



普通高等教育“十二五”应用型本科规划教材

# C语言程序设计案例教程

主编 段新娥 副主编 李 荣



西安交通大学出版社  
XI'AN JIAOTONG UNIVERSITY PRESS



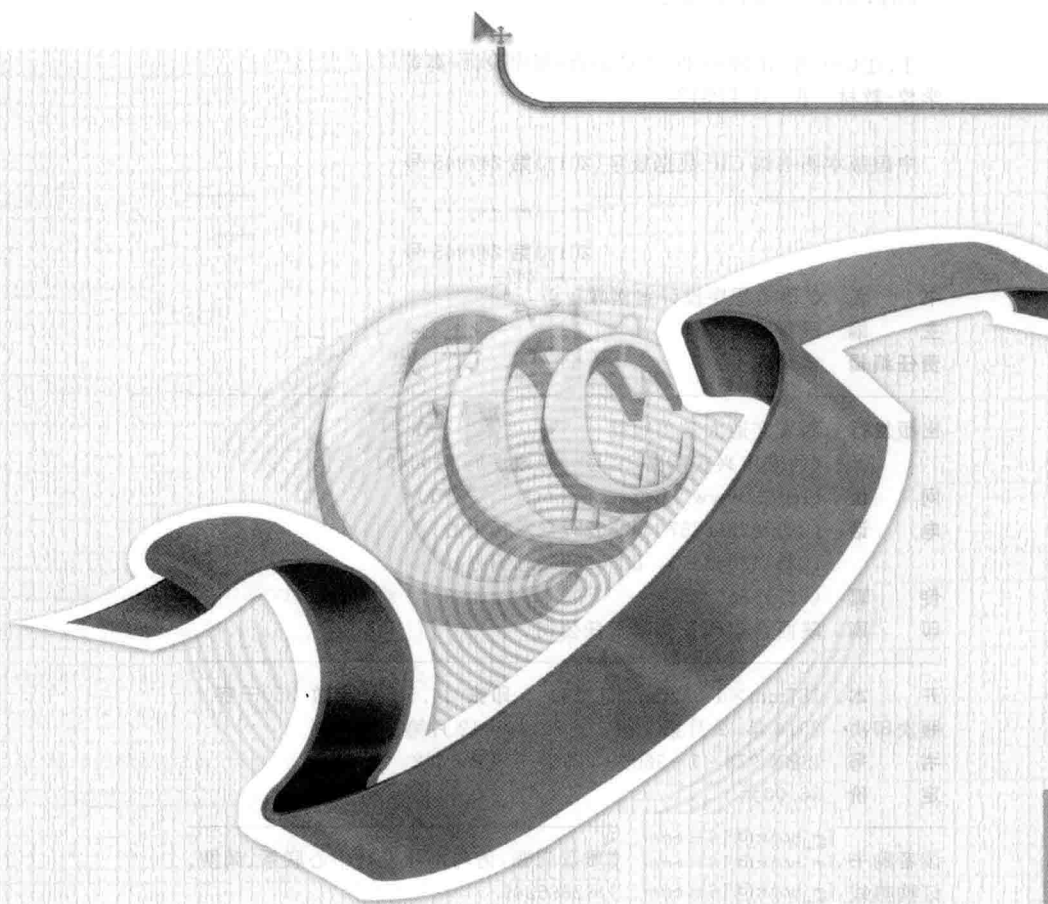
普通高等教育“十二五”应用型本科规划教材  
山西省教育科学规划十二五规划课题成果

# C语言程序设计案例教程

主 编 段新娥

副主编 李 荣

参 编 邸东泉 段海英 王海荣 周锁成 韩俊芳



西安交通大学出版社  
XI'AN JIAOTONG UNIVERSITY PRESS

## 内容提要

C 语言是一种面向问题的通用程序设计语言,具有语言简洁、类型丰富、结构完整、表达力强、直接操作内存单元、适用于模块化结构等特点。本书对 C 语言编程基础进行了由浅入深、循序渐进的介绍。全书共分 12 章,包括 C 语言的特点;C 语言开发环境介绍;C 语言中数据类型、运算符、表达式;模块化设计;数组;函数;指针;结构体;文件;位运算等。书中重要知识点都与实例紧密结合,有助于读者理解知识、掌握知识、应用知识。

书中内容安排合理,讲解浅显易懂,每章有学习方法指导、典型案例做参考,非常实用。本书可以作为各类高等院校计算机专业程序设计基础课程教材或非计算机专业程序设计公共基础课程的教材,也可用于相关技术人员培训和程序设计爱好者自学的参考书。

---

### 图书在版编目(CIP)数据

C 语言程序设计案例教程/段新娥主编. —西安:  
西安交通大学出版社,2014. 11  
ISBN 978-7-5605-6796-9

I. ①C… II. ①段… 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(2014)第 247945 号

---

书 名 C 语言程序设计案例教程  
主 编 段新娥  
责任编辑 雷萧屹

---

出版发行 西安交通大学出版社  
(西安市兴庆南路 10 号 邮政编码 710049)  
网 址 <http://www.xjtupress.com>  
电 话 (029)82668357 82667874(发行中心)  
(029)82668315 82669096(总编办)  
传 真 (029)82668280  
印 刷 陕西奇彩印务有限责任公司

---

开 本 787mm×1 092mm 1/16 印张 14.5 字数 350 千字  
版次/印次 2014 年 12 月第 1 版 2014 年 12 月第 1 次印刷  
书 号 ISBN 978-7-5605-6796-9/TP·642  
定 价 36.00 元

---

读者购书、书店添货,如发现印装质量问题,请与本社发行中心联系、调换。  
订购热线:(029)82665248 (029)82665249  
投稿热线:(029)82669097 QQ:8377981  
读者信箱:lg\_book@163.com

版权所有 侵权必究

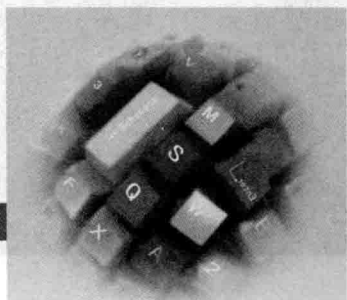
## C 语言程序设计案例教程 前言

## FOREWORD

C 语言是一种面向问题的通用程序设计语言,具有语言简洁、类型丰富、结构完整、表达力强、直接操作内存单元、适用于模块化结构等特点。C 语言既具有高级语言的优点,又具有低级语言的许多特点,由于它的显著优点,这门课的学习将直接为计算机专业的后续专业课如汇编语言、单片机、数据结构、计算机集成控制技术、网络技术等课程打下基础。通过理论和实践教学,使学生较好地掌握结构化编程的思想和思路;养成良好的编程习惯;学会独立和合作编写一定质量的程序;灵活运用 C 语言本身的特点来完成对问题对象的简单模型建构和方法的初步实现。

多年教学过程中,作者选用过同类教材多部,这些教材有的内容充实、知识新颖,有的案例丰富、讲解生动,但针对学生特点和学校有限的课时,这些教材又或多或少显现出不适宜,要么内容太多、太深,教学课时不够,学生难以接受;要么虽案例丰富但内容讲解不完整、不系统。针对这些问题,作者结合多年来的 C 教学经验和开发体会,在参考了国内外同类优秀教材,并与多名从事本教学教师研究后,确定教材编写内容,编写风格,编写出本套教材。本书可以作为各类高等院校计算机专业程序设计基础课程教材或非计算机专业程序设计公共基础课程的教材,也可用于相关技术人员培训和程序设计爱好者自学的参考书。为更好地推广本书的使用,让更多的读者受益,拟申报本套教材为规划教材。与同类教材相比本教材具有以下几个显著特点:

(1)内容精练、重点突出。C 是一个复杂的知识体系,其包含内容很多,为适应市场需求,且在有限的时间内将基础的、关键的知识介绍给读者,作者在内容组织上作了精心的选择和组织,力求使知识讲解全面、系统,突出重点,强调实用。全书每章都由知识讲解、例题分析、综合实例、本章小结、课后习题几部分构成,这样教材不仅有理论介绍,还有相应的练习,为进一步理解掌握抽象知识提供了保障。



# C 语言程序设计案例教程

(2)实例丰富、举一反三。C 程序设计课程是一门专业基础课,学生以前没有任何程序设计基础,针对这一现象,教材编写过程中,作者从不同角度设计了大量实例,将抽象的 C 概念、技术尽量同比较直观的与生活实际密切联系的实例结合起来,每个知识点、设计对应的生动小示例,每章最后设计一个综合实例。通过将实例融入知识讲解中,使知识与实例相辅相成,让学生对所学知识先形成比较深刻的感性认识,再带着好奇去探究其原理及应用,最终达到既有利于学生学习知识,又有利于指导实践的目的。

(3)由浅入深、言简意赅。本书体系结构力求由浅入深,循序渐进,理论与实践相结合,书中理论讲解通俗易懂、言简意赅;例题设计的宗旨是加深对概念、编程思想、编程方法的理解和说明,追求既简单又有典型性且完整的目标。

(4)资源丰富、指导到位。为更好地帮助读者理解和应用本教材的内容,本教材配有实验教材、习题指导、电子课件等教学资料。

本书由段新娥担任主编,负责整体结构的设计和全书的统稿定稿。李荣担任副主编。具体编写分工如下:第2、3、6、7章以及附录由段新娥编写,第1章由王海荣编写,第8章由周锁成编写,第4、5章由段海英编写,第9章由李荣编写,第10、11章由邸东泉编写,第12章由韩俊芳编写。

感谢各校专家、教师和广大读者长期以来的支持和帮助! 由于时间紧迫以及编者水平有限,教材中难免出现疏漏和不足,恳请读者批评指正,使本书得以改进和完善。

编者

2014 年 7 月

# C 语言程序设计案例教程 目录

## CONTENTS

### 第一章 C 语言概述

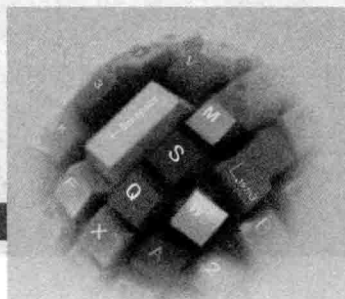
- 1.1 C 语言发展过程和特点 /001
- 1.2 C 语言程序结构 /003
- 1.3 C 程序的运行与开发环境 /006
- 小结 /022
- 习题 /022

### 第二章 基本数据类型、运算符与表达式

- 2.1 C 数据类型概述 /024
- 2.2 标识符、常量和变量 /025
- 2.3 整型数据 /027
- 2.4 实型数据 /028
- 2.5 字符型数据 /030
- 2.6 不同类型数据间的转换 /033
- 2.7 运算符和表达式 /035
- 2.8 数据类型综合案例 /040
- 小结 /041
- 习题 /042

### 第三章 顺序结构语句

- 3.1 结构化程序设计 /044
- 3.2 C 语言中的基本语句 /051
- 3.3 C 语言中的标准输入输出函数 /052
- 3.4 顺序结构程序设计综合案例 /062
- 小结 /063
- 习题 /064



## 第四章 选择结构控制语句

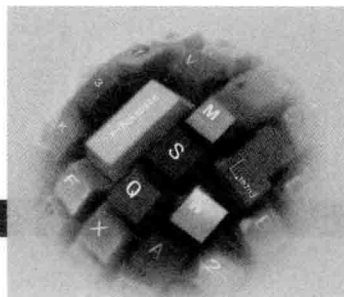
- 4.1 选择结构 /067
- 4.2 if 语句 /067
- 4.3 条件表达式构成的选择结构 /070
- 4.4 switch 语句 /071
- 4.5 选择结构程序设计典型案例 /073
- 小结 /074
- 习题 /074

## 第五章 循环结构控制语句

- 5.1 循环结构概述 /076
- 5.2 goto 语句以及用 goto 语句构成循环 /077
- 5.3 while 语句 /077
- 5.4 do-while 语句 /078
- 5.5 for 语句 /079
- 5.6 循环的嵌套 /081
- 5.7 几种循环的比较 /083
- 5.8 控制语句 /083
- 5.9 循环结构程序设计典型案例 /083
- 小结 /084
- 习题 /084

## 第六章 数组

- 6.1 引言 /087
- 6.2 一维数组 /087
- 6.3 二维数组 /093
- 6.4 字符数组与字符串 /096
- 6.5 数组典型案例 /104
- 小结 /106
- 习题 /107



## 第七章 函数

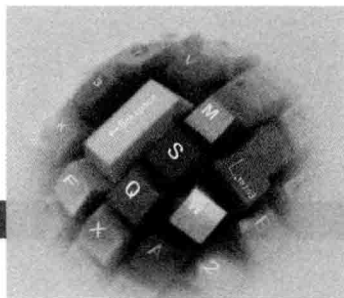
- 7.1 概述 /110
- 7.2 函数参数和函数的值 /114
- 7.3 函数的调用 /118
- 7.4 函数的嵌套调用和递归调用 /119
- 7.5 数组作为函数参数 /123
- 7.6 函数典型案例 /127
- 小结 /128
- 习题 /128

## 第八章 用户标识符的存储类别与预处理

- 8.1 变量的作用域 /132
- 8.2 变量的存储类别 /134
- 8.3 内部函数和外部函数 /138
- 8.4 如何运行一个多文件的程序 /139
- 8.5 编译预处理 /140
- 8.6 综合案例 /148
- 小结 /149
- 习题 /150

## 第九章 指针

- 9.1 地址和指针的概念 /153
- 9.2 变量的指针和指向变量的指针变量 /153
- 9.3 指针与数组 /156
- 9.4 指针与字符串 /161
- 9.5 指针与函数 /166
- 9.6 指针典型案例 /171
- 小结 /173
- 习题 /173



## 第十章 结构体与共用体

- 10.1 结构体类型的说明与结构体变量的定义 /176
- 10.2 结构体类型变量的初始化及引用 /177
- 10.3 结构体数组 /178
- 10.4 共用体 /180
- 10.5 枚举类型 /182
- 10.6 典型案例 /183
- 小结 /185
- 习题 /185

## 第十一章 文件

- 11.1 C 文件概述 /187
- 11.2 文件的打开和关闭 /187
- 11.3 文件的读写 /190
- 11.4 文件的定位、测试与随机读取 /195
- 11.5 典型案例 /199
- 小结 /201
- 习题 /202

## 第十二章 位运算

- 12.1 位运算符和位运算 /204
- 12.2 位运算举例 /209
- 12.3 典型案例 /209
- 小结 /210
- 习题 /210

## 附录

- 附录 A 常用字符与 ASCII 码对照表 /212
- 附录 B C 语言中的关键字及含义 /216
- 附录 C 运算符优先级对照表 /217
- 附录 D C 语言常用库函数 /219

## 参考文献

### ► 学习目标

- ☆ 了解 C 语言的发展和特点。
- ☆ 掌握 C 程序的基本组成结构。
- ☆ 熟悉 C 程序的开发环境。

## 1.1 C 语言发展过程和特点

上课前大家先思考一个问题,计算机为什么能按照人们的意愿完成诸如做数学题、查英语单词、统计游戏等级、管理信息等一些任务呢?这是因为计算机内有人们预先编写好的指挥计算机工作的指令,一个指令规定计算机执行的基本操作。计算机运行时,将这些指令取出来进行执行,最后将结果返回给操作者。由这些指令组成的序列称为程序。计算机从诞生到现在,能编写这些指令的语言有多种,经历了从低级语言到高级语言的转变,C 语言属于高级语言中的一种。

### 1.1.1 程序和程序设计

程序是计算机可以执行的、为解决特定问题用某种计算机语言编写的指令序列。计算机可以直接执行的程序称为可执行程序,其扩展名一般为 EXE、COM,可执行程序通常以文件方式存放在磁盘上,当需要执行某程序时,计算机就将该程序装入内存,等待执行。它包含的主要部分是二进制编码的机器指令和数据。所谓机器指令是能直接控制计算机的每一个部件的基本动作,机器指令的表达方式称为“机器语言”。计算机科学家沃思(Nikiklaus Wirth)把程序描述为:

程序 = 数据结构 + 算法

这说明程序是由数据结构与算法组成。数据结构是对数据的描述,在程序中要指定数据的类型和数据的组织形式;算法是对操作的描述,即操作步骤。由此可见设计一个合理的数据存储结构和算法对提高程序的执行效率起着很重要的作用。

程序设计就是根据计算机要完成的任务进行数据结构和算法设计,并且编写其程序代码,进行调试,直到得出正确结果的过程。其基本过程通常如下:

- (1) 分析问题,明确要解决的问题和要实现的功能。
- (2) 将具体问题抽象为数学问题,建立数学模型,确定合适的解决方案。
- (3) 确定数据结构,并根据数据结构设计相应的算法,写出算法描述。

(4)编写程序。

(5)调试并运行程序,直到得到正确结果。

程序设计方法经历了由传统的结构化程序设计(面象过程)到面向对象的设计。结构化程序设计是以函数为单位,采用模块分解与功能抽象,自顶向下、分而治之的方法,有效地将一个较复杂的程序设计任务分解成许多易于控制和处理的模块(函数)。各模块之间尽量相对独立,便于开发和维护。而面向对象的程序设计的思路与面向过程的程序设计的思路不同,面向对象程序设计是以类和对象为基础,程序的操作是围绕对象进行,相对于面向过程,面向对象程序设计处理问题的思路和人们日常生活中更相似,更易于调试和维护。但不要把两种设计方法对立起来,面向过程与面向对象程序设计各有用途、互为补充,且在面向对象程序设计过程中仍然要用到面向过程程序设计和知识。

不同的程序设计方法需要选用不同的编程语言进行编写程序,面向过程的编程语言有:汇编语言、BASIC、PASCAL、C 语言等,面向对象的编程语言有:C++、C#、Smalltalk、JAVA 语言等。

### 1.1.2 C 语言的发展及应用

C 语言是国际上广泛流行的计算机高级语言。它适合作为系统描述语言,既可以用来编写系统软件,也可用来编写应用软件。在 C 语言产生之前,系统软件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,为了提高系统软件的可读性和可移植性,最好改用高级语言,而一般高级语言又难以实现汇编语言对计算机硬件的直接操作,人们设想能否找到一种既具有一般高级语言特性,又具有低级语言特性的语言,于是 C 语言就在这种情况下应运而生。

C 语言是美国贝尔实验室于 1972 年至 1973 年间,在 B 语言的基础上设计出来的,最初只是为描述和实现 UNIX 操作系统提供一种工作语言而设计。后来,C 语言又被多次改进,并出现了许多版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,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 语言的发展和扩充,制定了一套新的 C 语言标准,称为 ANSI C。

因为 C 语言具有高级语言的特点,又具有汇编语言的特点,所以可以作为工作系统设计语言,编写系统应用程序,也可以作为应用程序设计语言,编写不依赖计算机硬件的应用程序。它几乎可用于所有领域,如嵌入式、便携式计算机、电视、电话、手机和其他大量设备,主要包括以下几个方面:

- (1)操作系统平台开发。
- (2)游戏软件开发。
- (3)嵌入式系统开发。
- (4)桌面应用系统开发。
- (5)交互式系统开发。
- (6)多媒体系统开发。
- (7)网络系统开发。
- (8)底层接口系统开发。

(9)软硬件系统设备开发。

C语言无处不在,它可应用于任何地方、任何领域,并且已拥有几百万个用户,其应用范围要强于任何一种计算机语言。如果仔细观察,就会发现C语言就在我们身边,例如经常使用的Windows、Linux操作系统等。

### 1.1.3 C语言的特点

C语言之所以能存在和发展,并具有强大的生命力,主要归功于其所具有的以下鲜明特点:

(1)C语言数据类型丰富,运算符丰富方便。C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。这些数据类型能用来实现各种复杂的数据结构(如链表、树、栈等)的运算,尤其是指针类型数据,使用起来比其他语言更为灵活、多样;C语言的运算符包含的范围很广,共有34种运算符(见附录A)。C把括号、赋值、强制类型转换等都作为运算符处理,从而使C的运算类型极其丰富,表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(2)程序设计结构化。结构化语言的显著特点是代码、数据的模块化,C程序是以函数形式提供给用户的,这些函数调用都很方便。C语言具有多种条件语句和循环控制语句(如if语句、switch语句、while语句、do...while语句、for语句、break语句和continue语句等),使程序完全结构化。

(3)C语言能进行位(W)操作。C语言能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C语言既具有高级语言的功能,又具有低级语言的许多功能,可用来编写系统软件,也可用来编写应用软件。C语言的这种双重性,使得它既是成功的系统描述语言,又是通用的程序设计语言。有人把C称为“高级语言中的低级语言”或“中级语言”,意为兼有高级语言和低级语言的特点。

(4)生成目标代码质量高,程序执行效率高。C程序一般只比汇编程序生成的目标代码的执行效率低10%~20%,却比其他高级语言的执行效率高。C语言的移植性非常好。用C语言编写的程序几乎不做修改就可用于各种计算机或各类操作系统。

C语言的这些特点使得C语言很快应用到了各计算机应用领域中的软件编写,如数据库管理、CAD、科学计算、图形图像处理、实时控制等软件。虽然近年来,软件开发领域出现了一些面向对象程序设计语言,如C++、JAVA等,但这些语言只是为解决编写大型软件的问题而产生的,学起来比C语言困难得多,更何况将来并不是所有的人都去编写大型软件。对于计算机专业学生来说,由于后续操作系统、数据结构等课程都是采用C作为背景语言进行讲解的,所以学好C语言将为以后课程的学习打下坚实的编程基础,熟练掌握C语言也成为计算机开发人员的一项基本功。

## 1.2 C语言程序结构

如何编写程序完成将结果显示在屏幕上,或者让计算机完成求两数据的和运算呢?在完成这一任务前先看几个简单的C程序,了解C语言程序的基本结构后,再来完成上述

任务。

### 1.2.1 C 程序几个简单实例

**【例 1-1】**在屏幕上输出一串字符“Hello, welcome you C program world!”。

程序如下：

```
#include<stdio.h>
/* example1-1.c 在屏幕上输出字符串 */
void main(){
printf("Hello, welcome you C program world! \n");
}
```

程序运行结果：

Hello, welcome you C program world!

程序说明：

(1)程序第一条语句#include<stdio.h>称为文件包含命令,其作用是把系统目录下的头文件 stdio.h 加载到本程序中,说明本程序中要用到这个文件中的函数。stdio 是“standard input & output”的缩写,即文件里的函数主要是处理数据流的标准输入/输出。其扩展名为.h,故该文件称为头文件或首部文件。在编程时,程序员可以加载系统文件中的函数,也可以加载用户自定义文件中的函数,当需加载系统目录下的头文件时,将文件名用“<>”括起来,如< stdio.h >。若加载用户自定义的文件,则文件名不需“<>”括起。

(2)/\* example1-1.c 在屏幕上输出字符串 \*/是一条注释语句,C 语言中可以用/\* ..... \*/对程序中的任何部分做注释。注释语句只起解释作用,不被编译器编译。

(3)main 是函数的名字,在此表示“主函数”。每一个 C 语言源程序都有且仅有一个 main() 函数,它是程序执行的起点。main 前面的 void 表示此函数是“空类型”,即执行此函数后不产生一个函数值。函数体用{}括起,其中包含由一条或多条语句。

(4)printf() 函数是一个库函数,其函数原型在头文件 stdio.h 中,该函数的功能是将圆括号内的内容输出到显示器上。“\n”是光标换行符,分号表示一个 C 语句的结束。

**【例 1-2】**求两个数的和。

程序如下：

```
/* example1-2.c      求两个数的和 */
#include<stdio.h>
void main(){
int num1, num2, sum;           /* 定义三个整型变量 */
num1 = 8; num2 = 12;           /* 给加数和被加数赋值 */
sum = num1 + num2;             /* 求两个数的和,并将结果存入整型变量 sum 中 */
printf("8 + 12 的和为: %d\n",sum); /* 将结果显示到屏幕上 */
}
```

程序运行结果：

8 + 12 的和为:20

程序说明:

(1)本程序也是由一个主函数构成的程序,但不同于【例 1-1】的是 main() 函数是由多条语句构成,其中“int num1, num2, sum;”语句是声明语句,声明了三个整型变量;其余为执行语句,前三条语句为三个整型变量赋值,给 num1, num2 分别赋值 8 和 12,然后将 num1 与 num2 的和赋给 sum。最后一条语句是输出语句,“%d”表示以“十进制”的格式输出数据(详见 printf 格式输出章节)。

(2)主函数是由声明和执行两部分组成。声明是指对变量的类型说明。C 语言规定,源程序中所有用到的变量都必须先声明后使用,否则编译时会出现语法错误(例 1-1 中没有用到变量,所以没有声明部分)。

**【例 1-3】**求任意两个数的和。

程序如下:

```
/* example1-3.c                                求任意两个数的和 */
#include<stdio.h>
int add(int x, int y);                          /* 加法器的函数原型声明 */
void main(){
    int num1, num2, sum;
    printf("please input value of num1 and num2:\n");
    scanf("%d %d", &num1, &num2);             /* 通过键盘给加数和被加数赋值 */
    sum = add(num1, num2);                      /* 调用函数求两个数的和 */
    printf("%d + %d = %d\n", num1, num2, sum);
}
int add(int x, int y){
    int z;
    z = x + y;
    return(z);
    /* return(x + y); */
}
```

程序运行结果:

```
please input value of num1 and num2:
10 20
10 + 20 = 30
```

程序说明:

(1)本程序包含 main 主函数和求和函数 add() 两个函数。

(2)add() 函数的作用是求 x 和 y 的和,函数返回值是两数的和。return 语句将 z 的值返回给主调函数 main(), 语句“sum=add(num1, num2);”就是 main() 函数通过调用 add() 来完成(num1+ num2)的计算,并将计算结果赋给变量 sum。

(3) `add()` 函数体的语句可以简写成一条语句“`return(x+y);`”。

### 1.2.2 C 程序结构特征

以上三个例子功能简单,却包含了 C 语言程序的基本结构特征,概括来说,一个 C 语言程序具有如下结构特征:

(1) 一个 C 语言程序通常由下面几个部分组成:

- ① 文件包含部分。
- ② 变量说明部分。
- ③ 函数原型声明部分。
- ④ 主函数部分。
- ⑤ 用户自定义函数部分。

但并不是每一个程序都必须包含上面 5 部分,一个最简单的 C 程序可以只含文件包含和主函数部分。

(2) 每一个 C 语言源程序都必须由一个且仅一个 `main()` 函数(又称主函数)和若干个其他函数结合而成,或仅由一个主函数构成。主函数的组成形式如下:

```
main() {  
    变量声明部分  
    执行语句部分  
}
```

(3) 每一条语句由分号结束,每行上可以写一条语句也可以写多条语句,同样一条语句也可以写在多行上,当一条语句写在多行上时,两行之间用“\”作续行符。

(4) 程序中允许使用注释。其中“`/*`”与“`*/`”是成对出现,且“`/`”与“`*`”以及“`*`”与“`/`”之间不能有空格。注释语句可以单独占用一行,也可以跟在语句之后。

## 1.3 C 程序的运行与开发环境

在计算机上如何实现上面三个例题呢?也就是代码在什么地方输入,计算机如何完成这一过程呢?

### 1.3.1 C 程序的运行

用高级程序语言编写出来的程序称为源程序,计算机并不能直接理解这样的语言,计算机只能理解机器语言,因此必须将程序语言翻译成机器语言,计算机才能理解。将源程序翻译成机器语言的过程称为编译,编译的结果是得到源程序的目标代码;最后还要将目标代码与系统提供的函数和自定义函数链接起来,就可以得到计算机能执行的程序。机器可执行的程序称为可执行程序。C 语言编写的程序要想通过计算机运行得到结果,同样要经过以上过程。

#### 1. C 语言源程序的编写

C 语言源程序的编写过程称为源程序的编辑,C 语言源程序可以在所有字处理软件(记

事本、WORD、EDIT 等)下编写,也可以用 C 语言集成开发工具(Turbo C、Visual C++等)进行编写。编写完成后,文件要以“.c”为扩展名保存文件。也就是说 C 语言源程序文件的扩展名为“.c”。例如【例 1-3】中的源程序文件编写完成后保存成“example1-3.c”。

## 2. C 语言源程序的编译

计算机不能直接执行源程序,必须经过“编译—链接”,生成可执行文件后,才能执行。

编译由编译器来完成,是指用 C 语言提供的编译器将编辑好的源程序翻译成二进制形式的目标代码文件的过程。目标代码文件的扩展名为“.obj”,又称 OBJ 文件。

例如对【例 1-3】中的源程序文件“example1-3.c”,在 Turbo C 环境下执行“compile”菜单中的“compile to OBJ”命令,对“example1-3.c”源程序文件进行编译。若编译成功生成“example1-3.obj”文件,否则弹出错误提示信息。

C 语言编译器在编译过程中,主要完成两个任务:一是执行程序中的预处理命令,二是进行词法和语法分析,检查源程序每一条语句的语法和词法错误。检查出的若是致命错误,编译系统将终止程序继续编译,不会生成 OBJ 文件,必须修改程序重新编译;检查出的若是警告错误,说明源程序在此处存在错误,但这种错误不会影响连接生成可执行程序,只是可能会影响程序运行结果,这时要根据具体情况确定是否需要修改。

## 3. 目标文件的链接

通过编译产生的目标代码程序尽管是机器语言的形式,但却不是机器可以执行的方式,这是因为目标程序只是一些松散的机器语言,要获得可执行的程序,还需将它们链接起来。开发软件过程中,为了支持软件的模块化,允许程序语言在不同的时期开发出具有独立功能的软件模块作为一个单元,一个可执行的程序中有可能包含一个或多个这样的程序单元,这样可以降低程序的重复所带来的低效率。

程序的链接工作由链接器来完成,链接器的任务就是将目标程序和与该目标程序有关的其他目标程序链接起来,生成扩展名为“.exe”的可执行程序,这种可执行的程序是一种可存储在磁盘存储器上的文件。

例如【例 1-3】中的源程序文件“example1-3.c”编译后生成目标文件“example1-3.obj”,现在就可以利用链接器生成“example1-3.exe”可执行代码。具体操作是在 Turbo C 环境下执行“compile”菜单中的“Link EXE File”命令,若没有错误,则可将目标程序和函数库链接,生成可执行文件“example1-3.exe”,否则需要检查程序中是否有错写函数名或缺少文件包含命令的情况。

一旦生成了可执行程序,就可以反复地被加载执行,而不需要重新编译、链接,如果修改了源程序,也不会影响到已生成的可执行程序,除非对修改后的源程序重新编译和链接。

## 4. 运行程序,查看结果

运行程序的目的是要得到最终的结果。例如【例 1-3】要想看到程序执行的结果,在 Turbo C 环境下需要执行“Run”菜单中的“Run”命令或按“ctrl+F9”键,就可执行已装入到 Turbo C 中的程序了。如果出现错误或运行结果不符合原意,可重新修改程序,再进行编译、链接和运行,直到结果正确为止。

说明:

①完成一个 C 语言程序的完整过程主要包括编辑、编译、链接、加载并运行 4 部分。

②上面对程序进行的编译、链接都只针对一个源程序文件,实际上,可以对多个源程序文件通过编辑、链接成一个可执行的文件,但被链接成可执行程序的目标程序中,只允许在一个程序中有且仅有一个可被加载的入口点,即只允许在一个源程序中包含一个 `main()` 函数。

③上面对 C 语言进行编译、链接的方式并不是唯一的,具体要参考编程人员选用的编译器的使用说明。

### 1.3.2 开发环境

在开发 C 语言程序之前,首先需要检查所用的计算机系统是否已经安装了 C 编译系统,如没有首先要下载并安装相应开发工具软件。C 语言程序开发工具主要有 Turbo C、Visual C++ 6.0、Dev-C++、Eclipse for C/C++。常用的 C 语言开发工具有 Turbo C 2.0、Turbo C/C++ 3.0、Visual C++ 6.0。Turbo C 2.0 是用于 DOS 环境下,在进入 Turbo C 2.0 集成环境后,不能用鼠标进行操作,主要通过键盘选择菜单;Turbo C/C++ 3.0 的开发界面和操作方法与 Turbo C 2.0 基本相同,只是进入 Turbo C/C++ 3.0 集成环境后,还能用鼠标进行操作;Visual C++ 是微软推出的一款 C++ 编译器,可提供编辑 C 语言,是一款功能强大的可视化软件开发工具。本书中所有的程序都可以选择在 Turbo C 2.0 或 Visual C++ 6.0 开发环境中进行编写调试。为了说明在 DOS 和 WINDOWS 环境下如何开发 C 语言程序,下面我们将对 Turbo C 2.0/ Visual C++ 6.0 这两款开发工具的使用方法进行具体的介绍。

#### 1. Turbo C 开发环境

Turbo C 是美国 Borland 公司推出的产品,Turbo C 常常被简称为 TC,TC 因小巧、简单、直观的操作赢得了不少 C 语言学习者的青睐。TC 最大的优点在于为用户提供一个集成开发环境,将程序的编辑、编译、链接和运行等操作全部集中在一个界面上进行,使得操作非常方便。Borland 公司在 1987 年首次推出 Turbo C 1.0 产品,而现在常用的 Turbo C 2.0 是在 1989 年推出的,随着 Turbo C 2.0 的推出,Turbo C 1.0 逐渐被淘汰。近年来,不少人也使用 Turbo C/C++ 3.0。为了介绍如何在 DOS 环境下使用 Turbo C 2.0 开发 C 语言程序,这里结合一个实例进行操作讲解,更能达到贴近动手操作的目的。

##### 1) Turbo C 2.0 下载安装

首先下载一个 Turbo C 2.0 的压缩包并解压(可以解压到任何地方),例如,解压放在 C 盘中的 Program Files/TC 目录下。解压后文件夹的目录如图 1-1 所示,其中在解压后的文件夹中找到一个名为 TC.EXE 图标和两个文件夹 INCLUDE 和 LIB。说明本系统具有了 Turbo C 2.0 编译器,可以进行 C 语言程序的开发。

##### 2) 界面介绍

双击 TC.EXE 图标,进入 Turbo C 2.0 集成开发环境主界面,屏幕上出现如图 1-2 所示界面。

其中最上方为 Turbo C 2.0 主菜单,显示下列内容:File(文件)、Edit(编辑)、Run(运行)、Compile(编译)、Project(项目)、Options(选项)、Debug(调试)、Break/watch(中断/观察)。除 Edit 外,其他各项均有子菜单,只要用 Alt 加上某项中第一个字母(即大写字母),就