程序设计基础教程（C语言）

CHENGXU SHEJI JICHU JIAOCHENG

主　编◎黄同成　周红波

湖南人民出版社

程序设计基础教程（C语言）

CHENGXU SHEJI JICHU JIAOCHENG

主　编◎黄同成　周红波

副主编◎刘金祥　成娅辉　夏太武
黄　磊　陈　智

CS 湖南人民出版社

图书在版编目（CIP）数据

程序设计基础教程：C语言 / 黄同成、周红波主编. —长沙 ： 湖南人民出版社，2011.12

ISBN 978-7-5438-8089-4
Ⅰ. ①程… Ⅱ. ①黄… Ⅲ. ①C语言－程序设计－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2011）第275280号

程序设计基础教程（C语言）

编 著 者　黄同成　周红波
责任编辑　赵颖峰　董静静
装帧设计　谌　茜

出版发行　湖南人民出版社［http://www.hnppp.com］
地　　址　长沙市营盘东路3号
邮　　编　410005
经　　销　湖南省新华书店

印　　刷　湖南天闻新华印务邵阳公司
版　　次　2012年1月第1版
　　2012年1月第1次印刷
开　　本　889×1194　1/16
印　　张　19.25
字　　数　15千字
书　　号　ISBN978-7-5438-8089-4
定　　价　45.00元

营销电话：0731-82226732　（如发现印装质量问题请与出版社调换）

内容提要

本书作为C语言程序设计的入门与应用教材，全书分三大模块，分别是基础篇、提高篇、应用拓展篇，共9章。主要内容包括：C语言概述、C语言程序设计的初步知识、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、编译预处理、指针、构造数据类型、文件、位运算、项目实践等。本书注重基础，突出应用。

为便于读者能够综合运用本章知识点提高实际编程能力，第九章详细分析了一个实际项目的开发全过程，包括系统分析及功能实现。在项目实例中综合应用前面各章所学的C语言知识，从而能够帮助读者将前面所学的知识点串联起来，真正能够从程序设计的角度加以灵活运用。

本书易教易学、学以致用、注重能力，对初学者容易混淆的内容进行了重点提示和讲解。本书适合作为普通高等院校应用型本科(含部分专科、高职类)各相关专业的程序设计教材，也适合编程开发人员培训、自学使用。

本书配有电子教案，并提供程序源代码，以方便读者自学。

前 言

C语言在计算机程序设计领域应用广泛，功能丰富，语句简洁，使用方便，语法灵活，用c语言编写的程序具有极强的可移植性。C语言作为一种既适合编写系统软件，又适合开发应用软件的高级程序设计语言，已经成为各高校学生学习的主要程序设计语言之一，各类计算机考试都包含了C语言程序设计的内容。

探讨如何使教师和学生轻松、愉快地完成C语言程序设计课程的教学和学习，具有重要的学术理论价值和社会实践意义。针对C语言比较难学的现实情况，本教材在编写过程中，充分吸收了近年来我国高等教育改革的成果，重视算法分析与设计，代码设计中尽可能增加注释语句，帮助学生更好地理解程序设计思想，选择贴近现实的综合性案例，帮助学生将本章知识点融会贯通，实现知识向能力的转化。对全书内容做了合理组织和精心安排，用简洁精练的语言和典型的例题帮助学生理解复杂的概念，每章内容都按照循序渐进的方式进行组织，由浅入深，相互呼应，引导学生掌握C语言的编程方法，提高他们的应用能力。

本书共9章，结构新颖，分为基础篇、提高篇、应用拓展篇三个部分，方便不同层次、不同专业根据需要选择教学内容。主要内容有C语言概述、简单的C程序设计、基本数据类型及表达式、选择结构、循环结构、函数及预处理、数组、指针、复杂结构类型、文件等。本书内容由浅入深，强化知识点、算法、编程的方法与技巧，增加“C语言程序设计项目实践”章节，培养学生综合分析与应用能力。为读者能够很好地学习和应用C语言进行程序设计打开了方便之门。

本书是由长期从事高等院校计算机基础教学的教师合作编写的，是参编者多年教学经验和智慧的结晶。本书是以培养学生程序设计的基本方法和基本技能为目标，以应用能力为侧重点的特色鲜明的教材。既顾及C语言本身内容的完整性和知识的系统性，又对C语言进行清晰、全面的讲解。本书采用Turbo C++3.0作为语法规则，所有例题均在Visual C++语言环境下上机调试并通过。

本书第1章由黄同成与周红波合写，第2章由周红波编写，第3、4章由刘金祥编写，第5～8章分别由夏太武、陈智、成娅辉和黄磊编写，第9章及附录由黄同成编写，全书的框架设计、统稿、改稿和定稿工作由黄同成教授完成。

在本书的写作过程中，我们对书稿进行了反复的修改，几易其稿，并得到了不少专家和任课教师的大力支持，他们为本书的编写提出了许多宝贵的意见和建议，在此表示衷心的感谢。由于编者水平有限，编写时间仓促，书中难免存在许多不足之处，恳请广大读者和同行批评指正。

编者

2011 年12 月

目　录

基础篇

基础篇

第 1 章　C 语言概述

C 语言是目前世界上广泛流行的一种结构化高级程序设计语言。使用 C 语言不仅可以编写应用软件，也可以编写系统软件。本章在简要介绍程序设计与算法的基础上，通过典型而简单的 C 语言实例，引入 C 程序设计的基本方法，使读者在短期内建立 C 程序设计的基本概念。同时，本章也对 C 语言的特点、集成开发环境 Visual C++ 6.0 及其调试方法、其他常用开发语言进行了简单介绍。

1.1　程序设计与算法简介

1.1.1　计算机语言与程序设计的概念

1. 计算机语言

计算机语言是人与计算机进行交流的工具，是用来书写计算机程序的工具。按照程序设计语言的发展过程，可以分为机器语言、汇编语言和高级语言三类，其特点如表 1-1 所示，举例说明如表 1-2 所示。

表 1-1　三类语言的特点比较

低级语言	机器语言	机器指令（由 0 和 1 组成），可直接执行	难学、难记 依赖机器的类型
	汇编语言	用助记符代替机器指令，用变量代替各类地址	克服记忆的难点 依赖机器的类型
高级语言		类似数学语言，接近自然语言	具有通用性和可移植性 不依赖具体的计算机类型

表 1-2　三类语言的程序举例

机器指令	汇编语言指令	指令功能	高级语言（C 语言）

10110000
00001000
	MOV　AL, 3	把 3 送到累加器 AL 中	#include <stdio.h>
			void main( )　　//完成 3+2 的运算
			{
			　int a, b, c;
			　a = 3;
			　b = 2;
			　c = a + b;
			　printf("a + b = %d\n", c);
			}

00000100 00000001	ADD　AL,2	2 与累加器 AL 中的内容相加（即完成 2+3 的运算），结果仍存在 AL 中	
11110100	HLT	停止操作	

注意：由于计算机只能识别 0 和 1 组成的机器语言，所以汇编语言和高级语言都需要翻译成机器语言才能执行。

将汇编源程序翻译为目标程序（机器语言）的过程称为汇编，如图 1-1 所示。

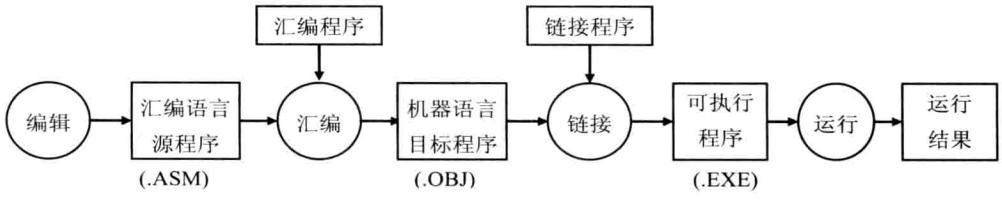


图 1-1　汇编过程

高级语言翻译为目标程序的方式有两种：解释方式和编译方式。

解释方式是将高级语言源程序逐句解释为机器语言并执行，好比口译方式，执行过程如图 1-2 所示。解释方式灵活方便，不产生目标程序。但因为是边解释边执行，程序执行效率低。

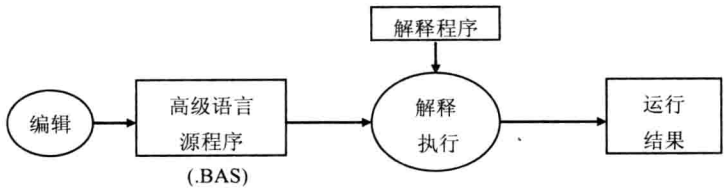


图 1-2　解释过程

编译方式是将高级语言源程序翻译成目标程序后，再链接成机器可直接运行的可执行文件，好比笔译方式，执行方式如图 1-3 所示。链接的原因是由于在目标程序中还可能要调用一些标准程序库中的标准子程序或其他自定义函数等，由于这些程序还没有链接成一个整体，因此，需通过“链接程序”将目标程序和有关的程序库组合成一个完整的“可执行程序”。由于产生的可执行程序可以脱离编译程序和源程序独立存在并反复使用，故编译方式执行速度快，但每次修改源程序后，必须重新编译。一般高级语言 C/C++，FORTRAN、PASCAL 等都采用编译方式。

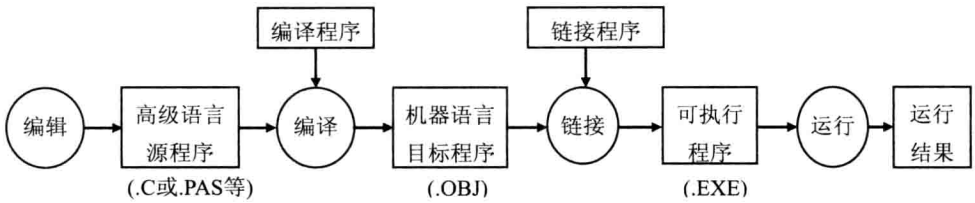


图 1-3　编译过程

说明：汇编语言源程序和高级语言源程序都是纯文本文件，可用文本编辑器生成，如记事本、VC++6.0 的新建文件功能。

2. 程序设计

计算机通过执行程序来完成工作，如计算、通信、控制等。所谓程序，就是遵循一定规则并能完成指定工作的一系列指令和数据的集合。采用计算机语言进行程序编写，以使计算机解决问题的整个处理过程就称为程序设计。计算机解决问题的基本过程如图 1-4 所示。

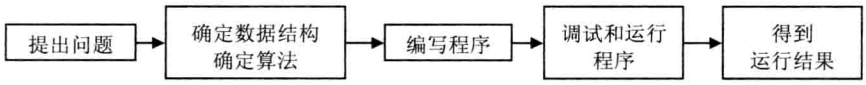


图 1-4　计算机解决问题的基本过程

1.1.2　算法简介

算法，就是解决某一应用问题的步骤，是程序设计的基础。例如，红、蓝两个墨水瓶中的墨水被装反了，要把它们分别按颜色归位，这就是一个“交换算法”。解答该算法的关键在于引入第三个瓶子，假设为白瓶子。过程如下：首先将蓝瓶子的墨水倒入白瓶子；其次将红瓶子的墨水倒入蓝瓶子；最后将白瓶子的墨水倒入蓝瓶子。

著名的计算科学家沃思（N.Wirth） 提出了一个经典的公式：

程序＝数据结构＋算法

数据结构描述的是数据的类型和组织形式，算法解决计算机“做什么”和“怎么做”的问题。每一个程序都要依赖数据结构和算法，采用不同的数据结构和算法会带来程序的不同质量和效率。实际上，编写程序的大部分时间还是用在算法的设计上。

一个算法应该具有如下特点：

① 有穷性。算法仅有有限的操作步骤（空间有穷），并且在有限的时间内完成（时间有穷）。如果一个算法需执行 10 年才能完成，虽然是有穷的，但超过了人们可以接受的限度，不能算是一个有效的算法。

② 确定性。算法的每一个步骤都是确定的，无二义性。例如，a 大于等于 b，则输出 1；a 小于等于 b，则输出 0。在算法执行时，如果 a 等于 b，算法的结果就不确定了。因此，该算法是一个错误的算法。

③ 有效性。算法的每一个步骤都能得到有效的执行，并得到确定的结果。例如，如果一个算法将 0 作为除数，则该算法无效。

④ 有 0 个或多个输入。

⑤ 有 1 个或多个输出。没有输出的算法没有任何意义。

算法的表示方法有多种，这里仅介绍自然语言法、流程图法和计算机语言法。

【例 1-1】 输出两个数中较大的一个数。

方法 1：用自然语言描述。

步骤 1：输入两个任意数，分别存入变量 x 和 y 中；

步骤 2：比较 x 和 y 的值，如果 x 大于 y，则输出 x 的值，否则输出 y 的值。

可以看到，用自然语言描述易于理解，但冗长，难于描述复杂算法。例如用自然语言描述输出 10 个数的最大值就很复杂。

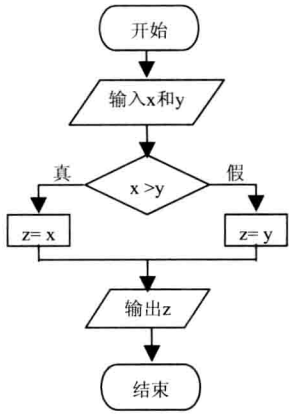
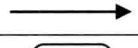
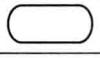


图 1-5　用流程图表示例 1.1 算法

方法 2：用流程图表示，如图 1-5 所示。

流程图（flow chart）是用框图和流程线来表示程序的执行过程。常用的符号如表 1-3 所示。可以看到，用流程图进行描述，直观、形象、易于理解，是目前使用较广泛的一种方法。

表 1-3　流程图常用的符号

图　形	名　称	说　明
	流程线	表示算法的流程方向
	开始、结束框	算法的开始和结束表示
	处理框 （矩形框）	表示算法的处理步骤
	输入输出框	表示原始数据的输入和处理结果的输出
	判断框	允许有一个入口，两个或两个以上的可选择出口

方法 3：用计算机语言——C 语言进行描述。

```
#include <stdio.h>
void main( )
{
    int x, y;
```

```
    scanf("%d, %d", &x, &y);          /* 输入两个整数依次存入 x,y 两个变量中 */
    if (x > y)                        /* x 和 y 比较 */
        {
        printf("最大值为:%d", x) ;    /* 如果 x>y，屏幕上显示的最大值为 x 的值 */
        }
    else
    {
        printf("最大值为:%d", y) ;    /* 否则，即 x≤y，屏幕上显示的最大值为 y 的值 */
        }
}
```

数据输入及程序运行结果：

2, 3↵

最大值为：3

说明：“↵”表示回车键。

【例 1-2】 求解 1+2+3+…+100。

方法 1：用自然语言描述。

步骤 1：先求 1+2，得到结果 3；

步骤 2：将步骤 1 得到的和加上 3，得到结果 6；

步骤 3：将步骤 2 得到的和加上 4，得到结果 10；

依次类推；

……

步骤 99：将步骤 98 得到的和加上 100，得到结果 5050。

方法 2：流程图法，如图 1-6 所示。

方法 3：用计算机语言——C 语言进行描述。

```
#include <stdio.h>
void main( )
{
        int i = 1, s = 0;
        while (i <= 100)
 /* 当 i<=100 时，执行 s = s + i */
        {
        s = s + i;
        i++;          /* i = i + 1 */
    }
    printf("1 +2 + 3 +  …+ 100 = %d", s) ;
}
```

程序运行结果：

1 + 2 + 3 + … + 100 = 5050

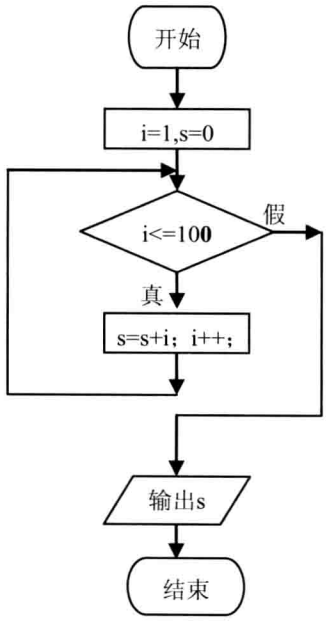


图 1-6 用流程图表示例 1.2 算法

1.1.3 结构化程序设计

程序设计是一门艺术，需要相应的理论、技术、方法和工具来支持。“清晰第一，效率第二”已成为当今主导的程序设计风格。 C 语言、Pascal 语言属于结构化语言的代表。

程序由以下三种基本结构组成：

① 顺序结构：按照从上到下的书写顺序依次执行语句。

② 选择结构：按照条件判断选择执行语句。

③ 循环结构：通过条件控制循环执行语句。

如果一个程序仅包含这三种基本结构，则称之为结构化程序。编写结构化程序的方法称为结构化程序设计方法。它的基本思路是：把一个复杂的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。其主要原则包括以下几个方面：

① 自顶向下，逐步求精。

② 模块化设计。

③ 限制使用 goto 语句。

1.1.4　面向对象程序设计

面向对象程序设计语言与结构化程序设计语言的根本不同点在于：前者的设计出发点是为了能直接地描述客观世界中存在的对象以及它们之间的关系。C++语言、Java 语言属于面向对象程序设计语言的代表。

面向对象的程序设计至少包含下面一些概念：

① 对象——是指现实世界中的一个实际存在的事物。它是面向程序设计的基本单元，具有属性（静态属性）和行为（动态属性）两个基本属性。例如，“张三”这个人就是一个对象，具有“性别”、“年龄”等属性和“说话”、“行走”等行为。

② 类——是具有相同属性和服务的一组对象的概括。例如，“张三”、“李四”等属性和行为相同的对象就构成了“人类”这个类。

类是对象的抽象，对象是类的实例（具体表现）。

③ 继承——一个类（称为子类）的定义可以在另一个已定义类（称为父类）的基础上进行，子类可以获得父类中的属性和行为，也可以加入自己的属性和行为，这种方式称为继承。例如，“大学生”这个类就是“人类”的子类，它继承了“人类”的所有属性和行为，也加入了自己的属性（如专业）和行为（如学习）。

1.2　C 语言的发展与特点

1.2.1　C 语言的出现与发展

C 语言是一种通用的、程序结构化、面向过程的计算机程序设计语言，它于 1972 年由 Dennis Ritchie 在贝尔电话实验室实现 Unix 操作系统时开发。C 语言不仅可用来实现系统软件，也可广泛用于开发应用软件。它还广泛使用在大量且不同的软件平台和不同架构的计算机上，而且几个流行的编译器都采用它来实现。C 语言还极大地影响了很多其他的流行程序设计语言，尤其是 C++程序设计语言，该语言是 C 语言的一个超集。

C 语言的起源与 Unix 操作系统的开发紧密相连。1969 年，美国贝尔实验室的 Ken Thompson 等人用 PDP-7 汇编语言编写了最初的 Unix 系统。接着，又对剑桥大学的 Martin Richards 设计的 BCPL（Basic Combined Programming Language）语言进行了简化，并为 Unix 设计了一种编写系统软件的语言，命名为 B 语言，并用 B 语言为 DEC PDP-7 写了 Unix 操作系统。B 语言简单而且很接近硬件，它是一种无类型的语言，直接对机器字操作，这和后来的 C 语言有很大不同。1972—1973 年间，贝尔实验室的 Denis Ritchie 改造了 B 语言，为其添加了数据类型的概念，并由此设计出来 C 语言。BCPL, B 和 C 全都严格符合以 Fortran 和 Algol 60 为代表的传统过程类型语言。它们都面向系统编程、小、定义简洁，以及可被简