普通高等教育“十二五”规划教材　计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

程序设计基础

——C语言

杨　莉　刘鸿翔◎主 编

科学出版社

普通高等教育“十二五”规划教材　计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

程序设计基础——C 语言

杨　莉　刘鸿翔　主　编
唐宏亮　邓　芳　副主编
余　慧　王海军

科学出版社
北　京

内 容 简 介

本书主要内容包括程序设计基础知识、C 语言数据类型、运算符与表达式、程序结构、数组、指针、其他数据类型、函数、文件等，通过对大量实例进行分析，力求提高和培养学生的程序设计能力。本书有配套辅导教程《程序设计基础实训指导教程——C 语言》（科学出版社出版），引导读者学习和巩固各章节内容。

本书结构清晰、内容精练、概念清楚、实例丰富、深入浅出，对读者可能遇到的疑难问题和易混淆概念作了详细的阐述。

本书可作为高等院校相关专业教材和参考用书，还可以作为参加二级 C 语言程序设计考试者的自学用书。

图书在版编目(CIP)数据

程序设计基础：C 语言/杨莉，刘鸿翔主编. —北京：科学出版社，2011
ISBN 978-7-03-032903-5

Ⅰ. ①程… Ⅱ. ① 杨… ② 刘… 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（2011）第 246298 号

责任编辑：戴 薇 郭丽娜 / 责任校对：马英菊
责任印制：吕春珉 / 封面设计：东方人华平面设计部

科学出版社 出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

双青印刷厂 印刷

科学出版社发行 各地新华书店经销
*
2012 年 1 月第 一 版 开本：787×1092 1/16
2012 年 1 月第一次印刷 印张：18 3/4
字数：449 000

定价：32.00 元

（如有印装质量问题，我社负责调换〈双青〉）
销售部电话 010-62142126 编辑部电话 010-62134021

前　言

随着计算机在社会各个领域的广泛应用，对计算机的应用能力，特别是程序设计能力的要求在不断提高。程序设计基础课程作为计算机能力培养的重要课程，侧重于培养学生掌握程序设计的基本方法和技巧，以及编写程序解决相关专业领域问题的能力。

在众多的程序设计语言中，C 语言因其具有完备的高级语言特性、丰富灵活的控制和数据结构、简洁而高效的语句表达以及清晰的程序结构，一直得到广泛的应用。即使在面向对象程序设计语言逐渐普及的今天，各理工类专业，尤其是计算机专业，程序设计语言入门课程都首选 C 语言。

本书共分九章，内容包括程序设计概述、数据类型、运算符与表达式、程序结构、数组、指针、其他数据类型、函数、文件、综合应用等。

根据初学者的特点和认识规律，本书内容的组织遵循先易后难、深入浅出的原则。从第 2 章开始每章后面增加本章小结及常见错误列举，对读者可能遇到的疑难问题和易混淆概念作了详细的阐述，帮助读者深刻理解。书中给出的综合应用，能帮助初学者了解大型程序的编写技巧，为学生提供了课程设计的指导资料，具有参考和借鉴价值，从而提高学生兴趣，激发上进，使得理论与实践结合得更为紧密。

程序设计是一门实践性很强的课程，读者在学习过程中一定要重视实践环节。在掌握概念的基础上，要求能够独立完成每章习题中的设计和开发型编程题，并要求能上机调试运行。与本书配套的《程序设计基础实训指导教程——C 语言》（与本书同时出版），主要用于上机实训、等级考试实训和自测练习。

本书主编十几年来一直从事程序设计基础的一线教学工作，也是学校程序设计基础的精品课程负责人，在多年的程序设计基础课程的教学改革实践基础上编写了本书。本书具有概念清晰、例题丰富、实用性强的特点。

参加本书编写与审稿工作的有杨莉、刘鸿翔、唐宏亮、邓芳、余慧、王海军、王芳、宋婉娟、杨宜波、万润泽、许庆炜、文中林。书中所给出的实例程序全部在 Visual C++ 6.0 环境下调试通过。在本书的内容编排和文字排版等方面，科学出版社给予了多方面的支持与帮助，在此表示衷心的感谢。

由于编者水平有限，书中难免有欠妥之处，恳请广大读者提出宝贵意见。

目　录

第 1 章　程序设计概述

自 1946 年世界上第一台计算机问世以来，计算机的应用已无处不在，无所不及。一种是人们利用计算机强大的计算能力和事务处理能力为自己的事业和生活服务，如利用 Photoshop 处理图片、利用 Word 编写文档、利用 PowerPoint 设计幻灯片、利用 Internet 获取信息等。而另一种则是利用计算机进行程序设计，学会程序设计意味着真正地走进了计算机的世界，程序设计语言本身就是与计算机进行交互的有力工具。本章主要介绍程序设计的相关概念和 C 语言程序设计概述及 Visual C++ 6.0 开发环境。

1.1　基 本 概 念

1.1.1　程序

程序一词来自生活，通常指完成某些事务的一种既定方式和过程，也可以将程序看成对一系列动作的执行过程的描述。日常生活中可以找到许多“程序”实例。例如，到图书馆借书的过程描述如下：

S1. 进入图书馆；
S2. 查书目；
S3. 按书目找书；
S4. 办理借书手续；
S5. 离开图书馆。

这是一个直线形程序，是形式最简单的程序，它给出一个包含具体步骤的序列。如果按顺序实施这些步骤，也就完成了该项事务。

上述过程是一种理想情况，而实际过程要复杂得多，这里再给出一个细化的过程描述。

S1. 进入图书馆；
S2. 查书目；
S3. 按书目找书；
S4. 如果该书已经借出，可以有两种选择：
　　S4.1. 回到 S2（进一步查找其他相关的书目）；
　　S4.2. 放弃借书，到 S6；
S5. 找到了要借的书，办理借书手续；
S6. 离开图书馆。

这个程序比前一个复杂一些，它不再是一个平铺直叙的动作序列，步骤更多，还出现了分情况处理（S4）和可能出现的重复性动作（S4.1）。如果仔细探究这个实例，可以发现还有很多情况未考虑，即这一程序还可以进一步细化。

现实生活中有许多程序性活动，通常需要按部就班完成一系列动作。对这种工作（事物、活动）过程细节动作的描述就是一个“程序”。

在一个程序描述中，总有一批预先假定的“基本动作”，这些基本动作是程序执行

者能够理解和直接完成的。一个程序总有开始与结束，在执行此程序的过程中，动作者需要按照程序的描述执行一系列的动作，在达到结束位置时工作就完成了。

1.1.2　计算机程序

计算机里的程序执行与日常生活中的程序性活动情况很相似。从这点出发，可以帮助理解计算机的活动方式。当然，人们日常生活中的程序性工作有更多变数，许多事情并不要求完全按程序做，可以有许多“灵活性”。而计算机对程序的执行则完全是严格的，必须一步步按程序中的指令办事，一点“商量”的余地都没有。

人们把需要计算机完成的任务编排出正确的方法和步骤，用计算机能够理解的语言表达出来，就是“程序”。例如，要计算机计算两个数的平均值，计算步骤如下。

1）确定要计算的是哪两个数。

2）求两个数的和。

3）将此和除以 2。

4）输出计算结果。

编写程序的目的就是将这些步骤用计算机能够接受的形式告诉计算机，指挥计算机完成任务。

综上所述，程序就是**为实现特定目标或解决特定问题而用计算机语言编写的、可以连续执行并能够完成一定任务的指令序列的集合**。程序是用汇编语言、高级语言等编写出来的可以运行的文件，在计算机中称为可执行文件（扩展名一般为.exe），是计算机软件的一个实例。

所谓指令，就是**计算机可以识别的命令**。虽然在人类社会中，各民族都有丰富的语言用以表达思想、交流感情、记录信息，但计算机不能识别它们。计算机所能识别的指令形式，只能是“0”和“1”的组合。**一台计算机硬件系统所能识别的所有指令的集合**，称为它的指令系统。

一个程序包括数据和算法两个部分。前者是对数据形式的表示和描述（数据流），即指定程序所使用数据的数据结构和组织形式，不同语言对数据定义不同，本书是指 C 语言的数据类型和数据结构；后者是对数据进行操作的描述（控制流），指定操作的步骤，即程序设计的算法。因此，数据和算法是程序中既对立又统一的两个概念，它们之间的对立统一体现了程序要处理的数据对象与处理数据对象方法之间的关系。没有数据，算法就没有运算处理的对象。算法具有通用性，它脱离于语言之外，是程序设计的灵魂。

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，即

程序=算法+数据结构

1.1.3　程序设计

程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，如本书给出的是 C 语言的程序设计。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。专业的程序设计人员常被称为程序员。目前程序设计方法主要有面向过程的结构化程序设计和面向对象程序设计。本书只介绍面向过程的结构化程序设计。

结构化程序设计方法的主要原则可以概括为自顶向下、逐步求精、模块化、限制使用 goto 语句。

1）自顶向下。程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。不要一开始就过多追求众多的细节，先从最上层总目标开始设计，逐步使问题具体化。

2）逐步求精。对复杂问题，应设计一些子目标作过渡，逐步细化。

3）模块化。一个复杂问题，肯定是由若干稍简单的问题构成。模块化是把程序要解决的总目标分解为分目标，再进一步分解为具体的小目标，把每个小目标称为一个模块。

4）限制使用 goto 语句。结构化程序设计采用单入口单出口的控制方式，程序由顺序、分支选择和循环三种基本结构组成。这三种基本结构的特点是每一种结构只有一个入口和一个出口，任何一个问题都可以用三种基本结构实现，任何复杂的程序都可以由这三种基本结构组成。结构化程序设计使得程序结构清晰、可读性好，在出现问题时，便于查错，易于修改，提高了程序设计的质量。

1.1.4　软件

软件是计算机系统的重要组成部分。软件由程序和有关程序的技术文档资料组成，具有专门而完善的功能。

软件系统包括系统软件和应用软件。系统软件是围绕计算机系统本身开发的程序系统，如我们使用的各类操作系统（Windows、UNIX、DOS 等）、语言编译程序、数据库管理软件等。应用软件是专门为了某种使用目的而编写的程序系统，常用的应用软件包括文字处理软件、专用财务软件、人事管理软件、图形处理软件等。

1.2　程序设计语言

程序设计语言的发展经历了以下几个阶段。

1. 第一代语言

第一代语言是机器语言，它由计算机的指令系统组成。由于计算机只能识别和存储二进制的数据和指令，机器语言中的每一条语句（机器指令）实际就是由 0 和 1 组成的二进制代码，它由操作码的二进制编码和操作数的二进制编码组成。机器指令通常随计算机类型不同而不同，编写效率较低。例如，通过某类型的计算机计算

$$a=\frac{2.58+5.36\times1.64}{1.27}$$

则其机器语言表示如表 1.1 和表 1.2 所示。

表 1.1　数据

地　址	数
0500	2.58
0501	5.36
0502	1.64
0503	1.27
0504	a

表 1.2　程序

地　址	指　令		注　释
	操 作 码	地 址 码	
0600	021	0501	取 5.36
0601	012	0502	乘 1.64
0602	010	0500	加 2.58
0603	013	0503	除以 1.27
0604	022	0504	存储 a
0605	035	0504	打印 a
0606	007	0000	停止

计算机虽然可以直接识别和执行机器语言程序，执行效率高，但是人工编写机器语言程序较为繁琐，且易出错。

2. 第二代语言

第二代语言是汇编语言，也称为符号语言，它用符号代替机器语言中的二进制编码，这样看起来较直观，不易出错。上例的汇编语言程序如下：

```
K:  LDA     5.36
    MUL     1.64
    ADD     2.58
    DIV     1.27
    STA     a
    PRINTF      a
    STOP
```

与机器语言一样，不同类型的计算机具有不同的汇编语言。计算机不能直接识别和执行汇编语言程序，必须把汇编程序（系统软件）转换成机器语言（目标程序）后，才能执行，如图 1.1 所示。

汇编语言与机器语言是一一对应的，一个复杂的程序包含许多汇编语言指令，写起来较繁琐。

机器语言和汇编语言都是面向机器的语言，因此两者也被称为“低级语言”。

图 1.1　汇编语言的汇编与执行

3. 第三代语言

第三代语言也称为算法语言。算法语言中的语句较为接近自然语言的英文字句，数据用十进制来表示，从而更容易为人们掌握和理解。并且算法语言独立于计算机系统，用它编写的程序可以在任何其他类型的计算机上执行。因此算法语言也称为“高级语言”。

常用的高级语言有 BASIC、FORTRAN、ALGOL、COBOL、PASCAL、C 语言等。上例用 BASIC 语言实现如下：

```
10  a=(2.58+5.36*1.64)/1.27
20  PRINTF a
30  END
```

用高级语言编写的程序称为“源程序”，计算机不能直接识别和执行，必须将它翻译成计算机能够识别的机器指令表示的目标程序。翻译方式有解释方式和编译方式。解释方式是解释一条执行一条。BASIC 语言采用解释方式翻译执行，C 语言源程序采用编译方式翻译执行，需要经过编译程序（系统软件）编译成目标程序（*.obj），再经过连接程序（系统软件）连接成可执行文件（*.exe）后，才能执行，其过程如图 1.2 所示。

图 1.2　语言的编译与执行算法

4. 第四代语言

第三代语言是过程化语言，必须描述问题是如何求解的。第四代语言是非过程化语言，只用描述需要求解的问题是什么。

例如，需要将某班学生的成绩按高低的次序输出。用第四代语言只需写出这个要求即可，而不必写出排序的过程。数据库查询语言就可以做到这一点，因此可以把数据库查询语言看成是最简单的第四代语言。

5. 第五代语言

第五代语言主要是为人工智能领域设计的，如知识库系统、专家系统、推理工程、自然语言处理等，称为智能化语言。在这些领域内将复杂的知识进行编码，使得计算机能从中得出推论。

第五代语言还处于萌芽状态，PROLOG 语言可以看作是它的一个例子。

1.3　问题求解与算法设计

1.3.1　计算机求解问题的步骤

人们所要解决的问题，大致有两类：一类问题可以抽象成数学模型，通过求解这个数学问题达到求解问题的本身；另一类问题由于人类的认识有限、使用方法有限或其他一些原因的存在，无法抽象出数学模型，因而只能通过模拟事物本身运作的过程来解决。但不管是哪一类问题，都需要首先明确问题的描述、问题的要求，然后根据已有的知识、认识的能力，以一种较抽象的方式来表达问题，再提出解决该问题的途径。若存在多种

途径，可从中选择最合适的一种，然后设计解决方案并把它付诸实施，最终解决这个问题。对于一些大的、复杂的问题还需作可行性研究。

综上所述，可以把人解决问题的步骤归纳为以下几步：

S1. 明确问题；

S2. 精确表达问题；

S3. 设计解决方案；

S4. 实施解决。

从解决具体问题的过程出发，目的是为了导出计算机帮助人解决问题的方法和步骤，让计算机按照人的思维过程，完成人布置的任务。当然，目前计算机只是一个电子装置，无法像科幻电影中的机器人一样，具有逻辑思维能力，可以自动完成工作。

计算机硬件必须在系统软件和应用软件的支持下，按照人所规定的解决方案，完成指定的工作。因此，计算机在配合人解决问题的过程中，只是根据人提出的解决方案，给出一个解决问题的数值或模拟。如果解决方案本身已经抽象出数学模型，它就可以解这个数学模型，进而得到数值解；如果解决方案是一个逻辑模拟模型，计算机可求得模拟实现过程。

使用计算机解决问题时，首先要把具体问题抽象为模型，再把模型表现为用具体的计算机语言描述的程序。在这一过程中，程序设计者不应首先考虑实现细节，而要把精力放在建立系统的总体模型上。如果把程序看作是构成系统的程序零件，首先要根据系统的要求或功能，考虑系统应由哪些程序零件构成，然后再考虑各个程序零件该如何实现。

因此，使用计算机解决问题，大致有如下步骤：

S1. 明确问题；

S2. 精确表达问题；

S3. 设计解决方案（模型或算法）；

S4. 把解决方案用计算机程序实现（程序设计）；

S5. 计算机运行、求解。

其中，S1～S4 都要由人来完成。对于实际中的复杂问题，S1～S3 通常会涉及问题所处领域的专业知识，由专业人士完成或参与完成，不具有通用的方法；S4 由程序员完成；S5 才是计算机的工作。

为了学习程序设计，读者应了解上面 5 个步骤，掌握计算机对简单问题的求解步骤。

1.3.2　算法定义

算法是**为了解决一个特定问题而采取的确定的、有限的、按照一定次序进行的、缺一不可的执行步骤**。一个好的算法将产生质量较好的程序。

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骤之后结束；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算法必须能够进行，如“计算 X/0”是不允许的。

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要求的特定功能。

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。学习高级语言时，一方面要熟练掌握该种语言的语法规则，因为它是算法实现的基础；另一方面必须认识到算法的重要性，加强分析问题的能力训练，写出高质量的程序。

设计算法的过程是从具体到抽象的过程。首先，应该清楚人工处理应该采取的主要步骤；其次，对这些步骤进行归纳整理，抽象出数学模型；最后，采用描述工具加以描述。算法描述工具包括伪代码、流程图、N-S 图、UML 等。

1.3.3　伪代码

伪代码（pseudocode）是一种算法描述语言。伪代码是介于自然语言与编程语言之间的文字和符号描述算法，是在算法开发过程中用来表达设计思想的符号系统。使用伪代码的目的是为了使被描述的算法可以容易地以任何一种编程语言（PASCAL、C、Java 等）实现。因此，伪代码必须结构清晰、代码简单、可读性好，并且类似自然语言。

伪代码只是像流程图一样用在程序设计的初期，帮助写出程序流程。简单的程序一般都不用写流程、写思路，但是复杂的代码，最好还是把流程写下来，总体考虑整个功能如何实现。写完以后不仅可以用来作为以后测试、维护的基础，还可用来与他人交流。但是，如果把全部的东西写下来必定会浪费很多时间，那么这个时候可以采用伪代码方式。

例如，求 Y=1-1/2+1/3-1/4+1/5-…前 30 项之和。用伪代码表示的算法如下：

```
BEGIN(算法开始)
0 =>Y
1 =>i
-1 =>f
while i <= 30
{
    Y + 1 / i * ( - f ) =>Y
    i+1 =>i
}
printf Y
END(算法结束)
```