C YUYAN CHENGXU SHEJI JIAOCHENG

高等教育21世纪课程教材

计算机基础教育课程体系改革教材

C语言程序设计教程（第二版）

■ 主编　龙佑喜　彭三城

湖南教育出版社

www.hneph.com

高等教育21世纪课程教材
计算机基础教育课程体系改革教材

C语言程序设计教程（第二版）

■ 主编　龙佑喜　彭三城

湖南教育出版社

图书在版编目(CIP)数据

C语言程序设计教程/龙佑喜，彭三城编著.—长沙
湖南教育出版社，2004

Ⅰ.C… Ⅱ.①龙… ②彭… 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(2004)第112904号

C语言程序设计教程

责任编辑：蒋芳

湖南教育出版社出版发行(长沙市韶山北路643号)
网　　址: http://www.hneph.com
电子邮箱:csgaojiao@163.com

湖南华商文化商务有限公司印刷

787×1092　16开　印张: 20.5　字数:492000
2004年11月第1版　　2006年1月第2版第1次印刷

ISBN7-5355-4380-4/G·4375
定价:28.00元

前　言

根据教育部非计算机专业计算机基础课程教学指导分委员会提出的高等学校计算机基础课程教学基本要求，我们组织了多名教学经验丰富的从事计算机基础教学和应用教学的专家和老师编写了《C 语言程序设计教程》一书。教材自 2004 年 11 月出版以来，已经全国多所兄弟院校使用，受到广大师生的普遍好评。在此期间，编者曾广泛听取了使用院校的意见和建议，在充分吸取了这些宝贵意见和建议的基础上，修订出版了《C 语言程序设计教程》（第二版）。该书是计算机基础教学的典型核心课程之一，也是计算机基础教学的基础和重点。

C 语言是目前国内外广泛使用的程序设计语言之一。C 语言功能丰富、表达能力强、使用方便灵活、程序执行效率高、可移植性好，广泛应用于系统软件设计、数据处理、科学计算等领域，不仅成为当前软件开发的主流语言，也成为很多高校计算机语言教学的首选语言。

该书在组织结构上，以 ANSI C 标准为主线，突出显示为两大部分。第一大部分从第 1 章到第 5 章，属于概念性基础部分，主要包括程序与程序设计语言、数据类型与表达式、控制结构的基本概念；第二大部分从第 6 章到第 13 章，属于技术与方法基础部分，主要包括结构化程序设计、模块设计、算法设计、面向对象程序设计方法等。在内容安排上遵循深入浅出、由简到繁、循序渐进的原则，突出让学生了解程序设计语言的基本知识、掌握程序设计的基本方法和常用算法，使学生具有使用 C 语言程序设计解决实际问题的初步能力。

本书语言表达严谨、流畅、逻辑性强、示例丰富，书中例题均在 C 语言的程序开发环境下调试通过，其中的关键性程序语句都有详细的注释，而且各章后附有很多历年的等级考试题目，方便各类人员自学及考试辅导。

本书由国防科学技术大学龙佑喜和湖南工业大学彭三城主编，参加编写的还有朱文球、满君丰，龙佑喜负责全书的统稿。刘震宇、张艳等老师为本书的修订再版做了大量细致的工作，为本书的顺利再版付出了心血，在此表示衷心的感谢！

由于作者水平有限，书中的不足和错误之处在所难免，恳请广大读者予以批评指正。

编者

2005 年 10 月

目　录

第1章　C语言程序设计概述

电子计算机自从20世纪40年代诞生以来，无论在硬件还是在软件方面，都有了极大的发展；在计算机应用的各个领域也都取得了丰硕的成果。

计算机本身是无生命的机器，要使计算机能够运行起来，为人类完成各种各样的工作，就必须让它执行相应的程序。这些程序都是依靠程序设计语言编制出来的。

在众多的程序设计语言中，C语言有其独特之处。它作为一种高级程序设计语言（更多的时候被称为中级语言），具备方便性、灵活性和通用性等特点；同时，它还向程序员提供了直接操作计算机硬件的功能，具备低级语言的特点，适合各种类型的软件开发。因此，C语言是深受软件工作者欢迎的程序设计语言。

本章主要从程序设计的角度，结合C语言的特点和发展，介绍有关程序设计的基本概念，以及C语言程序的基本结构、算法等内容。

1.1　程序与程序设计语言

1.1.1　程序的基本概念

为了让计算机按人的意图处理事务，人们必须预先设计好完成各种任务的程序，并预先将它们存放在存储器中。

所谓程序，实际上是用计算机语言描述的某一问题的解决步骤，是符合一定语法规则的符号序列。人们借助计算机能够处理的语言，告诉计算机要处理什么(即处理哪些数据)以及如何处理(即按什么步骤来处理)，这便是程序设计。通过在计算机上运行程序，向计算机发出一系列指令，便可按人们的要求解决特定问题。

解决某一个问题所编写的程序并不是唯一的，不同的用户（程序编写人员）由于其思考问题的方式、解决问题的思路不一样，因此，他们对于同一问题的解决所编写的程序也都往往不相同。不同的程序有不同的效率，这涉及到程序的优化，涉及到程序所采用的数据结构以及算法等多方面因素的综合。

1.1.2　程序设计语言

完成程序设计，自然离不开程序设计语言。了解程序设计语言的发展过程，有助于我们加深对程序设计语言的认识，使我们能更好地利用程序设计语言来解决有关问题。

当今程序设计语言发展非常迅速，新的程序设计语言层出不穷，其功能越来越强大。

但不管现代程序设计语言的功能如何增强，程序设计语言的种类怎样增多，从其发展历史以及功能情况看，大致可分成如下几个阶段：

1. 机器语言

由于物理器件和实现难易的影响，直到目前计算机还只能存放和识别由 0 和 1 组成的序列所表示的数据和指令。所谓机器语言，就是指该机器能够识别的指令的集合，即指令系统。在机器语言中，每条指令都用 0 和 1 组成的序列来表示。例如，以下是某计算机的两条机器指令：

加法指令：10000000

减法指令：10010000

不同类型的计算机，使用的机器语言不相同。

用机器语言编写的程序，计算机可以直接执行，执行效率高。但机器语言的指令不直观，难认、难记、难理解、且较烦琐，容易出错，写出的程序不能通用。编写机器语言程序时，要求程序员必须相当熟悉计算机结构，因而，目前很少直接用机器语言编程。

2. 汇编语言

20 世纪 50 年代中期，为了减轻人们使用机器语言编程的负担，开始采用一些“助记符号”来表示机器语言中的机器指令，这样便形成了汇编语言。助记符一般都是采用一个操作的英文字母的缩写，与机器语言相比，便于识别和记忆。例如，上例中的两条指令用汇编语言描述如下：

```
ADD    A, B     (表示数 A 加 B)
SUB    A, B     (表示数 A 减 B)
```

不过，计算机不能直接执行用汇编语言编写的程序，它必须经过一个叫汇编程序的系统软件翻译成机器语言程序后才能执行。我们称前者为源程序，后者为目标程序。

汇编语言指令和机器语言指令之间具有一一对应的关系。因而，不同的计算机其汇编语言也不尽相同，并且程序编写时仍需要对计算机内部结构比较熟悉，依然比较烦琐。但相对于机器语言而言，汇编语言要好多了。因而，在实际中，如果程序运行时间要求比较严格，程序与硬件操作联系紧密，人们还是常用汇编语言编写有关程序来解决这些问题。

3. 算法语言

汇编语言和机器语言是面向机器的，它们同属于低级语言的范畴。人们在使用它们设计程序时，要求对机器比较熟悉。为了克服低级语言的这一缺点，使人们将程序设计的精力集中在解决问题的算法上，便出现了面向算法过程的程序设计语言，称之为算法语言。比如：FORTRAN 语言、ALGOL 语言、PASCAL 语言、C 语言等。这类语言接近自然语言的形式，因而可以较大地降低编程的难度，提高编程的效率和质量，且设计的程序也更容易理解。

算法语言的一条语句相当于多条汇编语言指令或机器语言指令，表达能力强。在使用算法语言时，程序设计人员不需要熟悉计算机的指令系统，可以把精力集中在研究问题的求解方法步骤(过程)上。因此，算法语言也称为面向过程的程序设计语言。同时，算法语言不依赖于机器。为某种类型的计算机编写的算法语言程序，可以很方便地移植到其他类型的计算机上运行。举例来说，如果为苹果机写的一个程序能够方便地改为可以在 IBM PC 机上运行的程序，则称之为可移植的。相对于与机器有关的低级语言而言，算法语言

也就称为高级语言。

当然计算机也不能直接执行算法语言程序。如同汇编语言程序一样，算法语言程序(源程序)也必须先经过编译程序(或解释程序)翻译成机器语言程序(目标程序)后，才能由计算机执行。

我们现在要给大家介绍的 C 语言通常被称为中级计算机语言，之所以被称为中级语言，是因为它把高级语言的成分同汇编语言的功能结合起来了，通过后面章节的学习，我们将会了解到 C 语言允许对位、字节和地址这些计算机功能中的基本成分进行操作。

4. 面向任务的程序设计语言

利用算法语言求解一个复杂的问题，必须先要分析解决问题的过程，描述问题如何求解，然后才能用算法语言进行程序设计来实现。面向任务的程序设计语言是非过程化的语言，也就是说，不需要知道问题是如何求解的，只需要描述需求解的问题是什么，然后便可用程序设计语言来实现。

数据库操纵语言便是一种面向任务的程序设计语言。例如，设在某数据库应用系统中，有一个学生情况表 SS，若要从表中查找学生的信息，可以使用数据库查询语言(SQL)。采用 SELECT 语句便可完成这个任务，SELECT 语句描述如下：

SELECT SSNO，SSNAME，SSAGE，SSSEX FROM SS

该语句从 SS 表中查找到学生的 SSNO、SSNAME、SSAGE、SSSEX 等方面的信息。至于 SELECT 语句是如何进行查询的，用户就不必了解了。

面向任务的程序设计语言可以提高应用程序的开发速度和质量，也可使应用程序易于迅速地修改。同时，非计算机专业人员也能很方便地使用面向任务的程序设计语言开发自己的程序。

这类语言在应用系统的开发中使用较为广泛，尤其是一些管理信息系统应用程序的开发。

5. 面向对象的程序设计语言

面向对象程序设计语言在 20 世纪 90 年代开始流行。现在，面向对象的程序设计语言已成为程序设计的主流语言之一。由 C 语言发展出来的 C++就是一种非常优秀的面向对象的程序设计语言。

面向对象方法是一种分析方法、设计方法和思维方法的综合。面向对象方法学的出发点和所追求的基本目标是使人们分析、设计和实现一个系统的方法尽可能接近人们认识一个系统的方法。

在面向对象编程中，程序被看作是相互协作的对象集合，每个对象都是某个类的实例，所有的类构成一个通过继承关系相联系的层次结构。面向对象的程序设计语言具有对象生成功能、消息传递机制，以及类和继承机制。

对象是对客观事物的抽象，面向对象的编程，就是针对客观的事物设计程序。因此，面向对象的编程是非常直观的。面向对象的程序设计方法比面向过程的程序设计方法更清晰，更适合于开发大型复杂的软件。

综上所述，每一种语言都有它的优势和劣势。对于不同的问题，要根据实际情况来选择程序设计语言，以便更高效更优质地解决相关的问题。

1.2 C语言的发展及特点

1.2.1 C语言的发展概况

C语言是当今社会应用广泛，并受到众多用户欢迎的一种计算机算法语言。它既可作为系统软件的描述语言，也可用来开发应用软件。

C语言的出现是与UNIX操作系统紧密联系在一起的，C语言本身也有一个发展过程，目前仍然处于发展和完善之中。

从历史发展来看，C语言起源于1968年发表的CPL语言(Combined Programming Language)，它的许多重要思想来自于Martin Richards在1969年研制的BCPL语言，以及以BCPL语言为基础的由Ken Thompson在1970年研制成的B语言。K.Thompson用B语言写了第一个UNIX操作系统，用在PDP-7计算机上。D.M.Ritchie 1972年在B的基础上研制了C语言，并用C语言写成了第一个在PDP-II计算机上实现的UNIX操作系统。1977年出现了独立于机器的C语言编译文本《可移植C语言编译程序》，从而大大简化了把C语言编译程序移植到新环境所需做的工作，这本身也就使UNIX操作系统迅速地在众多的机器上实现。例如VAX，AT&T等计算机系统都相继开发了UNIX。随着UNIX的日益广泛使用，C语言也迅速得到推广。

1983年美国国家标准化协会(ANSI)根据C语言问世以来的各种版本，对C语言的发展和扩充制定了新的标准，称为ANSI C。1987年ANSI又公布了新标准——87ANSI C。

目前在微型计算机上使用的有Microsoft C，Quick C，Turbo C等多种版本。这些不同的C语言版本，基本部分是相同的，只在有关规定上略有差异。本书以Turbo C 2.0的环境对C语言作出介绍。Turbo C是一种快速、高效的编译程序。它不仅提供了一个集成开发的环境，同时也按传统方式提供了一个命令行编译程序版本，以满足不同用户的需要。

1.2.2 C语言的特点

事实证明，C语言是一种极具生命力的语言，它的特点是多方面的。一般可归纳如下：

(1)C语言具有结构语言的特点，程序之间很容易实现段的共享。它具有结构化的流程控制语句(如if-else语句，switch语句，while语句，do-while语句，for语句)，支持若干种循环结构，允许编程者采用缩进书写形式编程。因此，用C语言设计出的程序层次结构清晰。

(2)C语言的主要结构成分为函数，函数可以在程序中被定义完成独立的任务，独立地编译成代码，以实现程序的模块化。

(3)C语言运算符丰富，运算符包含的范围很广泛。C把赋值、括号、强制类型转换都当作运算符处理。灵活地使用各种运算符可以实现在其他的高级语言中难以实现的运算。

(4)C语言数据类型丰富。数据类型有整型、实型、字符型、数组型、指针型、结构体型、共用体型等。能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。尤其是C

语言的指针型数据的运算，更是灵活、多样。

(5)C 语言允许直接访问物理地址，即可直接对硬件进行操作，实现汇编语言的大部分功能。C 语言这一特点，使得它成为编制系统软件的基本语言(UNIX 的绝大部分就是由 C 语言写成的)。

(6)C 语言语法限制不太严格，程序设计自由度大。这样使 C 语言能够减少对程序员的束缚。“限制”与“灵活”是一对矛盾。限制严格，就易失去灵活性；而强调灵活，就必然放松限制。从这个角度来看，使用 C 语言编程，要求编程者对程序设计技巧要更加熟练一些。

(7)用 C 语言编程，生成的目标代码质量高，程序执行效率高。同时用 C 语言写的程序可移植性好。

C 语言优点很多，但是它也存在一些缺点，如运算优先级太多，数值运算能力方面不象其他高级语言那样强，语法定义不严格等。尽管 C 语言目前还存在一些不足之处，但由于它目标代码质量高、使用灵活、数据类型丰富、可移植性好而得到广泛的普及和迅速的发展，成为一种受到广大用户欢迎的实用的程序设计语言，同时也是一种在系统软件开发、科学计算、自动控制等各个领域被广泛应用的程序设计语言。

1.3 算法及其描述

在程序设计中，不可避免地需要涉及算法。有人这样说过：“计算机科学就是研究算法的科学”，足见算法在程序设计中的重要性了。下面从算法的概念和描述方法两方面，对算法的问题进行讨论。

1.3.1 算法的概念

著名的瑞士计算机科学家、PASCAL 语言发明者 N·沃思(Niklaus Wirth)教授提出了程序定义的著名公式：

程序＝算法+数据结构

这个公式的重要性在于它说明了程序与算法的关系。同时，说明了数据结构的选择也是十分重要的。对于程序而言，算法与数据结构是统一的关系。

通常认为，算法是对特定问题求解步骤的一种描述。对于同一问题，可以有不同的解题方法和步骤，当然，方法有优劣，有的方法只需很少的步骤，而有些方法则需要较多的步骤。我们希望采用简单的和运算步骤少的方法。

算法应当具备以下几个方面的特点：

(1)一个算法必须保证执行有穷步之后结束；

(2)算法的每一个步骤必须具有确切的定义；

(3)应对算法给出初始量；

(4)算法具有一个或多个输出；

(5)可行性——算法不能进行是不允许的，如“计算 X / 0”

算法必须能在有限时间内完成，且对相同的输入有相同的输出。在程序设计语言中，与

算法密切相关的便是语句，包括与程序执行处理有关的“功能语句”(如输入语句、输出语句、赋值语句、调用语句等)和与程序执行流程有关的语句(如条件语句、循环语句等)。对程序设计而言，算法的确定也就是如何合理安排这些语句以完成人们要求的特定功能。

1.3.2 算法的描述方法

从上面的分析可知，算法是描述某一问题求解的有限步骤，而且必须有结果输出。设计一个算法，或者描述一个算法，最终是由程序设计语言来实现的。但算法与程序设计又是有区别的，主要是一个由粗到细的过程。算法是考虑实现某一个问题求解的方法和步骤，是解决问题的框架流程；而程序设计则是根据这一求解的框架流程进行语言细化，实现这一问题求解的具体过程。例如，求 1+2+...+10，对于这个问题，我们可以用最原始的方法进行，即：

第 1 步：先求 1+2，得到结果 3。

第 2 步：将第 1 步得到的和 3 再加上 3，得到结果 6。

第 3 步：将第 2 步得到的和 6 再加上 4，得到结果 10。

……

第 9 步：将第 8 步得到的和 45 再加上 10，得到最后结果 55。

这种算法，虽然正确，但很繁琐。如果要求 1+2+...+1000，则需要 999 步，显然是不合适的。

一般而言，可以使用下面几种类型的工具描述算法：

1. 自然语言

自然语言即是人们日常进行交流的语言，如英语、汉语或其他语言等。自然语言用来描述算法，分析算法，作为用户相互之间进行交流，是一种较好的工具。但是将自然语言描述的算法直接在计算机上进行处理，目前还存在许多困难，包括有诸如语音语义识别等方面的问题，所以一般不用自然语言描述算法。

2. 专用工具

要对某一个算法进行描述，可以借助于有关图形工具或代码符号。常用的工具有流程图、N-S 图等。

20 世纪 50～60 年代兴起的流程图几乎成为了程序设计及算法描述的必用工具。

人们已经提出了多种描述算法的流程图。这些方法的特点是用一些图框表示各种类型的操作，用线表示这些操作的执行顺序，图 1.1 为我国国家标准 GB1526—89 中推荐的一套流程图标准化符号的一部分。

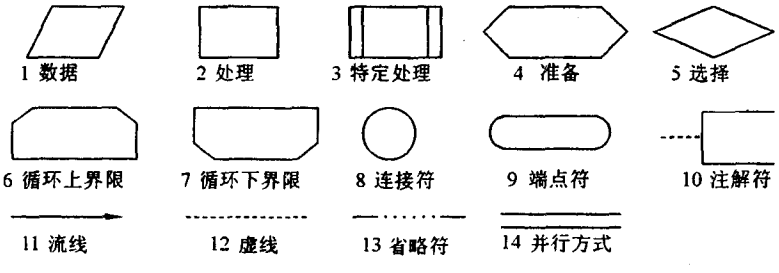

图 1.1 流程图标准化符号

• 数据　平行四边形表示数据，其中可注明数据名称、来源、用途或其他文字说明。

• 处理　矩形表示各种处理功能。矩形内可注明处理名称或其简要功能。

• 特定处理　带有双竖边线的矩形。矩形内可注明特定处理名称或简要功能，表示已命名的处理。该处理为在另外地方已得到详细说明的一个操作或一组操作。

• 判断　菱形表示判断。菱形内可注明判断的条件。它只有一个入口，但可以有若干个可供选择的出口。

• 循环界限　循环界限包含循环的上界和下界，中间是要循环执行的处理内容，称为循环体。循环界限由去上角的矩形(表示上界限)和去下角的矩形(表示下界限)构成。

• 端点　扁圆形表示转向外部环境或外部环境转入的端点符。例如，程序流程的起始点。

• 注解　注解是程序的编写者向阅读者提供的说明。它用虚线连接到被注解的符号或符号组上。

例 1.1 求 5!。

用流程图表示的算法如图 1.2 所示：

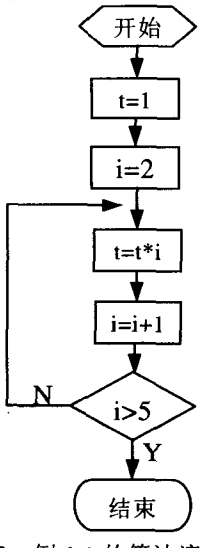

图 1.2　例 1.1 的算法流程图

3. 伪代码表示

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。它如同一篇文章，自上而下地写下来。每一行（或几行）表示一个基本操作。它书写方便、格式紧凑，也比较好理解。

例 1.2 求 5!。

用伪代码表示的算法如下：

开始

　　置 t 的初值为 1

　　置 i 的初值为 2

　　当 i<=5，执行下面操作：

　　　　使 t=t*i

　　　　使 i=i＋1