

普通高等院校**计算机科学与技术**“十二五”规划教材

辽宁省教学改革与创新成果奖

PROGRAM DESIGN in C

C语言 程序设计

(第2版)

葛日波 何 毅 / 编著
刘丽艳 付 蓉



北京邮电大学出版社
www.buptpress.com

普通高等院校计算机科学与技术“十二五”规划教材
辽宁省教学改革与创新成果奖

C 语言程序设计

(第 2 版)

葛日波 何 毅 刘丽艳 付 蓉 编 著



北京邮电大学出版社

· 北京 ·

内 容 简 介

本书是以 C 语言零起点读者作为主要对象的程序设计教程。2008 年 8 月出版了第 1 版,本次再版融入了新的教学与改革成果,更加突出了“教师方便教,学生容易学”的特点。

本书紧密结合技术和应用背景,按照“从问题中来——到问题中去”的思路进行精心策划,用大量的图表和程序实例将知识融入其中,大大降低了知识的抽象性和复杂性,符合读者的认知规律,很容易入门。每章后配有大量的习题,有助于读者巩固和提高。

本书结合作者多年的教学与改革实践编写而成,内容先进,体系合理,概念清晰,讲解透彻,图例丰富,分散难点,文字流畅,通俗易懂,是学习 C 语言程序设计的一本好教材。本书还配有《C 语言程序设计习题解答与上机指导(第 2 版)》、《C 语言程序设计习题与详解》两本辅助教材。

图书在版编目(CIP)数据

C 语言程序设计/葛日波等编著. --2 版. --北京:北京邮电大学出版社,2013.8
ISBN 978-7-5635-3551-4

I. ①C… II. ①葛… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 154117 号

书 名: C 语言程序设计(第 2 版)

著作责任者: 葛日波 何 毅 刘丽艳 付 蓉 编著

责任编辑: 毋燕燕

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发行部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京鑫丰华彩印有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 23.25

字 数: 559 千字

版 次: 2008 年 8 月第 1 版 2013 年 8 月第 2 版 2013 年 8 月第 1 次印刷

ISBN 978-7-5635-3551-4

定 价: 48.80 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

C语言是目前国内外广泛使用的计算机语言,是高等院校普遍开设的程序设计课程。针对应用型人才培养要求,以零起点的读者作为主要对象,作者编写了《C语言程序设计》一书,于2008年8月由北京邮电大学出版社出版。该书出版后受到了广大读者的欢迎,认为内容编排独特,组织形式新颖,概念清晰,叙述详尽,深入浅出,通俗易懂,很容易入门,被不少高校选做了教材。为了帮助读者学习,作者同时编写了《C语言程序设计习题解答与上机指导》和《C语言程序设计习题与详解》,并于2009年8月由北京邮电大学出版社出版。《C语言程序设计习题解答与上机指导》包括三部分内容,第一部分是对《C语言程序设计》一书中所有课后习题进行了解答,以方便读者学习时参考。第二部分是上机预备知识,介绍了自上而下程序设计的方法,使用VC++ 6.0环境上机编程的方法以及常见编译错误信息,为读者独立上机编程进行指导。第三部分是上机练习与编程,与《C语言程序设计》一书的章节同步,编排了9次上机练习与编程任务,以方便读者选做,通过练习来巩固知识,提高能力。《C语言程序设计习题与详解》与《C语言程序设计》一书的章节一致,精编了400多道选择和填空题,并对每个题目进行了详细解析,以帮助读者更好地理解知识。三本教材相互补充、相互完善,形成了立体化体系。

作者一直从事程序设计类课程的教学与研究,先后主持多项课题——2010年主持开发的“C语言程序设计电子课件”获辽宁省第11届教育软件设计大赛三等奖;2012年主持申报的“面向能力和素质协调发展的C语言程序设计课程教学改革与创新”获辽宁省教学成果三等奖。根据第1版的使用情况以及最新教改成果,在保持第1版内容框架和特色的基础上,对教材进行了全面修订,把新的教改成果融入了教材中。修订的主要内容有:

(1)对全书的文字部分进一步提炼,文字阐述更加精练、直白,使读者理解起来更容易。

(2)对书中所涉及前后有关联的内容,均给出了相关部分所在的准确页码,不仅方便读者快速查阅,还有利于读者掌握前后内容之间的联系。

(3)考虑到国内当前情况,书中的理论及程序全部修订为基于32位Windows操作系统以及Visual C++ 6.0编译环境,避免了因系统和环境不同而造成的混乱。

(4)每个程序实例的运行结果全部更换为真实运行情形的截图。这既保证了所有程序代码均经过调试完全正确,也方便读者在练习时通过比较运行结果来发现、分析和解决问题,加深对知识的理解,提高独立解决问题的能力。

(5)每个程序实例的后面增加了“程序分析”部分,详细解读程序实现的方法、技巧以及所用到的知识。这样就把知识融入了一个个看得见、摸得着的具体应用,以帮助读者更好地理解知识,掌握解决问题的方法和技巧。

(6)新增和调整了部分内容。

①第1章中增加了“1.2.3 C程序风格”和“1.8 使用VC++ 6.0环境上机编程”两部

分。使读者在学习之初就明确编写程序需要遵循的一些基本原则,养成良好的编程习惯,掌握使用 VC++ 6.0 开发环境上机编程的简单方法。

②第2章更改为“表达式和程序设计基础”,原来的“2.6 标准库函数”、“2.7 简单程序设计”分别更改为“2.6 部分库函数”、“2.7 程序设计基础知识”。

在“2.7 程序设计基础知识”中增加了“程序与程序设计”、“算法及其描述”以及“一个完整的程序设计实例”三部分。读者通过这部分的学习可以了解程序设计的大致过程,明确算法设计的重要性、算法的三种基本结构以及使用流程图和 N-S 图描述算法的方法。最后通过一个例子介绍了“自上而下”进行算法设计的过程,使读者掌握算法设计的基本方法,提高程序设计能力。此外,新增加的内容使第2章到第3章、第4章的过渡更加自然。

③考虑到许多学校把本课程放在一年级学习,学生之前缺乏计算机方面的知识,对命令窗口等操作不熟悉,在学习“8.12 命令行参数”时比较困难。为此,增加了“附录 F 命令窗口”,通过一个程序实例详细介绍了如何通过命令窗口实现向 C 程序传递参数的方法。

(7)同步修订了《C 语言程序设计习题解答与上机指导》。在原有第1版基础上增加了“第4部分 综合实训”,提供了“学生成绩管理系统”的完整实现过程,读者通过学习可以进一步熟悉如何综合运用所学知识来设计和编写应用程序。同时编排了20个实训题目,以方便师生在实际教学中选做。该书将与本书一起同时由北京邮电大学出版社出版。

(8)提供了丰富的教学资料,可以协助教师进行课前准备、课堂教学、上机实验以及实训,尤其是对本课程零起点的教师,可以使其轻松上手。所有教学资料可以到 <http://www.buptpress.com> 下载专区下载。

根据作者的教学体会,给出使用本教材的几点建议:

本书内容比较丰富,建议采取课堂讲授与自学相结合的方式组织实施,讲授以 72~100 学时为宜。

(1)要善于设计和使用引言。利用引言把知识前后的脉络讲清楚,使学生明确已学的知识遇到了什么新问题?新的问题需要如何去解决?这样学生带着问题进入新知识的学习,可以激发他们的兴趣,使他们清楚新知识的来由以及前后知识之间的联系,便于他们更加系统地掌握和使用。本书在每一章的开始都加了引言,电子教案中每一讲的开始部分也设计了引言,大家可以参考。

(2)要注意选取有一定连续性的程序实例。讲授时采用“实例驱动”的方式,先从实例中引出问题,带着问题去讨论知识,然后再回到编程实现环节。这样抽象的知识就与一个个生动的实例结合在一起,不再是抽象的教条和生硬的语法。尽量采用现场编程的方式进行,不要拿提前准备好的代码来讲。通过边演示边讲解的方式介绍编程思路和算法实现,通过演示把关键点和需要注意的事项揭示出来,学生不仅可以很容易地接受知识,还可以掌握算法设计的方法以及上机编程与调试的技巧。

(3)要善于利用习题。本书各章包括不同类型、不同难度的习题 266 道,在《C 语言程序设计习题解答与上机指导(第2版)》一书中,提供了所有习题的参考解答和程序代码。这些是对教材中例题的拓展,希望读者能充分利用它。在练习时,尤其是编程题,建议先独立思考,实在解决不了就阅读参考解答,待问题搞清楚后,再选择类似的问题独立设计和完成编码,这样举一反三,就一定可以学好 C 语言。

此外,为了帮助读者上机练习以及检验学习效果,在《C 语言程序设计习题解答与上机指导(第2版)》一书中编排了9次上机与编程任务,给出了不同类型的题目67个,这些题目没有给出参考答案,目的就是方便老师以作业形式布置给学生课后独立完成。

(4)有条件的院校最好安排一次课程实践。由于受学时的限制,只靠课堂上的时间就使学生学好C程序设计是十分困难的,如果条件允许,最好在学完本教材后安排一次课程实践,这个环节很重要,可以大大提高学生的编程能力。

本书由葛日波主编,何毅、付蓉、刘丽艳参与编写,朱志刚、张治海、闫会娟承担了部分工作。在编写本书的过程中,得到了杨元生教授的热情指导和帮助。

感谢广大读者给予我们的理解与厚爱,感谢北京邮电大学出版社的鼎力支持与密切合作。

由于作者水平有限,书中难免有缺点和不足,殷切期望专家和读者批评指正。

作者电子邮件地址:ecocity@dlut.edu.cn

作者

2013年6月

目 录

第 1 章 C 语言初步	1
1.1 C 语言的背景	1
1.2 C 程序结构	2
1.2.1 程序结构	2
1.2.2 两个程序例子	4
1.2.3 C 程序风格	6
1.3 标识符	7
1.4 数据类型	9
1.5 变 量	11
1.6 常 量	14
1.6.1 常量的种类	14
1.6.2 常量的用法	18
1.7 输入/输出	20
1.7.1 格式化输出	20
1.7.2 格式化输入	24
1.7.3 字符输入/输出	27
1.8 使用 VC++ 6.0 环境上机编程	29
1.8.1 创建 C 源程序文件的简单方法	29
1.8.2 编辑源代码	32
1.8.3 编译程序	33
1.8.4 链接程序	33
1.8.5 运行程序	35
习 题	36
第 2 章 表达式和程序设计基础知识	41
2.1 表达式	41
2.1.1 初级表达式	42
2.1.2 二元表达式	43
2.1.3 赋值表达式	45
2.1.4 后缀表达式	47

2.1.5 一元表达式	49
2.1.6 逗号表达式	51
2.2 副作用	52
2.3 表达式求值	53
2.3.1 无副作用的表达式求值	53
2.3.2 有副作用的表达式求值	53
2.4 混合类型表达式	55
2.4.1 隐式类型转换	56
2.4.2 强制类型转换	57
2.5 语 句	59
2.6 部分库函数	60
2.7 程序设计基础知识	65
2.7.1 程序与程序设计	65
2.7.2 算法及其描述	66
2.7.3 一个完整的程序设计实例	68
习 题	78
第3章 分支程序设计	82
3.1 关系与逻辑运算	82
3.1.1 逻辑数据	82
3.1.2 关系运算	82
3.1.3 逻辑运算	84
3.2 两路分支	88
3.2.1 if...else 语句	88
3.2.2 if 语句	91
3.2.3 if 语句的嵌套	91
3.2.4 条件表达式	93
3.3 多路分支	93
3.3.1 switch 语句	93
3.3.2 if...else-if 语句	98
习 题	102
第4章 循环程序设计	110
4.1 循环概述	110
4.2 while 语句	112
4.3 for 语句	115
4.4 do...while 语句	118
4.5 循环的嵌套	121

4.6 break 与 continue 语句	130
习 题	134
第 5 章 数 组	141
5.1 概 念	141
5.2 一维数组	143
5.2.1 定义数组	143
5.2.2 访问数组元素	144
5.2.3 用数组存数据	145
5.3 顺序查找	149
5.4 冒泡排序	151
5.5 二维数组	154
5.5.1 定义数组	155
5.5.2 访问数组元素	155
5.5.3 用数组存数据	156
习 题	159
第 6 章 指 针	163
6.1 概 念	163
6.1.1 指针常量	163
6.1.2 取地址运算	163
6.1.3 指针变量	164
6.2 指针与变量	165
6.2.1 定义指针变量	165
6.2.2 指针的初始化	165
6.2.3 用指针处理变量	167
6.2.4 程序举例	168
6.3 多级指针	173
6.4 指针与数组	176
6.4.1 指针与一维数组	176
6.4.2 指针与二维数组	183
6.5 动态内存分配	189
6.5.1 概 念	189
6.5.2 内存管理函数	190
习 题	194
第 7 章 字 符 串	201
7.1 字符串的存储	201

7.2 指针与字符串	202
7.3 字符串输入/输出	203
7.4 字符串处理函数	210
习 题	217
第8章 函 数	220
8.1 结构化程序设计	220
8.1.1 自上而下程序设计	220
8.1.2 C程序的结构化	221
8.2 函数定义	222
8.2.1 函数定义格式	222
8.2.2 函数定义举例	223
8.3 原型声明与函数调用	224
8.3.1 原型声明	224
8.3.2 函数调用	225
8.4 向函数传值	228
8.5 向函数传地址	228
8.6 向函数传数组	234
8.6.1 传一维数组	234
8.6.2 传二维数组	237
8.7 向函数传字符串	240
8.8 指针型的函数	241
8.9 函数指针	243
8.10 作用域	245
8.11 变量的存储类型	246
8.12 命令行参数	250
8.13 递 归	252
习 题	259
第9章 结构、联合与枚举	266
9.1 类型定义	266
9.2 结构类型	267
9.2.1 结构的声明	267
9.2.2 定义结构变量	269
9.2.3 结构指针	270
9.3 结构的处理	271
9.3.1 结构变量的初始化	271
9.3.2 访问结构成员	271

9.4 复杂结构	276
9.4.1 嵌套结构	276
9.4.2 含数组的结构	279
9.4.3 含指针的结构	283
9.5 结构数组	285
9.6 结构与函数	286
9.7 联合	300
9.8 枚举	304
习 题	306
第 10 章 文 件	313
10.1 文件概述	313
10.1.1 文件的概念	313
10.1.2 文件的分类	313
10.1.3 文件和流	314
10.1.4 文件指针	314
10.1.5 文件操作的基本步骤	314
10.2 文件的打开与关闭	315
10.2.1 文件的打开	315
10.2.2 文件的关闭	316
10.3 文件操作	317
10.3.1 文本文件读写函数	317
10.3.2 二进制文件操作函数	325
10.4 其他函数	327
习 题	331
附录 A ASCII 码表	335
附录 B C 语言中的运算符	336
附录 C C 语言库函数	337
附录 D 位运算	345
附录 E 预处理命令	347
附录 F 命令窗口	354
参考文献	357

第 1 章 C语言初步

C语言是一种功能很强的高级计算机语言。本章主要介绍C语言的发展背景、C程序的基本组成要素与C程序的结构、数据类型、常量、变量、数据的输入/输出等内容。

1.1 C 语言的背景

C语言是目前国际上广为流行的一种通用的结构化程序设计语言(Programming Language),它不仅是开发系统软件(System Softwares)的程序设计语言,也是开发应用软件(Application Softwares)的理想工具。

C语言的发展根源可以追溯到 ALGOL。1967 年 Martin Richards 推出了 BCPL(Basic Combined Programming Language)。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础设计出了 B 语言,并用 B 语言编写了第一个 UNIX 操作系统(Operating System)。由于 B 语言过于简单,所以其功能有限。1972 年,Dennis Ritchie 在 B 语言的基础上开发出了 C 语言。C 语言既保持了 BCPL 和 B 语言的优点,又克服了它们的缺点。C 语言的发展过程如图 1-1 所示。

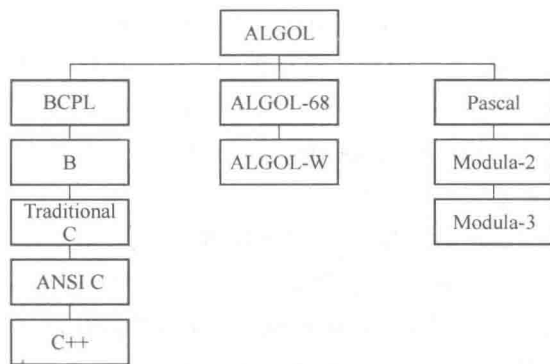


图 1-1 C 语言的发展过程

1978 年,Brain W. Kernighan 和 Dennis Ritchie 合著了 *The C Programming Language* 一书,被称为传统 C 语言。1983 年美国国家标准化协会(American National Standard Insti-

tute)制定了C语言标准,称为ANSI C,并在1989年通过认定。1990年,国际标准化组织(International Standard Organization,ISO)接受了ANSI C标准。本书使用ANSI C标准。

1.2 C 程序结构

1.2.1 程序结构

每个C程序都由一个全局声明区(Global Declaration Section)和一个或多个函数(Functions)组成,其中必须有且只能有一个称作main的函数,图1-2是典型的C程序结构示意图,其中图1-2(a)是只含有一个main函数的简单结构,图1-2(b)是包含了两个函数的程序结构,一个是main函数,另一个是称作fun的函数。

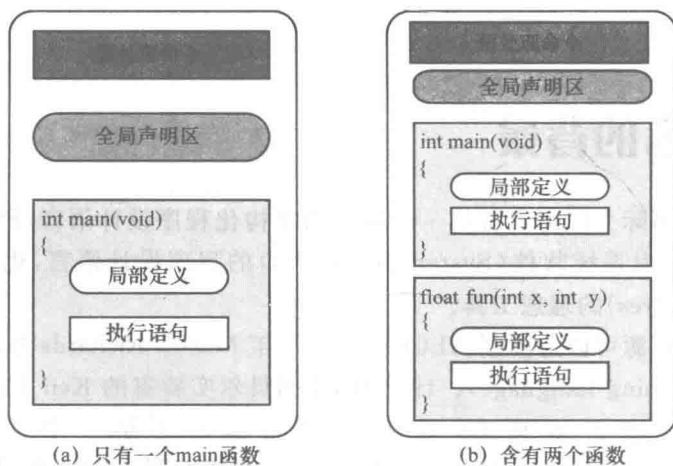


图 1-2 典型 C 程序的结构

说明以下几点:

(1) 关于全局区。全局区也叫全局声明区,它是位于程序开头部分的区域,主要用来进行全局对象的声明。该区域中声明的对象所有程序代码都可以访问。

(2) 关于函数。

① 函数是具有独立功能的命名的代码段(Code Segment)。

② 函数是C程序的基本结构单位(Unit)。

③ 函数包括函数头(Function Header)和函数体(Function Body)两部分。

函数头包含了函数值类型、函数名和参数列表。函数体跟在函数头后,是用{ }括起来的部分,包含了局部定义区(Local Definition Section)和可执行语句(Statement)两部分,局部定义在函数体的开始部分,主要用于定义函数内部要用到的变量(Variable)。语句部分在局部定义的后面,它是由若干条语句组成的。图1-3显示了函数的结构情况。

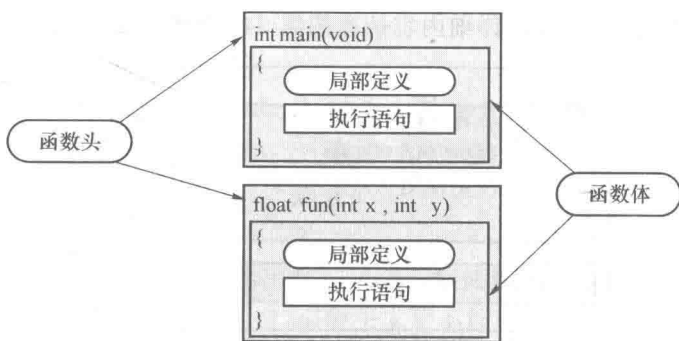


图 1-3 函数的结构

④ main 函数必须要有,它是整个程序的控制部分。

每个程序中,有且必须有一个称作 main 的函数。程序的执行是由 main 函数的开始而开始,由它的结束而结束。其他函数都是在开始执行 main 函数以后,通过函数调用(Function Call)而得以执行的。

⑤ 函数包括库函数(Library Function)和用户定义函数(User-Defined Function)。

库函数是由开发商提供的函数,可以直接使用;用户定义函数是用户根据需要自己编制的函数。

⑥ 函数间只存在调用(Calling)和被调用(Called)的关系。

main 函数只能由操作系统调用,它可以调用任何函数,其他函数之间可以相互调用。

(3) 关于语句(Statements)。

① 语句是 C 程序的最小功能单位。

一条语句经编译(Compile)转化为可执行的机器指令(Instruction)。

② C 语言中的语句以分号(Semicolon)结束。

③ C 语言中的语句包括表达式语句、空语句、复合语句和控制语句,详细内容参见第 59 页 2.5 节。

④ C 语言中允许一行写多条语句,也允许一条语句占多行,本书特别提倡一行一句的书写规范。

(4) 关于注释(Comments)。

① 注释是为了增强程序的可读性而在程序代码中加入的说明性信息,格式是:

/* 注释的内容 */

② 注释以 /* 开始,由 */ 结束。

③ 可以在除一条可执行语句内部之外的任何地方加注释。

④ 注释可以跨多行。图 1-4 给出了几个注释的范例。

⑤ 注释部分在编译时被忽略。

根据这一特点,在调试程序时,若不想让某部分代码参加编译和运行,就可以把这部分代码的首尾用 /* 和 */ 括起来,该部分代码就暂时被屏蔽了。

(5) 关于预编译命令(Precompiler Directives)。

① 预编译命令是由系统提供的命令格式。

之所以称作预编译命令,是因为这些命令在编译前实施相应处理。预编译命令包括文

件包含、宏定义和条件编译三种,详细内容请参阅第 347 页附录 E。

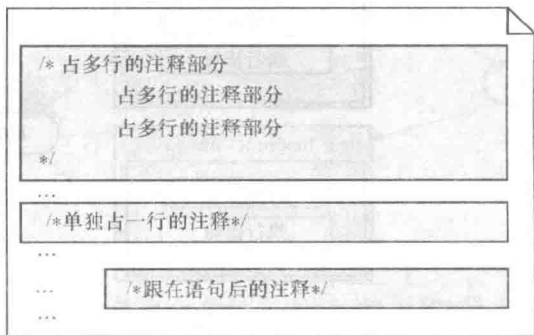


图 1-4 注释范例

② 所有的预编译命令都是以 # 开头,末尾没有分号。

如:

```
#include <stdio.h>           /* 文件包含命令 */
#define PI 3.14159           /* 宏定义命令 */
```

③ 预编译命令一般都放在程序最开始的位置。

运行效率、可读性和可维护性是衡量一个程序好坏的重要指标。一个好的程序就是力争做到有很高的运行效率,方便别人阅读、日后修改和升级。良好的书写格式、适当地添加注释都是增强程序可读性的有利手段。

1.2.2 两个程序例子

下面通过两个具体的程序来研究 C 程序的结构问题。

图 1-5 给出了一个非常简单的程序。程序中只有一个 main 函数,只包含了一条预编译命令,没有全局声明部分,且函数体中没有局部定义,只包含两条语句。功能是在屏幕上输出“Hello World!”。

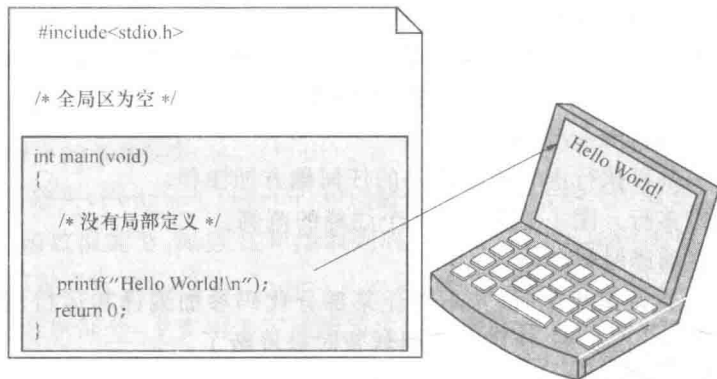


图 1-5 一个简单的 C 程序

程序开头是一条预包含命令：

```
#include <stdio.h>
```

它的作用是把尖括号中名字为 `stdio.h` 的头文件包含到程序中，程序编译时系统将用 `stdio.h` 的内容替换这条语句。在 `stdio.h` 中包含了库函数 `printf()` 的相关信息。

在预包含命令后面是 `main` 函数的函数头：

```
int main( void )
```

其中，`int` 是函数的返回值类型，表示整型；`main` 是函数名；圆括号是函数的标识，括号中的 `void` 表示该函数是无参数的。注意：函数头末尾不能有分号。

在函数头后面用 `{ }` 括起来的部分，是 `main` 函数的函数体，里面有两条语句，一条用来输出信息，一条用来结束程序。第一条语句“`printf( ... );`”是调用库函数 `printf` 来输出信息，由函数名 `printf` 和括在一对圆括号中的参数组成。本例中的参数只有一个，是用双引号括起来的部分，是要输出的内容。“`\n`”是换行符，控制在输出“Hello World!”后换行。`printf()` 后的分号是语句结束符，C 语言中的每一个语句都以“`;`”终止。第二条语句“`return 0;`”用来结束(Terminate)`main` 函数执行，把控制还给操作系统(Return the control to OS)。

图 1-6 给出了一个较复杂的 C 程序。程序包含两个函数——`main` 和 `average`。`main` 函数调用 `average` 函数，因此 `main` 函数称作调用函数(Calling Function)，`average` 函数称作被调用函数(Called Function)。

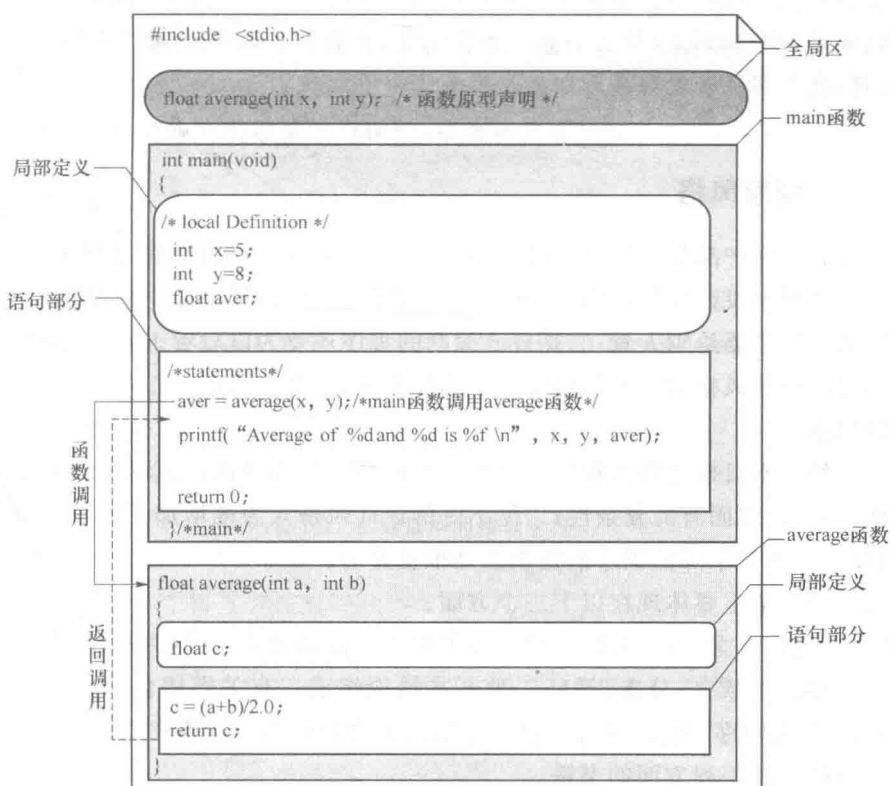


图 1-6 一个较复杂的 C 程序

average 函数的作用是接收两个整数,返回一个浮点型的平均值。在函数头中, float 用来指定返回值的类型, a 和 b 称作形式参数 (Formal Parameters), 用来接收调用函数传递来的数据, int 用来指定接收数据的类型。函数体中共有三条语句, 第一条语句是局部变量定义语句, 后面两条是执行语句。第一条语句的作用是申请了一个名字为 c 的用来存储 float 型数据的变量 (Variable)。第二条语句的作用是求 a 和 b 的平均值, 并把结果存储到 c 中。第三条语句的作用是把 c 的值传递给调用函数。

由于 main 函数要调用库函数 printf 输出数据, 因此在程序开头同样有一条预包含命令。因为 main 函数要调用用户自定义函数 average, 因此在全局区写了一条函数原型声明语句, 以便调用的顺利进行, 具体细节参阅第 224 页 8.3 节。

main 函数的函数体包含了六条语句, 前面的三条是局部变量定义语句, 分别用来定义整型的变量 x、y 和浮点型变量 aver, 并给 x 和 y 分别赋了初始值 5 和 8。第四条语句是函数调用语句, 把 x 和 y 的值传给函数 average, 并把 average 函数的返回结果存储到变量 aver。在函数调用中, x 和 y 称作实参 (Actual Parameters)。第五条语句也是函数调用语句, 是调用库函数 printf 把 x、y 和 aver 中的数据输出到屏幕上。



特别说明一点:

教材中的很多例子, 作者在认为需要的地方都适当加了注释, 用它来告诉大家一个信息、提示一个知识点、指出一个注意事项等, 主要目的是便于大家更好地理解和掌握知识, 这与实际开发程序时加的注释是完全不同的。

1.2.3 C 程序风格

程序设计的最终产品是程序, 但仅设计和编制出一个运行结果正确的程序是不够的, 还应该养成良好的程序设计风格。作为一个称职的程序员, 应该记住两个最基本的出发点: 一是自己编制的程序还要给别人看; 二是自己编制的程序还要为以后看。

良好的程序设计风格需要在编程实践中逐步养成, 主要包括:

1. 设计风格

程序设计的根本目标是降低程序的复杂性和提高程序的可读性。程序的复杂性主要来自两个方面: 一是问题固有的复杂性; 二是不好的设计风格人为地增加了程序的复杂性。因此, 良好的设计风格对于降低程序的复杂性是很重要的。

程序设计的风格主要体现在以下三个方面。

(1) 结构要清晰。为了达到这个目标, 要求程序是模块化结构的, 并且是按层次组织的, 每个模块内部都是由顺序、分支、循环三种基本结构组成。有关模块化的内容可以参阅第 220 页 8.1 节; 有关顺序、分支、循环三种基本结构的内容可以参阅第 82 页 3.1 节, 更多的知识大家可以查阅软件工程方面的书籍。

(2) 思路要清晰。要求程序是模块化结构的, 在设计的过程中要遵循自顶向下、逐步细化的原则, 详细内容可参阅第 82 页 3.1 节。