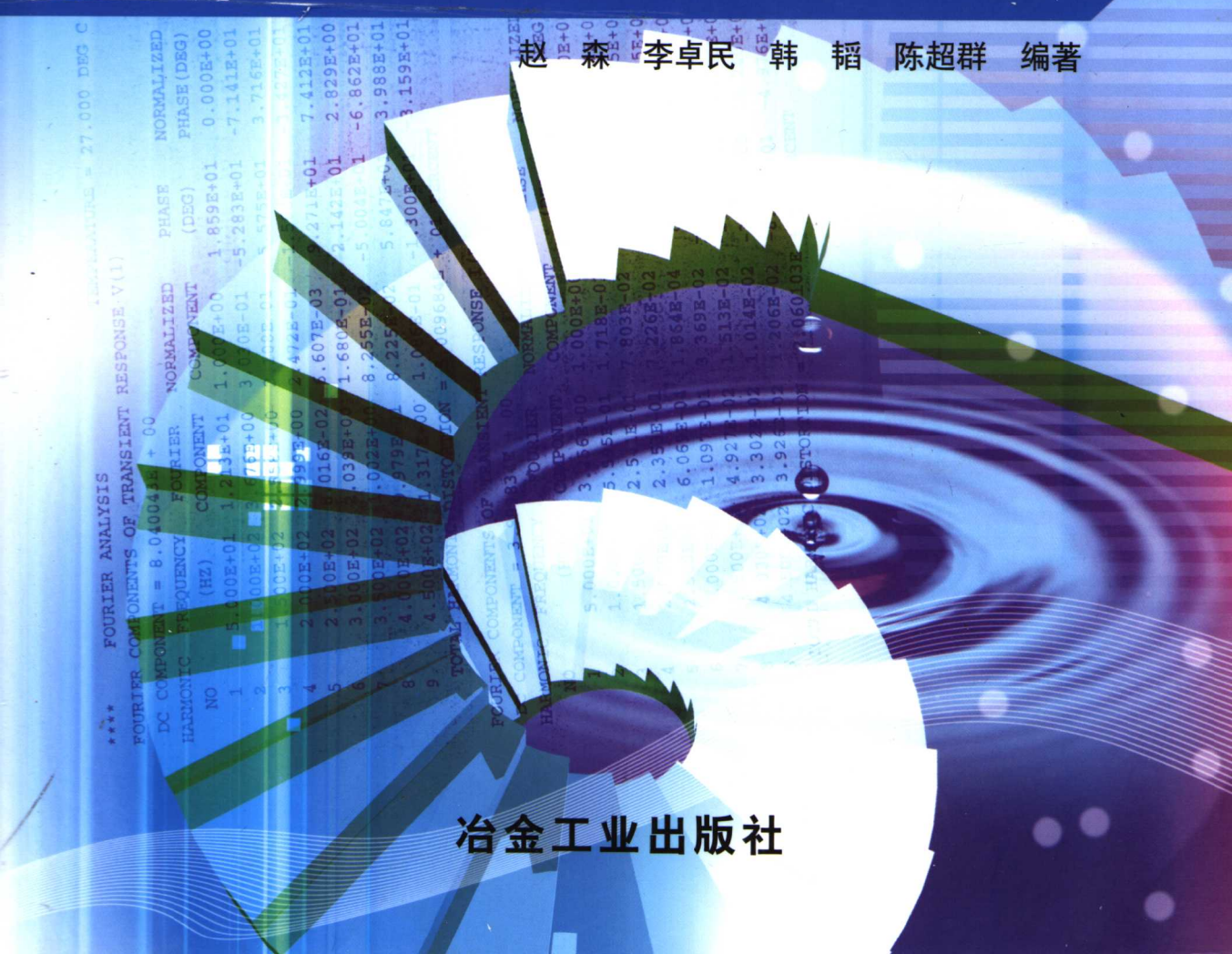
高职高专21世纪计算机规划教材

C程序设计

赵 森 李卓民 韩 韬 陈超群 编著



冶金工业出版社

高职高专21世纪计算机规划教材

C程序设计

赵　森　李卓民　韩　韬　陈超群　编著

北　京

冶金工业出版社

2005

内 容 简 介

本书全面介绍了C语言的基本概念和基本用法，并辅以大量的例题，使读者能通过本书掌握C语言的基本内容，并进行简单的程序设计。

本书主要内容有：程序设计基础与C语言、C语言的数据类型、运算符与表达式、三种结构的程序设计方法、函数、数组、指针、结构体与共用体、编译预处理、位运算、文件、字符屏幕和图形函数。本书内容编排合理、语言简练、通俗易懂，而且每章后面都提供大量的习题，供读者课后练习。

本书适用面广，既可作为各大专院校C程序设计教材，也可作为C语言的培训教材和C语言自学者的参考用书。

图书在版编目（CIP）数据

C程序设计 / 赵森等编著. —北京：冶金工业出版社，2005.2
ISBN 7-5024-3713-4

I. C… II. 赵… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字（2004）第142914号

出版人 曹胜利（北京沙滩嵩祝院北巷39号，邮编100009）
责任编辑 程志宏
湛江蓝星南华印务公司印刷；冶金工业出版社发行；各地新华书店经销
2005年2月第1版，2005年2月第1次印刷
787mm×1092mm 1/16； 16.75印张； 385千字； 260页
28.00元
冶金工业出版社发行部 **电话**：（010）64044283 **传真**：（010）64027893
冶金书店 **地址**：北京东四西大街46号（100711） **电话**：（010）65289081
（本社图书如有印装质量问题，本社发行部负责退换）

前　言

一、关于本书

C 语言是常用的结构化程序设计语言，在国内外都得到广泛的应用。C 语言以其功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好等优点，越来越受到人们的欢迎。

《C 程序设计》是计算机的一门基础课程，C 语言是计算机应用人员应掌握的程序设计语言之一，初学者往往感到比较难学。针对这一情况，作者结合多年教学、科研和应用经验，结合 C 语言程序设计的特点，对全书内容做了精心安排，分解难点、由浅至深，力求用通俗易懂的语言和丰富的例题清晰地解释 C 语言中各个复杂概念，使读者能轻松掌握该门语言。希望本书的出版问世，对普及 C 语言程序设计起到积极的作用。

二、本书内容结构

本书共分为 14 章，其内容结构安排如下：

第 1 章：程序设计基础与 C 语言。主要介绍了程序设计的概念、程序设计语言与 C 语言、程序设计风格等。

第 2 章：C 语言的数据类型。主要介绍了数据类型、基本数据类型、常量和变量等。

第 3 章：运算符与表达式。主要介绍了 C 语言的运算符、算术运算、赋值运算、关系运算、逻辑运算、条件运算。

第 4 章：顺序结构程序设计。主要介绍了标准输入/输出函数以及顺序结构程序设计的基本思想和方法。

第 5 章：分支结构程序设计。主要介绍了分支结构程序设计的思想和基本语句。

第 6 章：循环结构程序设计。主要介绍了循环结构程序设计的思想、基本语句以及程序举例。

第 7 章：函数。主要介绍了函数的概念、函数的定义与声明、函数的传值调用、函数的嵌套调用和递归调用、变量的存储属性、内部函数和外部函数等。

第 8 章：数组。主要介绍了一维数组、二维数组和多维数组、字符数组和字符串等。

第 9 章：指针。主要介绍了指针变量的定义和初始化、指针与数组、指针与字符串、指针与函数、返回指针值的函数、指针数组和指向指针的指针等。

第 10 章：结构体和共用体。主要介绍了结构体类型、用结构体指针处理链表、共用体类型、枚举类型、定义类型等。

第 11 章：编译预处理。主要介绍了宏定义、文件包含、条件编译等。

第 12 章：位运算。主要介绍了数值在计算机中的表示、位运算、位段等。

第 13 章：文件。主要介绍了 C 文件基础、文件类型指针、文件的打开与关闭、文件的读写、文件的定位、文件操作的检测、非缓冲型文件系统等。

第 14 章：字符屏幕和图形函数。主要介绍了 C 语言的工作模式、字符屏幕函数和图

形函数等。

附录给出了常用字符与 ASCII 码对照表、C 语言的关键字、运算符的优先级和结合方向、Turbo C 常用库函数表、Turbo C 2.0 使用指南。

三、本书特点

本书结构清晰、内容全面、通俗易懂，注重基础知识的介绍，是广大读者理想的 C 语言学习用书，读者如结合每章的练习进行学习，就能很快掌握 C 语言程序设计的基础知识和技巧方法。

四、本书适用对象

本书适用面广，既可作为各大专院校 C 程序设计教材，也可作为 C 语言的培训教材和 C 语言自学者的参考用书。

由于编者水平和经验有限，编写时间仓促，缺点和错误在所难免，希望广大读者批评指正。

虽然经过严格的审核、精细的编辑，本书在质量上有了一定的保障，但我们的目标是力求尽善尽美，欢迎广大读者和专家对我们的工作提出宝贵建议，联系方法如下：

电子邮件：service@cnbook.net

网址：www.cnbook.net

本书附送的电子教案和参考答案可在该网站的下载中心免费下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者参考选购。

编　者

2004 年 12 月

目　录

第 1 章　程序设计基础与 C 语言

“C”、“C++”、“汇编”、“Java”等都是人们经常遇到的术语，它们都是程序设计语言（Programming Language），并且各有各自的特点。利用这些程序设计语言可以设计出具有自己风格，又能满足不同用户不同需要的软件产品。那么，什么是程序设计语言呢？什么是 C 语言呢？本章将逐步回答这些问题。

1.1　程序设计的概念

本节将介绍程序、算法和数据结构的概念及相关知识，重点介绍算法的相关知识。

1.1.1　程序设计与程序

什么是程序设计呢？有人说程序设计就是用计算机语言编写程序。这是一个不严格的通俗的说法。要弄清楚什么是程序设计，首先要弄清楚一个概念——程序。什么是程序呢？在计算机领域，“程序”是指按照计算机程序设计语言规范书写出来的一系列语句，它表达了程序员要求计算机所要执行的操作。对于计算机来说，一组机器指令就是程序，如人们常说的机器代码或机器指令等；对于程序员来说，用某种高级语言编写的语句序列也是程序，如人们常说的源代码或源程序等。

著名的计算机科学家沃思（N.Wirth）曾提出过这样一个经典公式：

算法 + 数据结构 = 程序

该公式给出这样的信息：算法和数据结构是程序的两个重要方面。编写程序时，首先要明确程序操作的对象，即数据；数据结构就是对数据及它们之间的关系的描述。然后设计出对这些数据进行操作的具体步骤，算法就是对这些操作步骤的一种描述。

1.1.2　算法

本小节将重点介绍算法的概念与相关知识，包括算法的特性、构成要素、表示方法等。

1.　算法的概念

算法（algorithm）一词源于算术（algorism）。计算机没有像人类的思维能力，它只能机械地执行人们给它的指令和命令。如果要利用计算机解决某些问题，首先必须制定相关的解题方案，然后将方案转换为计算机能“理解”的语言输入计算机，使它按照人们的意思去执行相关指令进而解决问题。所以，对于制定的方案，其中的每一个细节都要准确地定义，并且把全部解决过程完整地描述出来。因此，“算法”就是指解决方案及对方案准确、完整的描述。

【例 1-1】输入一个整数，然后判断该数的正负。

分析：设 num 存放该整数。

要判断该数的正负，就必须与 0 进行比较，步骤如下：

（1）输入一个整数到 num 中，如果输入的不是整数，则提示输入错误，程序结束。

（2）将 num 和 0 比较，如果 num = 0，就可以输出“该整数为零”的提示信息，程序

结束。

（3）否则，如果 num > 0，就输出“该整数为正数”的提示信息，程序结束。

（4）到此可断定 num < 0，输出“该整数为负数”的提示信息，程序结束。

通过例 1-1 可以看出，对于每一个问题，首先要分析题目，然后制定解决这个问题的方案，并把该方案具体化为一个个的解题步骤（从（1）到（4）步），算法就是对这些步骤的描述。

2. 算法的特性

通过例 1-1 可以看出一个算法应具有以下 5 个特性：

1）有穷性

有穷性指每一个算法都必须总是（对于任何合法的输入值）在执行有穷的步骤之后结束，而且每一步都可以在有穷的时间内完成。

2）确定性

确定性指算法中每一条指令都必须有确切的定义，不会产生二义性。而且，在任何条件下，算法只有惟一的一条执行路径，即对于相同的输入不能有不同的输出。

3）可行性

可行性指每一个算法应该是可行的，即算法中所描述的操作都是可以通过已经实现的基本运算执行有穷的次数来实现。

4）输入

输入指每一个算法都有零个或者多个输入，它们是算法所需的初始量或者被加工的对象的表示。这些输入取自某个特定的对象集合。

5）输出

输出指每一个算法都有一个或者多个输出。它们是与输入有某些特定关系的量。

3. 算法的构成要素

算法含有两大要素：一个是操作，另一个是控制结构。

操作，即构成一个算法的操作取自哪个操作集。对计算机操作的描述与程序设计语言的级别有关，在高级语言中所描述的操作主要包括：算术运算（+、－、*、/）、逻辑运算（&、|、~等）、关系运算（==、>=、<=、>、<、!=等）、函数运算、位运算、I/O 操作等。

控制结构，即控制算法的各种操作的执行顺序。结构化程序设计方法规定，一个程序只能由顺序、选择和循环三种基本控制结构或由它们派生出来的结构组成。

（1）顺序结构：该结构中各个操作都是按照书写的顺序执行的，详细见后面第 4 章的内容。

（2）选择结构：根据指定的条件进行判断，根据判断结果在两条分支路径中选取其中的一条执行，详细见后面第 5 章的内容。

（3）循环结构：根据给定条件是否满足决定是否继续执行循环体中的操作，详细见后面第 6 章的内容。

另外，由上面三种基本结构还可以派生出其他（如“多分支结构”）的结构。

1966 年 Bohm 和 Jacopini 已经证明由这三种基本结构可以组成任何结构的算法，解决任何问题。

4. 算法的表示方法

常见的算法表示方法有：自然语言法、流程图法、N-S结构图法、伪代码法、PAD图法等。

1）自然语言表示法

就是使用人们日常使用的语言（如中文、英文等）来表示算法，例1-1就是使用了自然语言表示法。可以看出，这种表示法较其他表示法更通俗易懂，容易被人们接受；但是这种表示法往往不太准确，容易引起“歧义”。

2）流程图表示法

就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循序。流程图常用的符号如表1-1所示。

表1-1　流程图常用符号表

符号	符号名称	含义
	端点符	表示算法的开始或结束
	数据	表示数据的输入或输出
	处理	表示各种处理
	判断	表示对某个条件进行判断
	流线	表示执行的流程

如图1-1所示是用流程图表示的三种基本控制结构。

【例1-2】用流程图表示法描述例1-1的算法。

流程图如图1-2所示。

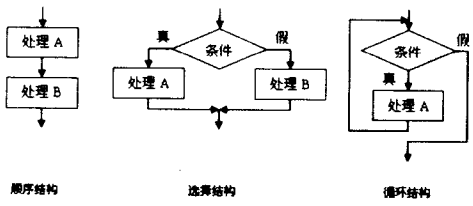


图1-1　流程图表示三种基本结构

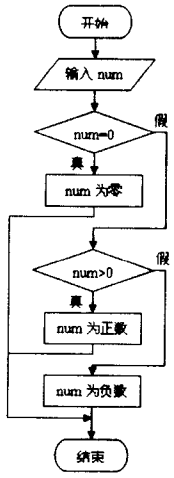


图1-2　例1-2流程图

可以看出，流程图表示法比较直观形象，便于理解；但是流程图一般占用的篇幅较大，画图繁琐，且由于它允许用流线使流程任意转移，从而会使程序如同乱麻似的难以阅读和维护。

3）N-S结构图表示法

N-S结构图是由美国学者I.Nassi和B.Shneiderman在1973年提出的一种无流线的流程图。这种结构图是针对流程图因灵活的流线带来的隐患而提出的，它完全取消了流线，不

允许有任意的控制流。1974 年 Chapin 又对 N-S 图进行了扩展。

如图 1-3 所示是用 N-S 结构图表示的三种基本控制结构。

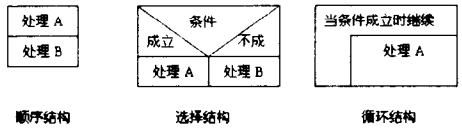


图 1-3 N-S 结构图表示基本控制结构

【例 1-3】用 N-S 结构图表示法描述例 1-1 的算法。

N-S 结构图如图 1-4 所示。

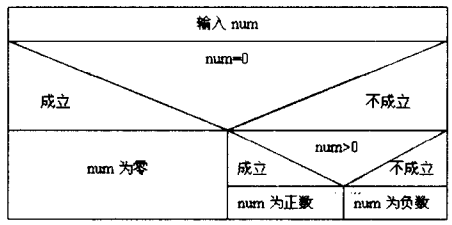


图 1-4 例 1-3 的 N-S 结构图

N-S 结构图既保留了流程图形象直观的特点，同时又去掉了容易导致程序混乱的流线，对于初学者来说比较适合；但是 N-S 结构图的修改不太方便。

4）伪代码表示法

就是用一种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。

【例 1-4】用伪代码表示法描述例 1-1 的算法。

（1）用自然语言把算法描述出来，主要是按顺序描述问题。

① 输入整数 num。

② 把 num 和 0 比较。

③ 打印结果。

（2）在前一阶段的基础上，把算法用伪代码表示出来。

```
input num;
if num = 0 then print "num是零";
else
   if num > 0 then print "num是正数";
   else print "num是负数";
```

（3）根据伪代码用 C 语言写出真正的程序语言代码。（略）

（4）检查完善程序。（略）

使用伪代码表示方法，不像用流程图或 N-S 结构图那样费事地画图；也不用拘泥于程序设计语言那固定、严格的语言规则；同时方便修改；也有利于人们把复杂困难的问题逐步细化，从而方便解决问题。

5）PAD 图表示法

PAD（Problem Analysis Diagram）图也就是问题分析图，它类似于 N-S 结构图，也没有流程图中的流线，并且规则地安排了这样的关系：从上到下表示执行顺序，从左到右表示层次关系。

如图 1-5 所示是用 PAD 图表示的基本控制结构。

【例 1-5】用 PAD 图表示法描述例 1-1 的算法。

PAD 图如图 1-6 所示。

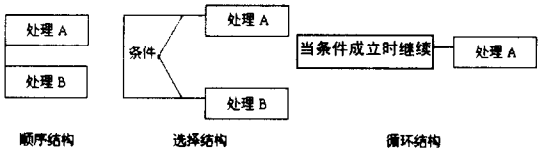


图 1-5　PAD 图表示三个基本控制结构

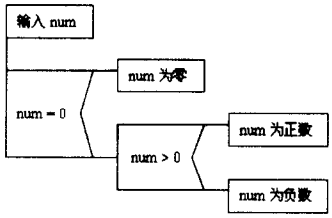


图 1-6　例 1-5 的 PAD 图

1.1.3　数据结构

数据结构是一门专门的课程，包含着十分丰富的内容，下面只是结合一个具体例子说明数据结构的概念。

【例 1-6】用计算机管理公司人事结构。

一般而言，在公司中每一个员工向上只有一个直接上级主管，而每个主管可以有一个或以上的下属员工，公司老总是没有上级的，而基层员工是没有下属的。如果用计算机来管理公司的人事结构，首先要记录每个员工的资料，然后就是员工之间的上下级关系，同时还允许通过计算机对员工资料进行相关操作，如通过某主管的资料就可以知道他的直接下属有哪些人，又如某个员工升职为主管后可以通过相关操作修改这个员工及相关员工的资料等。

在计算机中，可以用一个记录来表示每个员工，以记录其相关资料（记录由若干字段组成，每个字段分别记录姓名、年龄、职位等），为了表示员工之间的上下级关系，可以在员工记录中增加若干字段用于指示该员工的下属（基层员工该字段无内容）。这样，公司的人事结构就构成一个层次结构，如图 1-7 所示，在数据结构中，称该结构为树。

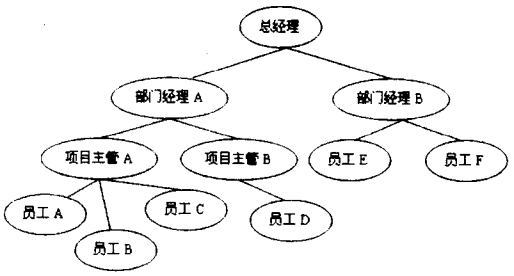


图 1-7　公司人事结构的树表示

由上例可以看出，问题世界是由各种各样的实体构成的，如上例中的员工；实体间一般不是孤立的，而是相互间存在一定的关系，如上例中员工之间的上下级关系；可以根据实体之间的关系对实体进行相关操作，如上例中的通过主管查找下属的操作。在计算机中，用数据元素（能独立、完整地描述问题世界中的实体的最小数据单位）来表征实体，用数据元素之间的关系（结构）表征实体间的关系。这样，就把相互之间存在着一定关系的数据元素的集合及定义在其上的基本操作（运算）称为数据结构。

1.2 程序设计语言与C语言

程序设计语言是人与计算机进行交流的工具。人们通过程序设计语言把自己的想法、意图、方案“告诉”计算机，让它们按照自己的意图执行相关操作。同时，人们也通过程序设计语言来控制计算机的工作。

C语言是一种程序设计语言，由于它具有很多优点，因而得到了广泛地应用。

1.2.1 程序设计语言的发展

伴随着计算机技术的日新月异、计算机硬件的更新换代，程序设计语言也在不断的升级，总的来说，程序设计语言可以分为三大类：面向机器的语言、面向过程的语言和面向对象的语言。

1. 面向机器的语言

面向机器的语言主要是利用计算机CPU（中央处理器）的指令系统进行编程。如机器语言，该语言的语句是由一组一组计算机可以识别的0/1序列组成，每一组其实表示计算机CPU指令系统中的一条指令，如图1-8所示。

10000000 加指令

图1-8 机器语言示例

编程时就是编写上述形式的指令，然后通过指令完成特定的功能。

机器语言对于计算机来说可以直接识别，执行程序的时候运行效率高；但是对于程序员来说，该语言不直观，指令难记、难辨，编程时容易出错；而且不同的计算机有不同的指令系统，通过机器语言就不可能编写出适合各种机器的“通用”程序。面向机器的语言还有汇编语言。

2. 面向过程的语言

面向过程的语言向程序员提供了一系列人们易于理解的编程规范、机制、编程语言，而不是具体的机器指令，程序员编程时按照这些编程规范、机制把解决问题的步骤用编程语言描述出来，告诉机器“如何做”，而不用考虑具体的机器将如何利用自己的指令去完成相应的功能。C语言就是一种面向过程的语言，另外，PASCAL、FORTRAN语言等也是面向过程的语言。

3. 面向对象的语言

面向对象的语言能使程序比较直接地反映客观世界的本来面目，并且使软件开发人员能够运用人类认识事物时所采用的一般思维方法进行软件开发。C++、Java等就属于面向对象的语言。

1.2.2 C语言简介

C语言是一种广受欢迎的程序设计语言，属于面向过程的语言。

1. C语言的发展

C语言的前身是Martin Riohards于20世纪60年代开发的BCPL语言；70年代UNIX的研发者Ken Thompson对BCPL语言进行改进和发展，形成B语言；此后美国Bell研究所的Dennis Ritchie和Brian Kernighan对B语言进一步充实和完善，推出了C语言。

C语言的快速推广使C语言出现了许多不同的版本。1978年Dennis Ritchie和Brian Kernighan合作推出The C Programming Language的第一版本，书末的参考指南给出当时C语言的完整定义，成为那个时候C语言实际上的标准，称为“K&R C”。1983年ASC X3成立了一个专门的技术委员会J11，起草了关于C语言的标准草案，1989年被ANSI正式通过成为美国国家标准，称为“C89”。此时The C Programming Language（2nd Ed）根据当时最新的ANSI C进行了更新，1990年被ISO批准为国际标准，称为“C90”。随后ISO对C90进行了补充和扩充，于1995年出台“C95”。1999年，ANSI和ISO通过并发布了新版本的C语言标准，称为“C99”。

2. C程序的程序结构

程序结构也就是程序中语句的安排、顺序等，下面通过两个例子来讨论C程序的程序结构。

【例1-7】显示“This is my first program.”。

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

运行结果如图1-9所示。

```
This is my first program.
```

图1-9　例1-7的运行结果

说明：

① C程序是由一个或若干个函数组成的，每一个函数完成相对独立的功能。每个C程序都必须有（且仅有）一个称为main()的函数，称为“主函数”，程序的执行总是从主函数开始的。

② 对于main()来说，main是函数名，函数名后面一对圆括号内是函数的参数，如下面的printf()函数括号内的“This is my first program.\n”就是该函数的参数，main()函数没有参数所以不写（其实main函数是有参数的，后面第7章会讲到）。main()后面一对花括号内的部分称为函数的函数体。一般而言，函数体由说明部分和执行部分组成（详细见后面第7章的讲解部分）。说明部分用来定义数据类型；执行部分用来给出实现函数功能的操作。本例的函数体中只有一个语句，是执行部分。

③ C语言规定程序中的每个语句必须以分号（;）结束。

④ 程序中的printf()函数是C语言函数库中的一个函数，作用是在屏幕上输出指定的内容（printf函数的详细介绍见后面第4章）。C语言提供了强大的函数库，包含了丰富的库函数，这些函数已经设计好，每个函数都实现特定的功能，只要按函数名就可方便地调用这些函数，如本例的printf函数，可以直接调用它而不必设计实现它的函数体。当然，在调用这些函数之前，要通过称为编译预处理命令的#include，将一些后缀名为“.h”文件包含到C文件中。后缀名为“.h”的文件称为“头文件”，头文件中定义了库函数所用到的宏和变量，如stdio.h就定义了I/O库所用到的宏和变量（详细见后面第11章的介绍）。

下面给出一个比上例复杂一点的例子来进一步讨论C程序结构。

【例 1-8】求三个整数中的最小值。

程序如下：

```
#include <stdio.h>
main()
{
    int value1, value2, value3;                          /*变量定义*/
    int minValue;
    int GetMin(int x, int y, int z);                     /*函数声明*/
    printf("Please enter 3 integers:\n");
    scanf("%d, %d, %d", &value1, &value2, &value3);      /*输入三个整数*/
    minValue = GetMin(value1, value2, value3);           /*函数调用*/
    printf("MinValue is %d\n", minValue);
}
int GetMin(int x, int y, int z)                          /*函数定义*/
{
    int minValue = x;                                    /*假设第一个数为最小*/

    if (y < minValue)                                    /*如果第二个数小，就赋值给 minValue*/
        {
            minValue = y;
        }
    if (z < minValue)                                    /*如果第三个数小，就赋值给 minValue*/
        {
            minValue = z;
        }

    return minValue;                                     /*将最小值作为函数返回值*/
}
```

运行三次的结果如图 1-10 所示。

说明：

① 本程序由两个函数组成，main()就是上面提到的主函数，是程序的入口点和开始点；另一个 GetMin()是一个由用户自己定义（即自己设计、实现）的函数，作用是从三个整数中选出值最小的一个整数。该程序函数调用流程如图 1-11 所示。

```
Please enter 3 integers:
1,2,3
MinValue is 1
Please enter 3 integers:
3,2,1
MinValue is 1
Please enter 3 integers:
2,1,3
MinValue is 1
```

图 1-10 运行三次的结果

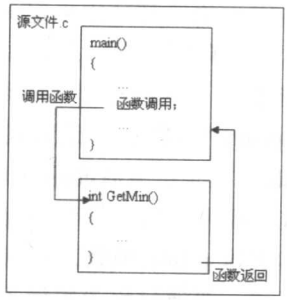


图 1-11 例 1-8 函数调用流程图

② 在 main()函数体中，前三行为函数的说明部分，这里定义了四个整型变量（变量及变量的定义见第 2 章），还声明了下面要用到的函数（函数声明见第 7 章）。剩下的四行是函数的执行部分，这里实现了调用自定义函数选出三个整数中最小者的功能。读者自行分