

面向 21 世纪高等职业技术教育计算机类规划教材

C 语言程序设计案例教程

(第二版)

主 编 李培金

副主编 雷育春 石 锋

参 编 贾步忠 宋 翔 薛 静

李艳华 李 晗 张 浩

陕西省精品课程配套教材

西安电子科技大学出版社

2008

内 容 简 介

本书是陕西省省级精品课程的配套教材。全书主要包括：C 语言程序结构及运行环境、各种数据类型及使用方法、C 语言模块化程序设计方法、顺序结构、选择结构、循环结构、数组、函数、编程预处理、指针、结构体、位运算、文件及综合应用等。

本书的特点是在内容安排上采用循序渐进的方式，在组织形式上采用通俗易懂的案例教学和启发式教学的方式，并辅以大量便于说明问题的案例。本书每章后附有相当数量的习题和实验题，供读者练习。

本书可作为高职高专院校信息、电子类等专业的教材，也可供对 C 语言感兴趣的其他读者自学使用。

★ 本书配有电子教案，需要者可与出版社联系，免费提供。

图书在版编目(CIP)数据

C 语言程序设计案例教程/李培金主编. —2 版. —西安：西安电子科技大学出版社，2008.1

面向 21 世纪高等职业技术教育计算机类规划教材

ISBN 978-7-5606-1270-6

I. C… II. 李… III. C 语言—程序设计—高等学校：技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 182913 号

策 划 马晓娟

责任编辑 马晓娟

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2008 年 1 月第 2 版 2008 年 1 月第 5 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 20.5

字 数 478 千字

印 数 20 001~24 000 册

定 价 27.00 元

ISBN 978-7-5606-1270-6/TP·0668

XDUP 1541022-5

如有印装问题可调换

本社图书封面为激光防伪覆膜，谨防盗版。

前 言

自 2003 年此书第一版出版以来,受到了广大读者的普遍好评,同时,不少读者也给本书提出了一些改进建议,编者在此表示衷心的感谢。

本书主编致力于 C 语言程序设计教学工作近 20 年,是“C 语言程序设计”省级精品课程建设项目负责人,同时,本书也是省级精品课程建设的配套教材。为了不辜负专家和广大读者对本书的厚爱,本着对读者认真负责的态度,本书编者与西安电子科技大学出版社进行了充分的研究与准备,在保证内容体系完整的基础上,于 2007 年对第一版的《C 语言程序设计案例教程》进行了全面修订。修订内容主要体现在以下几个方面:

1. 在内容的组织形式上,调整了部分章节的先后次序,使内容的衔接更加合理、自然,结构更加清晰,有利于老师教学与学生自学。

2. 在内容叙述上,增加了更多的示例,使概念、语法及应用均与案例教学相结合,这样,学生对内容的理解与掌握将更加容易。

3. 为了提高学生的 C 语言程序设计应用能力,除了扩充每章的案例以外,还对课后习题进行了部分的更换与增加,同时,还新增加了上机实验题,旨在通过对程序调试手段的学习和真正动手上机实践,使学生对 C 语言的认识迈上一个新的台阶。

全书内容共分 13 章,主要介绍了 C 语言程序设计的基本概念、程序结构及语法规则、C 语言的运行环境、各种数据类型及使用方法、C 语言的运算符与表达式、模块化程序设计方法、顺序结构、选择结构、循环结构、数组、函数、编译预处理、指针、结构体、共用体、位运算、文件和综合应用等内容。

本书由李培金老师担任主编,雷育春、石锋老师担任副主编。其中,李培金老师修订了第 1、2、3、10、12 章,并负责完成全书的修改与定稿;雷育春老师修订了第 8、9、11、13 章;宋翔老师修订了第 4 章;薛静老师修订了第 5 章;贾步忠老师修订了第 6、7 章。

本书是高等职业学院、高等技术学院、高等专科学校的教材,适用于信息技术及电子类的相关专业,如计算机应用、计算机网络、信息管理、电子商务、计算机科学技术、会计电算化、电器自动化等。同时,本书也可供优秀职高学校选作教材。对于那些要提高自己的应用技能或参加某些计算机证书考试的读者,本书也不失为一本好的参考书。

最后,恳请广大读者将本书的使用情况及各种意见、建议及时反馈给我们,以便我们在今后的工作中不断改进和完善。

编 者

2007 年 10 月

第一版前言

C 语言是一种流行的程序设计语言，是计算机专业学生的一门必修课，也是全国计算机等级考试及 NIT 考试中比较重要的科目之一。本书是根据教育部关于高职高专培养目标及对 C 语言课程的基本要求，并结合我们多年针对高职高专学生的 C 语言教学实践经验编写的。

全书内容共分 13 章，主要介绍了 C 语言程序设计的基本概念、程序结构及语法规则、运行环境、各种数据类型及使用方法、运算符与表达式、顺序结构、选择结构、循环结构、数组、函数、编译预处理、指针、结构体、共用体、位运算、文件和综合应用等内容。

本书有以下主要特点：

(1) 从实用的角度阐述 C 语言的基本概念、语法规则及基本操作方法。

(2) 在内容的组织形式上，采用案例教学和启发式教学的方法。

(3) 内容叙述深入浅出，通俗易懂，且全书结构清晰，层次分明，示例丰富。

(4) 良好的源程序书写风格、程序设计风格和算法优化思想贯穿于全书。

(5) 以案例为主线，通过任务驱动的学习方式，既能使学生掌握相应的知识及调试程序的基本技能，又能培养学生应用知识的能力和编程的能力。

本书由李培金老师担任主编，石锋、雷育春老师担任副主编，丁春莉老师担任主审。其中，李培金老师编写了第 1、2、12 章和附录，并负责全书的修改与定稿；石锋老师编写了第 3、10 章；雷育春老师编写了第 8、9 章；李艳华老师编写了第 6、13 章；贾步忠老师编写了第 7 章；宋翔老师编写了第 4 章；李晗老师编写了第 5 章；张浩老师编写了第 11 章。

本书可作为高职高专院校信息、电子类等专业的教材，也可供对 C 语言感兴趣的其他读者自学使用。

在本书出版之际，主编代表全体作者，感谢西安电子科技大学出版社的编辑对本书编写风格所提出的宝贵建议，以及对本书的出版所做的一切工作。

由于作者水平有限，书中难免有不当之处，恳请专家和广大读者批评指正。

李培金
2003 年 9 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言的发展与特点	1
1.2 C 语言程序的结构和语法规则	2
1.2.1 C 语言程序的基本结构	2
1.2.2 C 语言函数的一般结构	4
1.2.3 C 语言函数的语法规则	5
1.3 C 语言的编译环境	6
1.4 本章小结	11
习题	11
实验题	12
第 2 章 数据类型、运算符与表达式	14
2.1 C 语言的数据类型	14
2.2 标识符、常量和变量	14
2.2.1 标识符	14
2.2.2 常量	16
2.2.3 变量	16
2.3 整型数据	17
2.3.1 整型常量	17
2.3.2 整型变量	17
2.4 实型数据	18
2.4.1 实型常量	18
2.4.2 实型变量	19
2.5 字符型数据	19
2.5.1 字符常量	19
2.5.2 字符变量	20
2.5.3 字符串常量	22
2.6 算术运算与算术表达式	22
2.6.1 算术运算	23
2.6.2 算术表达式	23
2.7 赋值运算与赋值表达式	24
2.8 自增、自减和逗号运算	25
2.8.1 自增(++)、自减(--)运算	25
2.8.2 逗号运算符及其表达式	26

2.9 本章小结	27
习题	29
实验题	31
第 3 章 顺序结构程序设计	33
3.1 程序设计概述	33
3.1.1 算法的概念	33
3.1.2 算法的特征	33
3.1.3 算法的描述	34
3.1.4 结构化程序设计方法	35
3.2 C 语言语句	36
3.3 格式化输出函数 printf()	37
3.3.1 printf() 函数的一般格式	38
3.3.2 格式指示符	39
3.3.3 使用说明	41
3.4 格式化输入函数 scanf()	41
3.4.1 scanf() 函数的一般格式	41
3.4.2 格式指示符	41
3.4.3 使用说明	42
3.5 单个字符的输入/输出函数	43
3.6 顺序程序设计举例	45
3.7 本章小结	47
习题	48
实验题	51
第 4 章 选择结构程序设计	53
4.1 关系运算及其表达式	53
4.1.1 关系运算符及其优先次序	53
4.1.2 关系表达式	54
4.2 逻辑运算及其表达式	54
4.2.1 逻辑运算符及其优先次序	54
4.2.2 逻辑表达式	55
4.3 if 语句和条件运算符	56
4.3.1 if 语句	56
4.3.2 条件运算符	62
4.4 switch 语句	63
4.5 选择结构程序设计举例	67
4.6 本章小结	70
习题	71
实验题	74

第 5 章 循环结构程序设计	77
5.1 循环语句概述	77
5.2 while 语句和 for 语句	77
5.2.1 while 语句	77
5.2.2 for 语句	78
5.3 do-while 语句	80
5.4 循环结构的嵌套	81
5.5 break 语句和 continue 语句	83
5.6 goto 语句	84
5.7 循环结构程序设计应用	85
5.7.1 应用举例	85
5.7.2 穷举与迭代算法	86
5.8 本章小结	90
习题	91
实验题	95
第 6 章 数组	98
6.1 数组的概念	98
6.2 一维数组	98
6.2.1 一维数组的定义	98
6.2.2 一维数组元素的引用	100
6.2.3 一维数组的初始化	101
6.2.4 一维数组应用举例	101
6.3 二维数组	104
6.3.1 二维数组的定义	104
6.3.2 二维数组元素的引用	105
6.3.3 二维数组的初始化	106
6.3.4 二维数组应用举例	106
6.4 字符数组与字符串	109
6.4.1 字符数组的定义和基本操作	109
6.4.2 字符数组的整体操作	111
6.4.3 常用的字符串处理函数	112
6.4.4 字符数组应用举例	114
6.5 本章小结	116
习题	117
实验题	120
第 7 章 函数	122
7.1 函数的定义与调用	122
7.1.1 函数的定义	122
7.1.2 函数的返回值与函数类型	125

7.1.3	对被调用函数的声明和函数原型	127
7.1.4	函数的调用	129
7.1.5	函数的参数传递	130
7.2	函数的嵌套调用和递归调用	132
7.2.1	函数的嵌套调用	132
7.2.2	函数的递归调用	134
7.3	数组作为函数参数	137
7.3.1	数组元素作为函数实参	137
7.3.2	数组名作为函数的形参和实参	138
7.4	内部变量与外部变量	141
7.4.1	内部变量	141
7.4.2	外部变量	142
7.5	内部函数和外部函数	145
7.5.1	内部函数	145
7.5.2	外部函数	145
*7.5.3	多个源程序文件的编译和连接	148
7.6	变量的动态存储与静态存储简介	149
7.6.1	动态存储与静态存储的概念	149
7.6.2	内部变量的存储方式	149
7.6.3	外部变量的存储方式	152
7.7	本章小结	153
	习题	155
	实验题	160
第 8 章	编译预处理	163
8.1	宏定义	163
8.1.1	无参宏定义	163
8.1.2	带参宏定义	167
8.2	文件包含	172
*8.3	条件编译	173
8.3.1	#ifdef、#else 和#endif 条件编译指令组合	173
8.3.2	#ifndef、#else 和#endif 条件编译指令组合	174
8.3.3	#if、#else 和#endif 条件编译指令组合	174
8.4	本章小结	175
	习题	176
	实验题	177
第 9 章	指针	179
9.1	指针和指针变量的概念	179
9.1.1	指针的基本概念	179
9.1.2	指针变量的基本概念	179

9.2 指针变量的定义与应用.....	180
9.2.1 指针变量的定义与相关运算.....	180
9.2.2 指针变量作为函数参数.....	185
9.3 指向数组的指针变量.....	187
9.3.1 概述.....	187
9.3.2 通过指针引用数组元素.....	188
9.3.3 指向数组的指针变量作为函数参数.....	190
9.3.4 二维数组的指针及其指针变量.....	193
9.4 字符串的指针和指向字符串的指针变量.....	196
9.4.1 字符串的表示和引用.....	196
9.4.2 字符串指针作函数参数.....	200
9.5 返回指针的函数.....	201
9.6 指针数组与主函数 main()的形参.....	203
9.6.1 指针数组.....	203
9.6.2 指针数组作为函数参数的使用.....	204
9.6.3 主函数 main()的形参.....	205
9.6.4 指向指针的指针变量.....	207
9.7 函数的指针和指向函数的指针变量.....	209
9.8 本章小结.....	211
习题.....	212
实验题.....	216
第 10 章 结构体、共用体和枚举类型.....	218
10.1 结构体类型与结构体变量的定义.....	218
10.1.1 结构体类型的定义.....	218
10.1.2 结构体变量的定义.....	219
10.2 结构体变量的引用与初始化.....	220
10.2.1 结构体变量的引用.....	220
10.2.2 结构体变量的初始化.....	221
10.3 结构体数组.....	221
10.3.1 结构体数组的定义.....	221
10.3.2 结构体数组的初始化.....	222
10.4 指向结构体类型数据的指针.....	223
10.4.1 指向结构体变量的指针.....	223
10.4.2 指向结构体数组的指针.....	224
10.4.3 指向结构体数据的指针作函数参数.....	224
10.5 单链表.....	225
10.5.1 链表概述.....	225
10.5.2 建立单链表.....	227
10.5.3 在单链表中插入结点.....	228

10.6 共用体类型和枚举类型.....	228
10.6.1 共用体类型.....	228
10.6.2 枚举类型.....	230
10.7 定义已有类型的别名.....	231
10.8 本章小结.....	231
习题.....	232
实验题.....	239
第 11 章 位运算.....	242
11.1 数值在计算机中的表示.....	242
11.1.1 二进制数的简单运算.....	242
11.1.2 数值在计算机中的表示.....	242
11.2 位运算.....	243
11.2.1 位运算符.....	243
11.2.2 位运算符的功能实现.....	244
*11.3 位段简介.....	247
11.3.1 位段的定义和位段变量的说明.....	247
11.3.2 位段的使用.....	248
11.4 本章小结.....	249
习题.....	249
实验题.....	250
第 12 章 文件.....	251
12.1 文件概述.....	251
12.2 文件的打开与关闭.....	252
12.2.1 文件的打开(fopen()函数).....	253
12.2.2 文件的关闭(fclose()函数).....	254
12.3 文件的读/写操作.....	254
12.3.1 读/写文件中的一个字符——fgetc()函数和 fputc()函数.....	254
12.3.2 读/写一个字符串——fgets()函数和 fputs()函数.....	257
12.3.3 读/写一个数据块——fread()函数和 fwrite()函数.....	259
12.3.4 对文件进行格式化读/写——fscanf()函数和 fprintf()函数.....	259
12.3.5 读/写函数的选用原则.....	260
12.4 位置指针与文件定位.....	260
12.4.1 位置指针复位函数 rewind().....	260
12.4.2 随机读/写与 fseek()函数.....	260
12.4.3 返回文件当前位置的函数 ftell().....	261
12.5 本章小结.....	261
习题.....	262
实验题.....	263

第 13 章 C 语言综合应用	265
13.1 C 语言的菜单、图形与窗口设计功能	265
13.1.1 文本窗口	265
13.1.2 图形操作	273
13.2 C 语言程序设计综合应用	287
13.2.1 利用 C 语言调用汇编语言	287
13.2.2 C 语言综合应用实例	288
13.3 本章小结	298
习题	298
实验题	298
附录	300
附录 A 标准 ASCII 码字符编码表	300
附录 B 运算符的优先级和结合方向	301
附录 C 标准库函数	302
附录 D Turbo C V2.0 常见错误	307
参考文献	314

第 1 章 C 语言概述

C 语言是一种面向过程的，并且很灵活的程序设计语言，它有许多程序设计技巧。C 语言的语法源自 ANSI C 标准(ANSI: 美国国家标准化协会)。掌握了 C 语言后，再学 C++、VC++、Java 等其他语言就比较容易了。

本章主要介绍 C 语言程序的结构、书写规则以及编译环境 Turbo C 的基本操作。

1.1 C 语言的发展与特点

1. C 语言的发展

在 C 语言诞生以前，系统软件(例如操作系统)主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差，但一般的高级语言又难以实现对计算机硬件的直接操作(这正是汇编语言的优势)，因此人们盼望有一种兼有汇编语言和高级语言特性的新语言。C 语言就是在这种背景下应运而生的。

C 语言是贝尔实验室于 20 世纪 70 年代初期研制出来的，并随着 UNIX 操作系统的日益广泛使用，迅速得到推广。后来，C 语言又被多次改进，并出现了多种版本。20 世纪 80 年代初(1983 年)，美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本对 C 语言进行发展和扩充，制定了 ANSI C 标准(1989 年再次做了修订)。本书以 ANSI C 标准为基础来介绍 C 语言。

目前，在微机上广泛使用的 C 语言编译系统有 Microsoft C(简称 MSC)、Turbo C(简称 TC)、Borland C(简称 BC)等。虽然它们的基本部分都是相同的，但还是有一些差异的，所以请读者注意自己所使用的 C 编译系统的特点和规定(可参阅相应的手册)。本书选定的上机环境是 Turbo C V2.0。

下面是一个计算圆面积的 C 语言程序段：

```
main()                /*告诉编译器 C 程序由此开始执行*/
{
    float r,s;         /*定义圆半径 r 与面积 s 为实型数据*/
    r=5.356;           /*给半径 r 赋值*/
    s=3.14159*r*r;     /*计算面积 s*/
    printf("%f\n",s);  /*输出面积 s 的值*/
}                      /*程序执行结束*/
```

其中，/*与*/之间的内容称为注释内容。

由上述 C 语言程序可以看出，C 语言是一种面向过程的程序设计语言。

2. C 语言的特点

C 语言是近年来较流行的高级程序设计语言之一，许多大型软件均是用 C 语言编写的(如 UNIX 操作系统)。C 语言同时具有汇编语言和高级语言的双重特性。具体来说，C 语言的主要特点如下：

(1) C 语言是一种模块化的程序设计语言。模块化的基本思想是将一个大的程序按功能分割成一些模块，使每一个模块都成为功能单一、结构清晰、容易理解的小程序。

(2) C 语言简洁，结构紧凑，使用方便、灵活。C 语言总共只有 32 个关键字，9 条控制语句，源程序书写格式自由。

(3) C 语言的运算功能极其丰富，数据处理能力强。C 语言总共有 34 种运算符，如算术运算符，关系运算符，自增(++)、自减(--)运算符，复合赋值运算符，位运算符及条件运算符等。同时，C 语言还可以实现其他高级语言较难实现的功能。

(4) C 语言的可移植性好。C 语言程序基本上可以不作任何修改，就能运行于各种不同型号的计算机和各种操作系统环境上。

(5) C 语言可以直接调用系统功能，实现对硬件的操作。这是其他高级语言所不具备的。

当然，C 语言本身也有缺点，如 C 语言的语法限制不太严格，在增加程序设计的灵活性的同时，在一定程度上降低了某些安全性，这就对程序设计人员提出了更高的要求。

要学好任何一门计算机语言都不是一件很容易的事，学习 C 语言也不例外，但掌握了 C 语言后，再学其他语言就比较容易了，所以对有志于从事计算机编程的人而言，C 语言是一门要认真加以钻研的计算机语言。

1.2 C 语言程序的结构和语法规则

1.2.1 C 语言程序的基本结构

下面通过几个简单的示例，介绍 C 语言程序的基本构成和书写格式，使读者对 C 语言程序有一个基本的了解。在此基础上，再进一步了解 C 语言程序的语法和书写规则。

【例 1.1】 求三个数的平均值的 C 语言程序。

```
/*功能：求三个数的平均值*/
main()                               /* main()称为主函数*/
{float a,b,c,ave;                     /*定义 a, b, c, ave 为实型数据*/
  a=7;
  b=9;
  c=12;
  ave=(a+b+c)/3;                      /*计算平均值*/
  printf( " ave=%f\n ",ave);          /*在屏幕上输出 ave 的值 */
}
```

程序运行结果：

```
ave=9.333333
```

【例 1.2】输出两个数中的较大值的 C 语言程序。

```
/*功能：输出两个数中的较大值 */
main()                                /*主函数*/
{int num1,num2;                       /*定义 num1、 num2 为整型变量*/
  scanf( " %d,%d " ,&num1,&num2);    /*由键盘输入 num1、 num2 的值*/
  printf( " max=%d\n " ,max(num1,num2)); /*在屏幕上输出调用 max 函数后的函数值*/
}
/*以下是用户设计的函数 max()*/
int max(int x,int y)                  /*x 和 y 分别取 num1 和 num2 传递的值*/
{if(x>y) return x;                   /*如果 x>y, 则将 x 的值返回给 max */
 else return y;                      /*如果 x>y 不成立, 则将 y 的值返回给 max */
}
```

程序运行情况：

5,8↵ (“↵”表示按回车键，以下相同)

max=8

在以上两个示例中，【例 1.1】所示的 C 语言程序仅由一个 main()函数构成，它相当于其他高级语言中的主程序；【例 1.2】所示的 C 语言程序由一个 main()函数和一个其他函数 max()(自己设计的函数)构成，函数 max()相当于其他高级语言中的子程序。由此可见，一个完整的 C 语言程序结构有以下两种表现形式：

(1) 仅由一个 main()函数(又称主函数)构成，结构如图 1-1(a)所示。

(2) 由一个且只能有一个 main()函数和若干个其他自定义函数结合而成，结构如图 1-1(b)所示。其中，自定义函数由用户自己设计。

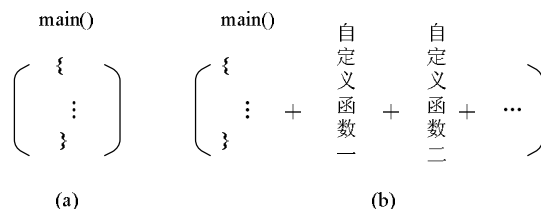


图 1-1 C 语言程序结构示意图

结合以上示例可以看出，C 语言程序结构有以下基本特点：

(1) C 语言程序是由函数(如：main()函数和 max()函数)组成的，每一个函数完成相对独立的功能，函数是 C 语言程序的基本模块单元。main 是函数名，函数名后面的一对圆括号“()”是用来写函数的参数的，参数可以有，也可以没有(本程序没有参数)，但圆括号不能省略。

(2) 一个 C 语言程序总是从 main()函数开始执行的。主函数执行完毕，程序就执行完毕。

(3) C 语言编译系统区分字母大小写。C 语言把大小写字母视为两个不同的字符，并规定每条语句或数据说明均以分号(;)结束。分号是语句不可缺少的组成部分。

(4) 主函数 main()既可以放在 max()函数之前，也可以放在 max()函数之后。习惯上，

将主函数 `main()` 放在最前面。

(5) C 语言程序中所调用的函数，既可以是由系统提供的库函数，也可以是由设计人员自己根据需要而设计的函数。例如，在【例 1.2】中，`printf()` 函数是 C 语言编译系统库函数中的一个函数，它的作用是在屏幕上按指定格式输出指定的内容；`max()` 函数是由用户自己设计的函数，它的作用是计算两个数中的较大值。

1.2.2 C 语言函数的一般结构

任何函数(包括主函数 `main()`)都是由函数说明和函数体两部分组成的。其一般结构如下：

```
[函数类型] 函数名([函数形式参数表]) /* 函数说明部分 */
{
    [数据说明部分]
    函数执行部分
} /* 函数体部分*/
```

其中，加方括号“[]”时，表示其中的内容可以省略，以下相同。

1. 函数说明部分

函数说明部分由函数类型(可缺省)、函数名和函数形式参数表(简称形参表)三部分组成。其中，函数形参表的一般格式为

([数据类型 参数 1,][数据类型 参数 2,]...)

例如，【例 1.2】中的函数 `max()`，其函数说明部分如图 1-2 所示。

```
int      max      (int x, int y)
  ↓      ↓      ↓
函数类型 函数名   函数形参表
```

图 1-2 函数说明部分结构示意图

注意：如果函数不需要参数，则可用 `(void)` 或 `()` 表示，但主函数 `main()` 例外。例如，【例 1.1】和【例 1.2】中的主函数 `main()`。

2. 函数体部分

函数体部分由函数说明部分以下的一对大括号“{}”内的若干条语句构成。函数体一般又由数据说明部分和函数执行部分两部分构成。如果一个函数内有多对大括号，则最外面的一对大括号是函数体的范围。

1) 数据说明部分

数据说明部分由变量定义、自定义函数声明、外部变量说明等部分组成，其中变量定义是主要的。例如，【例 1.2】中 `main()` 函数体里的“`int num1, num2;`”语句定义了两个整型变量 `num1` 和 `num2`。

2) 函数执行部分

函数执行部分一般由若干条可执行语句构成。例如，在【例 1.2】的 `main()` 函数体中，除变量定义语句“`int num1, num2;`”外，其余 5 条语句构成该函数的可执行语句部分。

有关函数的详细内容，将在后续章节介绍。

1.2.3 C 语言函数的语法规则

C 语言函数的语法规则一般可归纳为以下四条：

(1) 函数体中的数据说明语句，必须位于可执行语句之前。换句话说，数据说明语句不能与可执行语句交织在一起。例如，下面程序中变量定义语句“int max;”的位置是非法的：

```
main()
{ int x,y;          /*定义两个整型变量 x 和 y*/
  x=2;             /*将 2 赋值给变量 x*/
  y=9;             /*将 9 赋值给变量 y*/
  int max;         /*变量定义出现在赋值语句“x=2;”和“y=9;”之后，属非法定义！*/
  if(x>y) max=x;
  else max=y;      /*如果 x>y 成立，则将 x 赋值给 max; 否则，将 y 赋值给 max*/
  printf(" max=%d\n",max); /*在屏幕上输出 max 的值*/
}
```

如何改正请读者思考。

(2) 如果不需要，也可以缺省数据说明语句。例如，下面的程序缺省了说明语句。

```
main()
{
  printf(" Happy new year! ");
}
```

程序运行结果：

Happy new year!

(3) 程序行的书写格式自由，既允许一行内写多条语句，也允许一条语句分写在多行上，但所有语句都必须以分号结束。如果某条语句很长，一般需要将其分写在多行上。例如，【例 1.1】的主函数 main()也可改写成如下所示的格式：

```
main()
{float a,b,c,ave;          /*定义 a, b, c, ave 为实型变量*/
  a=7;b=9;c=12;          /*将第 2、第 3、第 4 三行合并成一行*/
  ave=(a+b+c)/3;        /*计算平均值*/
  printf(" a=%f,b=%f,c=%f,ave=%f\n",
        a,b,c,ave);     /*一条语句可分两行书写*/
}
```

(4) 允许使用注释。一个高质量的程序，在其源程序中应加上必要的注释，以增强程序的易读性。C 语言的注释格式为：/*注释内容*/。

例如，在【例 1.1】和【例 1.2】中，以及本节其他部分给出的源程序中，凡用“/*”和“*/”括起来的文字，都是注释内容。

= “/*”和“*/”必须成对使用，且“/”和“*”以及“*”和“/”之间不能有空格，否则就出错。

- = 注释内容可以单独占一行，也可以跟在语句的后面。
- = 如果一行写不下，可另起一行继续写。
- = 注释中允许使用汉字。在非中文操作系统下，看到的是一串乱码，但不影响程序运行。

1.3 C 语言的编译环境

Turbo C(TC)是一个集源程序编辑、编译、连接、运行与调试于一体，用菜单驱动的综合软件环境，具有简单、方便、速度快、高效、功能强等特点。C 语言程序员可在 Turbo C 环境下完成所有工作。

1. 运行一个 C 语言程序的一般过程

运行一个 C 语言程序的一般过程如图 1-3 所示。

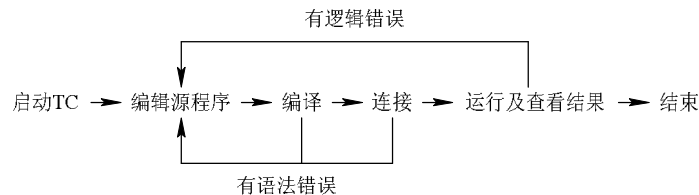


图 1-3 运行一个 C 语言程序的一般过程

- (1) 启动 TC，进入 TC 集成环境。
- (2) 编辑。如果源程序存在语法错误，则修改源程序中的错误。
- (3) 编译。如果编译成功，则可进行下一步操作，否则返回(2)修改源程序，再重新编译，直至编译成功。
- (4) 连接。如果连接成功，则可进行下一步操作，否则，根据系统的下一步提示，进行相应修改，再重新连接，直至连接成功。
- (5) 运行及查看结果。通过观察程序运行结果，验证程序的正确性。如果出现逻辑错误，则必须返回(2)修改源程序，再重新编译、连接和运行，直至程序正确。
- (6) 运行结果若正确，便可退出 TC 集成环境，结束本次程序运行。

2. TC 的启动、退出与命令菜单

1) 启动 Turbo C

输入 tc 后即可启动 Turbo C。启动 Turbo C 后其主菜单栏横向排列在屏幕顶端，并被激活，如图 1-4 所示。

主菜单的下面，依次是 Edit(编辑)窗口、Watch(消息)窗口和 7 个功能键。两个窗口中，顶端横线为双线显示的，表示该窗口是活动窗口。在图 1-4 中，编辑窗口是活动窗口。

编辑窗口的顶端为状态行，其中：

- = Line 1 和 Col 1：显示光标所在的行号和列号，即光标位置。
- = Insert：表示编辑状态处于“插入”状态。当处于“改写”状态时，此处为空白。用“Insert”键可实现“插入”与“改写”状态之间的转换。