教程 实验 练习

C语言程序设计基础与应用

杨开城 编著

人民邮电出版社
POSTS & TELECOM PRESS

教程 实验 练习

C语言程序设计基础与应用

杨开城 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计基础与应用 / 杨开城编著. -- 北京
: 人民邮电出版社, 2011.9
（教程·实验·练习）
ISBN 978-7-115-26165-6

Ⅰ. ①C… Ⅱ. ①杨… 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159975号

内 容 提 要

本书结合大量的应用实例和图表，全面深入地介绍运用 C 语言进行编程的知识与方法，包括 C 语言程序的基本构成，C 语言的基本数据类型和构造数据类型（数组、指针、结构体、共用体、位域和枚举）及其使用方法，运算符、程序控制语句的用法，预处理命令的用法，常见标准库函数（格式化输入输出、屏幕定位彩色输出、键盘读取、字符串操作、文件操作、内存分配与释放等）的用法。

为了激发和保持读者的学习兴趣，在每章的实验指导部分均针对所讲述的内容设计一些有趣的游戏类或界面类实验题目。为帮助读者强化巩固所学知识，每章还提供精心设计的习题，并在附录中给出部分习题的解析。

本书适合计算机专业的学生使用，也可以作为非计算机专业学生以及 C 语言零基础的学习者自学。

教程 实验 练习

C 语言程序设计基础与应用

◆ 编　　著　杨开城
责任编辑　李　莎

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1 092　1/16
印张：20.5
字数：502 千字　　2011 年 9 月第 1 版
印数：1－3 000 册　　2011 年 9 月北京第 1 次印刷

ISBN 978-7-115-26165-6

定价：38.00 元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

一、本书特色

本教材按照现代教学设计理念进行编写，不仅仅是知识单元的呈列，而是完整的学科能力培养。本教材具有以下特色。

1．注重编程能力的培养

目前大多数教材只罗列C语言的语法和应用实例，这仅能帮助读者理解C语言的概念和规则。但是，C语言的运用能力不是指在纸上写程序的能力，而是在编程环境中编写、调试程序的能力。本教材提供大量封装的程序，指导读者利用所学的各章节知识完成程序的某个模块的功能，帮助读者在编写、调试程序中提高编程能力，从而将学习C语言与编程能力的培养进行有机结合。

2．合理安排内容的逻辑顺序

C语言是具有良好结构的知识体系，但语法要素之间的练习又存在一定的结构不良特性。受此制约，不少教材往往在前面的章节中使用了后面章节才学习到的概念和术语，而且不给出详细的解释，导致读者学习起来有一定的障碍。本教材将线性的学习顺序和螺旋的学习顺序恰当地结合起来，科学地设计了知识单元出现先后的逻辑顺序。

3．丰富、有效又有趣的实验设计

本教材的实验设计有三大特色，一是采用 Borland C++作为编程环境，二是实验内容丰富、有趣，三是将实验内容做了难度区分以便读者进行有针对性的练习。

本教材坚持采用 Borland C++而不是 Visual C++ 6.0 作为编程环境，因为 Borland C++与商业开发环境相似，特别是概念、操作上的相似，读者能利用它编写出屏幕界面比较丰富的程序，能进行有效的逻辑思维训练。为了让读者掌握这一编程环境，本教材在前面几章重点介绍 Borland C++编辑器的使用与 Borland C++调试程序的方法，并设计了相应的实验内容训练读者的调试技巧。

本教材的每一章都设计了实验内容，并提供实验的具体任务及实验指导。与大多数教材大量使用数学问题作为编程内容不同，本教材选择了一些有趣的游戏软件和界面丰富的软件作为推荐的实验内容，如字幕动画、流星雨、推箱子、射击等。对于有点难度的实验，还提供了部分实现并在实验指导中有详细的编程提示。这无疑会大大提高读者的学习兴趣。

每一章基本上都有多个推荐的实验，并对实验难度进行了区分。不带星号的实验题目是较简单的，带一个星号的是难一点的，带两个星号的是难度最高的。同时，对于同一个实验，

有时也提供了难度等级不同的版本供读者选择。

4．设计了大量针对性强且具有难度区分的练习题

本教材根据章节内容的特点，有针对性地将练习题放入相应的小节中，以及时帮助读者巩固强化所学知识。同时，将练习题的难度做了区分，其区分方式与实验题目的相同。读者可以通过题目难度和自己的练习情况了解自己的掌握程度。

5．注意帮助读者克服错误的语言规则应用模式

C 语言中的知识类型主要是概念和规则，因此学习 C 语言的主要障碍之一就是要克服头脑中错误的语言规则应用模式。与大多数教材仅提供规则使用的正例不同，本教材不仅提供正例，还提供反例，而且每章都会总结并列举出读者常犯的错误。

二、学习建议

学习任何一种知识都会经历两个阶段：意义获取和能力生成。意义获取就是通常所说的“理解”，能力生成就是指将知识转化为解决实际问题能力的过程。理解是将知识转化为能力的基础。C 语言的学习也是一样。

读者首先要理解教材中讲解的内容，然后必须将自己所理解的知识付诸实践。建议读者按照下面的顺序组织自己的学习活动。

- 阅读教材和听教师讲解教材，特别要认真研读本教材中的例子程序。对于前面几章的例子程序，建议读者最好亲手输入 Borland C++中检验。对于后面几章的例子程序可以直接从下载的程序包中获取。
- 完成每一章的学习建议中的内容。
- 完成各章节的练习。
- 在理解例子的基础上，改写例子程序，既可将例子程序改写得更复杂，也可改得更简练。
- 重新编写例子程序。从例子的功能出发，重新编写整个例子程序，并将之与例子程序进行比对。
- 根据自己的能力水平选择合适的实验程序。在上机实验之前，最好在纸上写出程序再将其输入计算机。
- 复习一章的内容时，列出这一章的主要内容的提纲或示意图。

三、阅读对象

本教材适合软件学院、计算机专业的本科生、研究生及 C 语言爱好者使用。本教材深入浅出的讲解方式也适合非计算机专业的学生迅速、深入地掌握 C 语言的精髓。学习本教材的读者至少应学习过类似于“计算机基础”这样的前导课程。

本教材是杨开城先生从事 C 语言教学工作 20 余年的厚重积淀。希望本教材能对读者学习 C 语言有所助益，并能为 C 语言的教学工作进献一份绵薄之力。

欢迎广大读者与我们交流。本书作者的电子邮箱为 yangkc_beijing@foxmail.com，责任编辑的电子邮箱为 lisha@ptpress.com.cn。

编　者

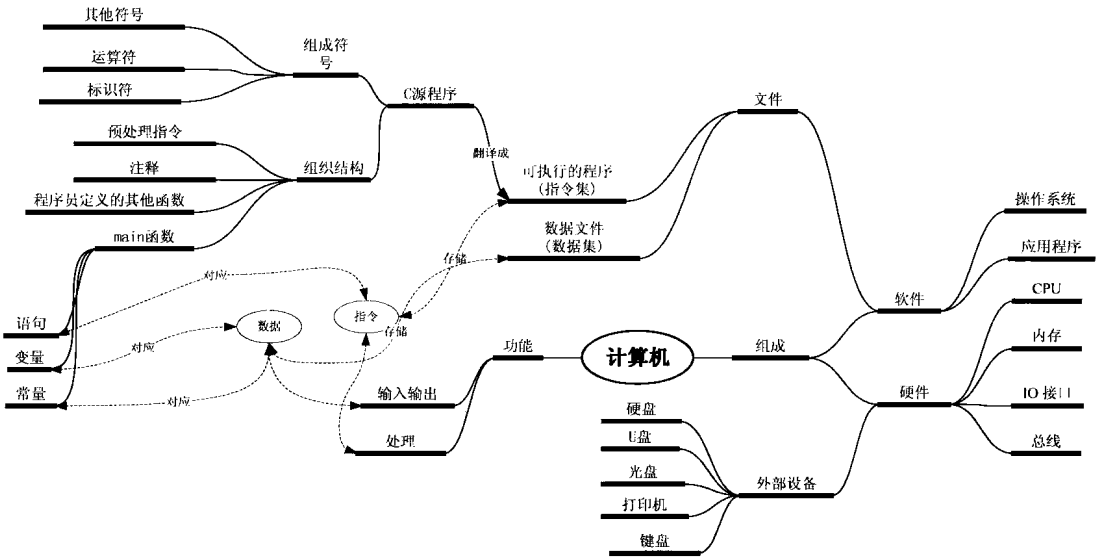
2011 年 8 月

目　录

第1章
C语言程序设计基础

学习建议

（1）下面的思维导图揭示了一些概念之间的关系。认真阅读本章，向这张图中增加下面一些概念以及连线，并连接成一张更大的思维导图。这些概念包括：入口、出口、函数定义、函数体、函数调用、注释符、字符串标志、变量名、函数名、声明部分、执行部分、字母、数字、下画线、分号、关键字（保留字）、大小写敏感、顺序执行、语句、参数。



通过画出这张图，读者应该明确“计算机”、“操作系统”、“程序”、“源程序”、“函数”、“变量”、“标识符”、“表达式”、“main 函数”、“数据”、“指令”、“运算符”之间的内在联系。

（2）将下面算术式中的十进制数转换成二进制、八进制和十六进制之后，进行算术计算，将计算结果转换为十进制数，并用十进制数进行验算。

25+17=　　1.25-0.05=

11×3=　　42÷6=

（3）认真完成本章的小练习。

（4）按照实验指导，完成所有的实验操作。如果读者的计算机中已经安装了 Borland C++，则跳过“Borland C++3.1 的安装”一节。

本章概览

C 语方程序设计基础
计算机工作原理
硬件系统
CPU
内存
输入输出接口
控制总线
地址总线
数据总线
外部设备
硬盘
显示器
打印机
...
软件系统
操作系统软件
应用软件
编程基本步骤
启动编程环境，输入源程序并保存
编译、连接，排除语法错误
试运行，排除逻辑错误
提交程序
二进制运算
加减
乘除
与 And
或 OR
异或 XOR
取反 NOT
移位
C 语言基本结构
C 语言源程序由多个函数组成
必须提供 main 函数的定义
main 函数是入口和出口
函数间的调用实现了程序的功能
函数名、变量名等必须是合法标识符
函数由声明部分和执行部分组成
函数声明部分定义变量，代表数据
函数执行部分的语句代表指令
数制转换
将十进制整数转化为二进制整数，除以 2 取余数
将十进制小数转化为二进制小数，乘以 2 取整数
进二进制数转化为八进制数，每 3 位成一组
将二进制数转化为十六进制数，每 4 位成一组
4 位二进制数从左向右各数位的权值是 8, 4, 2, 1

1.1　计算机的工作原理简介

计算机是一种用于计算和逻辑判断的设备，它处理计算和逻辑判断的速度是人所不能比的。无论计算机是进行计算还是逻辑判断，它都要处理数据。这些数据最终是由人从外部输入的。当计算机对这些数据进行处理后，还要将处理的结果输出。因此，简单地说，计算机就是一种数据的输入、处理、输出的设备。图 1-1 所示的计算机外围结构中，扫描仪、鼠标、键盘都是负责输入数据的，称为输入设备；打印机、屏幕（学名叫显示器）都是负责输出数据的，称为输出设备；集线器、调制解调器的功能更丰富，它们既可以用于输入数据，也可以用于输出数据，它们可以将单个计算机与计算机网络相连。

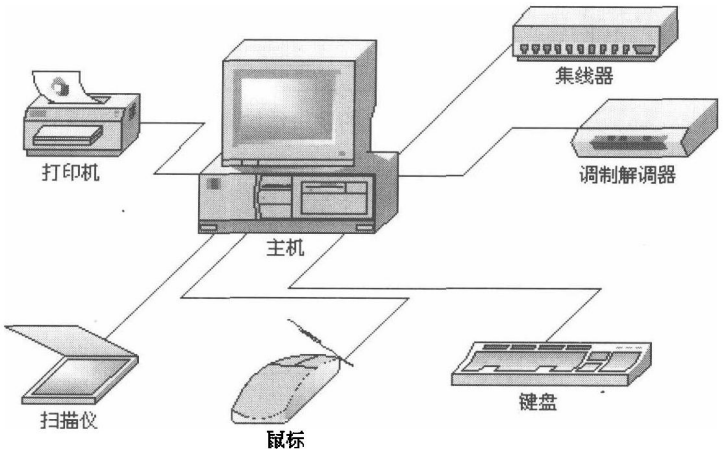


图 1-1　计算机设备示意图

计算机对数据的处理依靠两个系统来完成：硬件系统和软件系统。硬件系统是计算机执行输入、输出以及对数据进行处理的硬件设备和单元。图 1-1 所示的各种设备都是硬件。计算机的硬件是在软件的指挥下工作的。没有软件的支撑，计算机的硬件将无法工作。软件就是指挥硬件进行具体操作的指令集。

图 1-2 是计算机内部逻辑结构图。输入输出接口负责将数据输入内存中以及将内存中的数据输出。内存是一种计算机内部存储数据的硬件设备，由多个存储单元构成，这些存储单元是连续存放的，并且被编了号，这个编号称为内存单元的地址。计算机的其他硬件单元（如 CPU）将利用内存地址向内存中的特定单元存取数据。计算机所要处理的数据必须事先存放在内存中才能被处理。中央处理单元 CPU 负责管理其他硬件单元的工作，算术逻辑单元 ALU 负责完成计算和逻辑判断的工作。CPU 通过控制总线向其他硬件单元传递控制指令，通过地址总线来选择读取或写入的内存单元。计算机中的所有数据都通过数据总线进行交换。初学者可以将总线简单地看成是连接各部件的导线。

软件系统是各种程序的集合，一个程序基本上对应一个可执行的文件。这些文件用户可以查看到。打开 Windows 的“资源管理器”或者“我的电脑”，就会看到磁盘中存放的文件，如图 1-3 所示。这些文件有一些是可执行的程序，如后缀名是.exe 的文件；有一些是数据文

件，如后缀名是.ICO、.doc、.txt、.mp3、.ppt、.hlp 的文件。数据文件需要特定的程序解释才能被显示出来供人阅读。

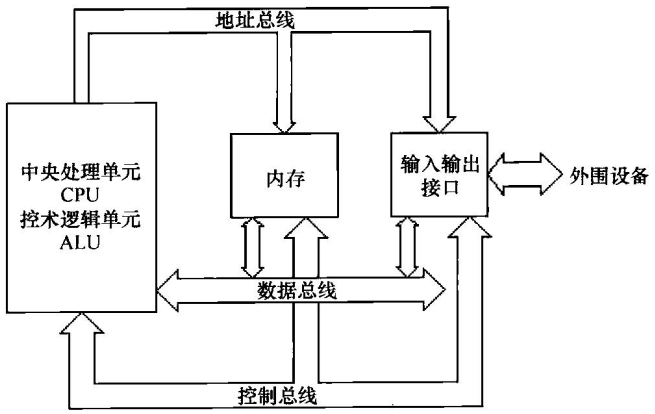


图 1-2　计算机的逻辑结构简图

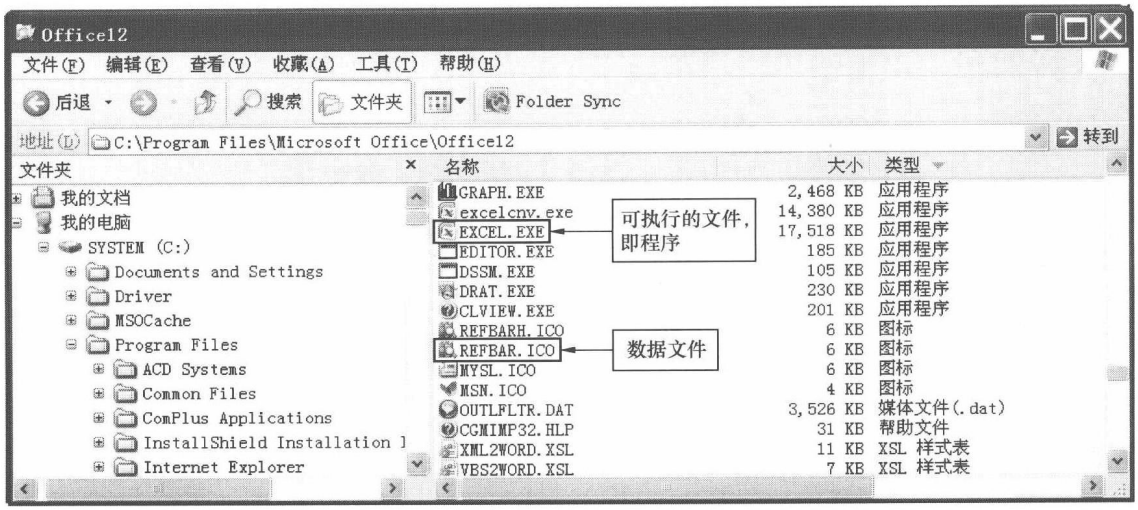


图 1-3　资源管理器中的文件列表

软件系统中，有一部分程序专门负责与计算机硬件打交道，负责管理计算机硬件资源，这部分程序被称为操作系统，属于系统软件。其他程序必须在操作系统的基础上运行，这部分程序被称为应用软件。初学 C 语言的用户的主要任务是学习如何编写应用软件。

综上所述，基本上可以这样理解计算机系统。

（1）计算机系统是一个输入—处理—输出数据的系统。

（2）计算机系统由软件系统和硬件系统组成，软件指挥硬件完成工作。

（3）软件系统是程序的集合。

（4）一个程序对应一个可执行文件，是计算机指令的集合。

（5）程序是人类控制计算机的手段，它是计算机进行数据处理的指令集。

1.2　C 语言的发展简史及其基本特性

为了使人类能方便地控制计算机的运行，人们发明了各种计算机程序的编制工具，计算机语言就是其中的一种。利用计算机语言编制程序的人被称为程序员。早先，程序员只能利用机器语言（即直接的机器指令）来编制程序。这样的程序难懂、易出错，只能局限于少量程序员阅读和使用。为了降低程序编制和维护的难度，人们又发明了汇编语言，利用特定的助记符来帮助程序员记忆机器指令。但利用汇编语言编制的程序通常不能是大规模的，在软件的开发和维护方面，汇编语言仍无法满足软件编制的需要。随着计算机语言的发展，人们发明了各种能直接表达计算式和逻辑式的语言，如 Pascal、Basic、C/C++、COBOL、Java 等。由于机器语言和汇编语言都是直接面向计算机的，与人们使用的自然语言有很大的区别，一般称为低级语言；而能够直接表达算式和逻辑式的语言，则相应地被称为高级语言。C 语言是众多高级语言中非常成功的一种。

C 语言是从 BCPL 和 B 语言演化而来的。1967 年，Martin Richards 开发出了 BCPL（Basic Compound Programming Language），其目的是编写操作系统软件和编译器。1970 年，Ken Thompson 在 BCPL 的基础上开发出了 B 语言，并在 DEC　PDP-7 计算机上利用 B 语言实现了第一个 UNIX 操作系统。

C 语言是贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发出来的，并逐渐应用于 UNIX 操作系统的系统软件和应用软件的开发。事实上，当今许多重要的操作系统和应用软件都是利用 C/C++来实现的。

随着 C 语言在各种计算机上的迅速推广，出现了许多不同的 C 语言版本。这些版本之间通常是不兼容的。这种情况给程序员开发适用于不同操作系统的程序带来极大的障碍。为此，1988 年，美国国家标准化协会制定了 C 语言的标准，这个标准称为 ANSI 标准或 ANSI C。本书的内容将基于 ANSI C 进行展开，但实验设计基于 Borland C++ 3.1 环境。

C 语言之所以能够生存和发展到今天，完全依赖于不同于其他语种的独特优势。简单地说，C 语言的主要特性包括以下几个方面。

（1）语言简洁、紧凑，使用方便、灵活。C 语言句法格式简洁，语法限制不严，程序书写极具美感。

（2）语言表述功能强大。C 语言一共有 30 多个关键字，9 种控制语句，运算符丰富，既有简单数据类型，又可以定义复杂数据类型，还允许直接访问内存的物理地址、进行位操作和强制类型转换。

（3）程序的可移植性好。C 语言提供了预处理器，可以利用预处理指令提高程序的可读性和可移植性。在一种操作系统下编写的程序，不需要修改或经过很小的修改，就可以在其他操作系统下重新编译运行。

（4）C 语言允许递归调用，在解决递归问题上具有独特优势。

1.3　C 语言程序的基本结构

本节的主要任务是了解 C 语言程序（简称 C 程序）的基本结构，即 C 程序是什么样子的。

一个 C 程序可以是非常简单的，也可以是特别复杂的，这取决于程序所要完成的功能。我们先来认识一个最简单的 C 程序（它的部分解释见图 1-4），争取凭借直觉和猜测就能看懂它。

【例 1-1】 第 1 个 C 语言程序。

```
/*This is the first C program*/

main( )
{

 printf("Welcome to C!");

 }
```

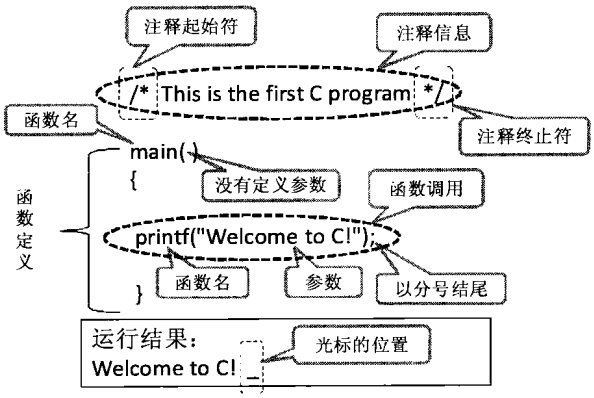


图 1-4 【例 1-1】的部分解释

- 程序【例 1-1】的运行结果是在计算机屏幕上显示文本“Welcome to C!”，光标停留在“!”的右边。在文本屏幕上不断闪烁的短线就是光标。光标是计算机文本屏幕上重要的对象，光标的位置就是程序输出文本的起始位置。光标还会随着文本的输出而变换位置。本书后续章节还会介绍如何控制光标的位置。
- 程序的第一行是注释信息。C 程序中，注释是程序员为了增加程序的可读性，人为增加的说明性信息。主要用来说明程序的功能、用途、符号的含义以及程序的逻辑等。计算机执行程序时，将忽略注释信息。注释不影响程序的功能。
- 注释由“/*”开始，由“*/”结束。在“/*”和“*/”中间放置注释的内容，具体写什么取决于程序员的意图。
- 从 main 开始到最后的}之间的部分称为函数，确切地说，是函数定义。main 是函数名。每一个函数都有名称。在同一个程序内部，不同函数之间不能重名。
- C 语言程序中，函数代表一小段可重复执行的计算机指令，执行一个函数被称为函数调用。函数可以带参数，也可以不带参数。
- 一个函数名后面跟着一对圆括号()，无论括号中间有无内容，都说明这个名称是函数名。如果()后面跟有一对大括号{}括起来的内容，则说明这是函数定义；如果后面跟的是分号或者其他内容，则说明这是函数调用。
- main 的后面跟着空的()，表明 main 函数没有参数。main()后面的{}中是函数的具体

内容，称为函数体。

- main 的函数体中只有一行文本。printf 也是一个函数名，它后面跟着()以及分号，因此这一行实际是一个函数调用。
- printf 函数实际上是 C 语言自己提供的函数，称为标准库函数。程序员可以随时根据实际需要调用它。
- main 函数中的 printf 函数调用的功能是：将“Welcome to C!”显示在计算机的屏幕上。文本“Welcome to C!”被放在了一对双引号中间。被双引号引起来的文本在 C 语言中称为字符串。"Welcome to C!"是字符串。
- 在 printf("Welcome to C!");中，字符串"Welcome to C!"称为函数 printf 的参数。

程序【例 1-1】非常简单，它所包含的新名词主要有：注释、函数、函数定义、函数调用、参数、标准库函数、字符串。下面通过【例 1-2】来认识一个稍微复杂一点的程序（它的部分解释见图 1-5）。

【例 1-2】 第 2 个 C 语言程序。

```
/* This is the second C program */

main( )
{
  int  a;

  a=20;
  a=a+40;
  printf("variable a = %d \n",a);
  }
```

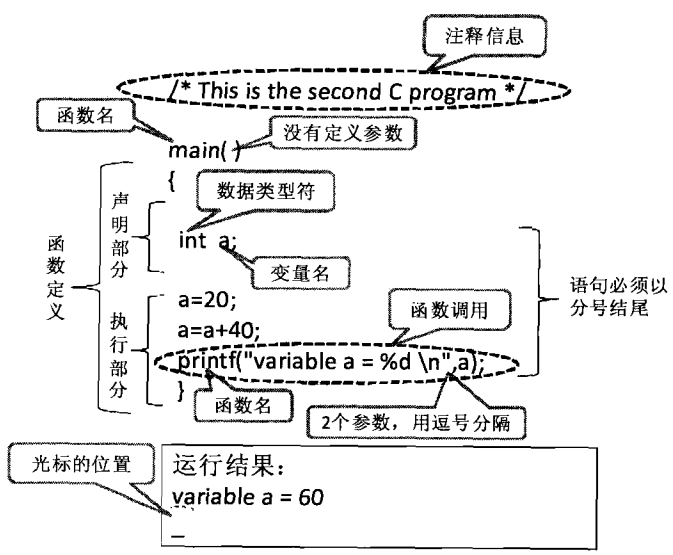


图 1-5 【例 1-2】的部分解释

程序【例 1-2】与【例 1-1】相比，多了 3 行文本，printf 函数的调用也复杂了一些。

- 【例 1-2】和【例 1-1】一样，整个程序就是一个 main 函数定义。但 main 函数中有了处理数据的功能。
- “int a;”表示定义了一个变量。C 语言是用变量来保存数据的。因此，如果程序要处理数据（通常是这样），就必须定义变量。
- “int”代表一种数据类型。C 语言根据数据的性质将数据区分为多种类型，并且为每种类型都规定了不同的操作。int（是 integer 的缩写）代表整数类型。符号 a 是变量名，这里代表一个整数值。C 语言规定，定义变量时，要以分号结尾。
- 当程序定义了变量时，计算机系统就为这个变量分配一块大小合适的内存单元，用来存放数据。
- 有了变量 a，程序就可以对 a 进行一定的操作。“$a=20$”表示将 a 的值设为 20。将一个变量设为某个特定值的操作，在 C 语言中被称为赋值，用“=”来表示。接下来的操作是“$a=a+40$”，它表示将 a 的值与 40 的和赋值给变量 a，这时 a 的值是 60。C 语言利用变量名来代表变量的值。在“$a+40$”中，a 表示变量 a 的值。“+”在 C 语言中表示相加，它是一种运算符。所谓运算符，即表示运算操作的符号。“=”在 C 语言中还被称为赋值运算符。
- 式子“a + 40”在 C 语言中被称为表达式。所谓表达式，就是指符合 C 语法规则的式子。表达式都有特定的值。
- 最后一行是 printf 函数的调用，有两个参数，第 1 个参数是一个字符串，第 2 个参数是变量 a。这两个参数用逗号分开。由于 printf 函数的功能是将第 1 个参数的字符串输出在计算机屏幕上，因此程序输出“variable a = ”可以理解，但为什么没有输出“%d”呢？对于这个问题，需要学完第 3 章后才能回答。这里只做这样的解释：%d 被变量 a 的值 60 代替了。
- printf 函数的第 1 个参数是一个字符串，这个字符串最后有一个“\n”，它代表换行符。printf 函数输出（或者说显示）换行符时，实际效果是将光标移动到下一行的行首。
- 函数 main 中除了“int a;”这一行之外，其他 3 行都代表程序对数据的操作。这 3 行以及定义变量 a 所在的那一行都被称为 C 语句。绝大部分 C 语句都能直接变成计算机指令，C 程序运行时，程序将按照顺序执行 C 语句。对于【例 1-2】来说，先执行 $a=20$，再执行 $a=a+60$。

注意：C 语言规定，语句必须以分号结尾。分号是 C 语句的结尾符。

- C 语言程序的函数由两部分构成，一部分定义变量（变量存放数据），称为声明部分；另一部分表示数据处理的操作，称为执行部分。函数的声明部分在前面，执行部分在后面，它们的顺序不能颠倒，也不能有交叉。下面的程序中包含有错误。

```
main()
{
 int a;

 a=20;
```

```
int b; /*错误！变量定义放到了执行部分了*/
b=a+30;
 }
```

【例 1-2】与【例 1-1】相比，可以说更具有真实感了，因为【例 1-2】能处理数据。具体地说，【例 1-2】定义了变量 *a* 来表示数据。然而，【例 1-2】在形式上还不能代表完整的 C 程序。我们需要花一点时间再认识一个程序【例 1-3】（它的部分解释见图 1-6），看看完整的 C 程序是什么样子的。

【例 1-3】 第 3 个 C 语言程序。

```
1.   /* This is the third C program */

2.   HaHaHa()
3.   {
4.    printf("^_^ \n");
5.    }

6.   main()
7.   {
8.     int  Money;

9.     Money=300;
10.    Money=(Money-30)*2;
11.    HaHaHa();
12.    printf("My pocket is full!\nI have %d dollars.",Money);
13.   }
```

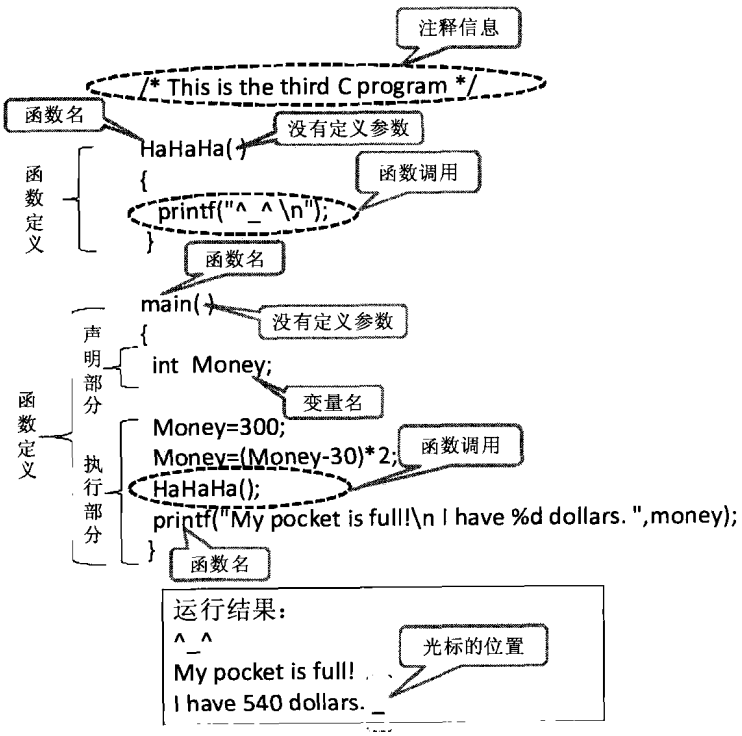


图 1-6 【例 1-3】的部分解释