地出现了不一致的地方。为了统一 C 语言版本，1983 年，美国国家标准局(American National Standards Institute，ANSI)成立了一个委员会，来制定 C 语言标准。1989 年，C 语言标准被批准，这个版本的 C 语言标准通常被称为 ANSI C。目前，几乎所有的开发工具都支持 ANSI C 标准，它是 C 语言用得最广泛的一个版本。1990 年，国际标准化组织(ISO) 接受了 87 ANSI C 为 ISO C 的标准。1994 年，ISO 修订了 C 语言的标准。1995 年，WG14 小组对 C 语言进行了一些修改，成为后来的 1999 年发布的 ISO/IEC 9899:1999 标准，通常被称为 C99。

目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的，但不同版本的 C 语言编译系统所实现的语言功能和语法规则会稍有差别。C 语言之所以能成为很受欢迎的语言之一，主要因为其具有强大的功能。许多著名的系统软件，例如 UNIX 操作系统等都是用 C 语言编写的。

1.3.2 C语言的特点

每一种编程语言都有它自己的特点，当然C语言也不例外，作为一种计算机高级语言，它有其独特的优点和缺点。

1. 紧凑简洁、方便灵活

C语言共有37个关键字，9种控制语句，程序主要用小写字母表示，且书写自由。C语言把低级语言的实用性与高级语言的基本结构和语句相结合，其程序比其他高级语言简洁、源程序短。

2. 数据类型丰富

C语言的数据类型有整型、浮点型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂数据类型的运算。指针概念的引入，使程序设计灵活、执行效率高。

3. 运算符丰富

C语言的运算符涵盖的范围广泛，共有34个。由于把括号、赋值、强制类型转换等都作为运算符处理，从而使C语言的运算类型极其丰富，表达式类型多样化，灵活地使用各种运算符可以实现在其他高级语言中难以实现的运算。

4. 结构化程序设计语言

C语言具有结构化程序设计语言所要求的三大基本结构，层次清晰，逻辑性强，便于维护、调试。

5. 程序设计自由度大

对绝大部分的高级语言而言，语法检查相对较严格，几乎可以检查出所有的语法错误。C语言允许程序员有较大的自由度，这样就要求程序员要仔细检查程序，以保证其正确性。这种较大的自由度给程序留下了出现一些潜在错误的可能性，降低了程序的健壮性。

6. 允许直接访问物理地址，直接对硬件进行操作

在计算机中，位(bit)是最小单位，1 bit就是一个二进制位。C语言能进行位运算，实现汇编语言的大部分功能，同时可以对硬件直接操作。由此可见，C语言虽然被认为是高级语言，但是其自身还具备很多低级语言的特征，所以也有人认为其介于高级语言和低级语言之间。

7. 程序执行效率高

在执行效率上，汇编语言仅次于机器语言，但基本接近于机器语言的效率。由于

C语言一般只比汇编程序生成的目标代码效率低10%～20%，所以其执行效率是很高的。

8. 可移植性好

就目前使用的操作系统以及各种型号的计算机而言，C语言的一个突出优点就是基本不用做修改即可使用。

总之，C语言既具有高级语言的特点，又具有低级语言的特征；既具有可移植性的特点，又能用来编写涉及底层的程序。正是由于C语言本身具有的这些特点，致使其应用更为广泛。然而C语言也有不足之处，例如对数组下标越界不做检查、对变量类型约束不严格等，这些会导致一些使用者很难及时发现的编制程序的问题。

1.3.3 简单的C程序介绍

下面是一个简单的C程序例子。

【例1-1】 在屏幕上输出“This is my first program.”。

```
#include <stdio.h>
int main()
{
    printf("This is my first program.\n");
    return 0;
}
```

程序分析：

程序的第1行#include <stdio.h>的作用是提供输入/输出库函数的有关信息，stdio即“standard input & output”的缩写。第2行main是函数名，也称为“主函数”，“int”表示函数类型，即main函数返回一个整数值。函数体即函数所要执行的语句，用大括号“{}”括起来。主函数中有一个输出语句printf，它是由标准输入/输出函数库提供的输出函数，双引号中的字符按原样输出，“\n”为转义字符，表示换行。程序运行结果是在屏幕上输出“This is my first program.”。

【例1-2】 计算两个数乘积。

```
#include<stdio.h>
int main()
{
    int mul(int x,int y);                    //对mul函数进行声明
    int i,j,k;                               //定义i、j、k3个变量
    scanf("%d,%d",&i,&j);                    //输入i、j两个变量的值
    k=mul(i,j);                              //调用mul函数，将函数返回值赋给变量k
    printf("i*j=%d\n",k);                    //输出i*j的运算结果
    return 0;
}
int mul(int x,int y)                         //定义整型函数mul，形参为整型变量x、y
{
```