

高等学校规划教材  
国家精品课程系列教材



# C语言程序设计

董卫军 邢为民 索琦 编著  
耿国华 主审

计算机公共基础课程群



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

高等学校规划教材  
国家精品课程系列教材

# C 语言程序设计

董卫军 邢为民 索琦 编著  
耿国华 主审

电子工业出版社  
Publishing House of Electronics Industry  
北京·BEIJING

## 内 容 简 介

本书是国家精品课程“计算机基础”系列课程“C 语言程序设计”的主教材，全书内容包括两大部分：第一部分为 C 语言基础篇，主要包括 C 语言概述、基本数据类型与运算、语句与流程控制、数组、指针、函数、复杂数据类型、编译预处理和文件的操作；第二部分为程序设计提高篇，主要包括程序设计方法和工程化程序设计。

本书体系完整、结构严谨、注重应用、强调实践，在编写时兼顾了计算机等级考试的要求。为方便教学，本书还配有电子课件，任课教师可登录华信教育资源网（[www.hxedu.com.cn](http://www.hxedu.com.cn)）免费注册下载。

本书可作为高等学校计算机程序设计基础课程的教材，也可作为全国计算机等级考试二级 C 语言的培训或自学教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

### 图书在版编目(CIP)数据

C 语言程序设计/董卫军, 邢为民, 索琦编著. —北京: 电子工业出版社, 2011.6

高等学校规划教材

ISBN 978-7-121-13573-6

I. ①C… II. ①董…②邢…③索… 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 089453 号

策划编辑: 索蓉霞

责任编辑: 索蓉霞 文字编辑: 张 京

印 刷: 北京市顺义兴华印刷厂

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 16 字数: 410 千字

印 次: 2011 年 6 月第 1 次印刷

定 价: 29.50 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。



# 前 言

程序设计在理工科大学生的知识体系中占有重要位置，其内容组织应该体现创造性思维的素质教育培养过程。面对信息新技术发展和国家对人才信息素质培养的需求，本教材力图在遵循教育和学习规律的基础上，优先注重内容在应用上的层次性，适当兼顾整体在理论上的系统性，注重学习、掌握、使用计算机的知识与技能，突出知识点与技术点的关系。

本书是国家精品课程“计算机基础”系列课程“C 语言程序设计”的主教材。本教材与传统 C 语言教材以语法介绍为主的书写方式不同，采用“问题导入语法、语法引入应用、应用面向实践”的方式，突出知识点与技术点的关联性，注重内容在应用上的层次性，兼顾整体在理论上的系统性，在有限的时间内使教学者传授更多的知识，使学习者学以致用，真正理解程序设计及其思想。

全书共 11 章，分 C 语言基础篇和程序设计提高篇两大部分，对 C 语言及程序设计的基本概念、原理和方法，从基本语句、基础使用、理论提升三个层面逐层展开。

基本语句层面从培养学生程序设计基本概念和初步逻辑思维能力入手，主要包括 C 语言概述、基本数据类型与运算、语句与流程控制三方面的知识，重点突出 C 语言的基本数据类型和程序控制的基本构架。通过学习，使学习者初步掌握 C 语言的基本语法和程序设计的基本概念。

基础使用层面从培养学生分析问题和解决问题的能力入手，主要包括数组、指针、函数、复杂数据类型、编译预处理和文件操作六方面内容。通过学习，使学习者初步掌握分析问题和解决问题的方法。

理论提升层面通过引入程序设计方法和工程化程序设计的理论，强化逻辑思维能力和程序设计能力培养，进一步提高分析问题和解决问题的能力，使学习者真正掌握程序设计技能。

本书体系完整、结构严谨、注重实用、强调实践，在编写时兼顾了计算机等级考试的要求。为方便教学，本书还配有电子课件，任课教师可登录华信教育资源网（[www.hxedu.com.cn](http://www.hxedu.com.cn)）免费注册下载。

本书由多年从事计算机教育的一线教师编写，董卫军编写第 4~6 章、第 11 章及附录，邢为民编写第 1~3 章和第 7 章，索琦编写第 8~10 章。全书由董卫军统稿，西北大学耿国华教授主审。感谢教学团队成员的帮助，感谢西北大学教务处多年来的支持。由于作者水平有限，书中难免有不妥之处，恳请读者指正。

编 者  
于西安·西北大学

# 目 录

## C 语言基础篇

<b>第 1 章 C 语言概述</b> .....	1	2.9.1 算术运算符和算术表达式	18
1.1 程序与程序设计 .....	1	2.9.2 简单赋值运算符和赋值表达式	21
1.2 C 语言的发展和特点 .....	2	2.9.3 关系运算符和关系表达式	23
1.2.1 C 语言的发展 .....	2	2.9.4 逻辑运算符和逻辑表达式	24
1.2.2 C 语言的特点 .....	3	2.9.5 自增、自减运算符	25
1.3 C 语言的程序结构及特点 .....	4	2.9.6 条件运算符和条件表达式	27
习题 1 .....	6	2.9.7 逗号运算符和逗号表达式	28
<b>第 2 章 基本数据类型与运算</b> .....	8	2.9.8 数据类型长度运算符	29
2.1 类型的重要性 .....	8	2.9.9 算术自反赋值运算符	30
2.2 C 语言的基本数据类型 .....	8	2.10 位运算 .....	30
2.3 标志符 .....	9	2.10.1 逻辑位运算符	31
2.4 常量与变量 .....	10	2.10.2 移位运算符	32
2.4.1 常量 .....	10	2.10.3 位自反赋值运算符	34
2.4.2 符号常量 .....	10	2.11 运算符的结合性和优先级	
2.4.3 变量 .....	11	总结 .....	35
2.4.4 变量的定义 .....	11	习题 2 .....	36
2.5 整型数据 .....	12	<b>第 3 章 语句与流程控制</b> .....	39
2.5.1 整数类型 .....	12	3.1 三种基本的程序控制结构	39
2.5.2 整型常量 .....	12	3.2 数据的输入与输出	39
2.5.3 整型变量 .....	13	3.2.1 格式化输出函数	40
2.6 实型数据 .....	13	3.2.2 格式化输入函数	42
2.6.1 实型类型 .....	13	3.3 字符输入与输出函数	44
2.6.2 实型常量 .....	14	3.3.1 字符输入函数	44
2.6.3 实型变量 .....	14	3.3.2 字符输出函数	45
2.7 字符型数据 .....	14	3.4 顺序结构语句	45
2.7.1 字符型常量 .....	15	3.4.1 语句概述	45
2.7.2 字符型变量 .....	16	3.4.2 顺序结构语句概述	46
2.7.3 字符串常量 .....	16	3.4.3 程序举例	47
2.8 变量赋值 .....	17	3.5 选择结构语句	49
2.8.1 变量初始化 .....	17	3.5.1 选择性问题的	49
2.8.2 变量赋值 .....	17	3.5.2 if 条件语句	50
2.9 运算符和表达式 .....	18		

3.5.3 开关语句 .....	55	5.4.2 通过指针访问二维数组 ...	103
3.6 循环结构语句 .....	57	5.4.3 指向一维数组的指针 变量 .....	104
3.6.1 问题的引入 .....	57	5.5 指针数组与指针的指针 .....	106
3.6.2 While 语句 .....	57	5.5.1 指针数组 .....	106
3.6.3 for 语句 .....	59	5.5.2 指向指针的指针 .....	107
3.6.4 do...while 语句 .....	61	5.6 对指针的几点说明 .....	109
3.6.5 break 和 continue 语句 .....	62	习题 5 .....	109
3.6.6 多重循环 .....	64	<b>第 6 章 函数</b> .....	114
3.6.7 goto 语句和标号 .....	65	6.1 引入函数 .....	114
习题 3 .....	65	6.1.1 函数的概念 .....	114
<b>第 4 章 数组</b> .....	72	6.1.2 使用函数的目的 .....	115
4.1 数组的引入与定义 .....	72	6.1.3 函数的分类 .....	116
4.1.1 问题的引入 .....	72	6.2 函数的定义与说明 .....	116
4.1.2 一维数组的定义 .....	73	6.2.1 自定义函数的定义 .....	116
4.2 一维数组的使用 .....	74	6.2.2 函数的返回值 .....	119
4.2.1 一维数组初始化 .....	74	6.2.3 自定义函数的说明 .....	120
4.2.2 一维数组元素的引用 .....	74	6.3 函数调用 .....	121
4.2.3 一维数组应用举例 .....	75	6.3.1 函数调用的一般形式 .....	121
4.3 二维数组 .....	80	6.3.2 参数传递 .....	122
4.3.1 二维数组的定义及使用 .....	80	6.3.3 函数的递归调用 .....	126
4.3.2 二维数组应用举例 .....	81	6.4 函数指针 .....	128
4.4 字符串与字符数组 .....	84	6.4.1 函数指针变量定义 .....	128
4.4.1 引入字符串 .....	84	6.4.2 函数指针变量的使用 .....	129
4.4.2 字符串与字符数组的概念 ...	84	6.5 函数应用举例 .....	131
4.4.3 字符数组初始化 .....	84	习题 6 .....	132
4.4.4 字符串基本处理函数 .....	86	<b>第 7 章 复杂数据类型</b> .....	137
4.4.5 字符数组应用举例 .....	87	7.1 变量的存储类别 .....	137
习题 4 .....	88	7.1.1 存储空间分类 .....	137
<b>第 5 章 指针</b> .....	93	7.1.2 变量的存储类别 .....	137
5.1 指针的概念 .....	93	7.2 变量的生存期和作用域 .....	141
5.1.1 指针的引入 .....	93	7.2.1 变量的生存期 .....	142
5.1.2 指针变量与指针常量 .....	93	7.2.2 变量的作用域 .....	142
5.2 指针的使用 .....	94	7.2.3 变量的存储类别与作用域 和生存期总结 .....	144
5.2.1 指针变量的定义与初始化 ...	94	7.3 结构体与共用体 .....	144
5.2.2 指针的运算 .....	95	7.3.1 结构体 .....	144
5.2.3 指针的安全 .....	99	7.3.2 共用体 .....	152
5.3 指针与一维数组 .....	99	7.4 枚举类型 .....	155
5.4 指针与二维数组 .....	101		
5.4.1 理解二维数组 .....	101		

7.5 用 typedef 定义类型 .....	158	9.2.3 字节级的文件的读/写 .....	181
习题 7 .....	159	9.2.4 文件结束判断函数 .....	183
<b>第 8 章 编译预处理与外部函数</b> .....	162	<b>9.3 字符串文件读/写</b> .....	186
8.1 编译预处理 .....	162	9.3.1 fgets 函数 .....	186
8.1.1 宏替换 .....	162	9.3.2 fputs 函数 .....	187
8.1.2 文件包含 .....	166	<b>9.4 数据的格式化读/写</b> .....	187
8.1.3 条件编译 .....	168	9.4.1 fprintf 函数 .....	187
8.2 外部函数 .....	170	9.4.2 fscanf 函数 .....	187
8.3 多个源程序文件的编译 连接 .....	172	<b>9.5 记录级的文件读/写</b> .....	188
习题 8 .....	172	9.5.1 fread 函数 .....	189
<b>第 9 章 文件的操作</b> .....	177	9.5.2 fwrite 函数 .....	189
9.1 文件的基本概念 .....	177	<b>9.6 文件位置指针的移动</b> .....	190
9.1.1 C 语言支持的文件格式 ..	177	9.6.1 rewind 函数 .....	191
9.1.2 文件操作的基本思路 .....	178	9.6.2 ftell 函数 .....	191
9.2 文件的基本操作 .....	179	9.6.3 fseek 函数 .....	191
9.2.1 文件指针 .....	179	<b>9.7 文件应用举例</b> .....	192
9.2.2 文件的打开与关闭 .....	180	习题 9 .....	195

## 程序设计提高篇

<b>第 10 章 程序设计方法</b> .....	198	<b>10.5 链表</b> .....	206
10.1 算法 .....	198	10.5.1 空间的申请 .....	207
10.1.1 算法的概念 .....	198	10.5.2 空间的释放 .....	207
10.1.2 算法描述 .....	198	10.5.3 链表的引入 .....	208
10.1.3 算法效率的衡量 .....	199	10.5.4 链表的基本操作 .....	209
10.2 查找 .....	199	<b>10.6 应用举例</b> .....	213
10.2.1 顺序查找法 .....	199	习题 10 .....	216
10.2.2 折半查找法 .....	199	<b>第 11 章 工程化程序设计</b> .....	219
10.2.3 分块查找法 .....	200	11.1 程序设计概述 .....	219
10.3 排序 .....	200	11.1.1 程序设计方法与风格 ..	219
10.3.1 插入排序 .....	200	11.1.2 程序设计的基本步骤 ..	220
10.3.2 交换排序 .....	201	11.1.3 程序设计方法的发展 ..	221
10.3.3 选择排序 .....	201	<b>11.2 结构化程序设计</b> .....	222
10.4 程序设计的基本思维原则 ..	202	11.2.1 结构化程序设计的基本 思想 .....	222
10.4.1 枚举原则 .....	202	11.2.2 结构化程序设计 的优缺点 .....	223
10.4.2 归纳原则 .....	204		
10.4.3 抽象原则 .....	205		

11.3	面向对象程序设计简述 .....	223	11.4.7	软件测试 .....	235
11.3.1	面向对象程序设计的基本 概念 .....	224	11.4.8	软件维护 .....	237
11.3.2	面向对象程序设计的 优点 .....	226	习题 11	.....	238
11.4	工程化程序设计 .....	227	附录 A	Visual C++集成环境使用 指南 .....	240
11.4.1	软件工程的基本概念 .....	227	附录 B	常用运算符及其优先级 和结合性 .....	244
11.4.2	可行性分析 .....	228	附录 C	标准 C 语言头文件 .....	245
11.4.3	需求分析 .....	229	附录 D	C 语言系统关键字 .....	246
11.4.4	概要设计 .....	231	附录 E	ASCII 码表 .....	247
11.4.5	详细设计 .....	232	参考文献	.....	248
11.4.6	编码 .....	234			



# C 语言基础篇

## 第 1 章 C 语言概述

C 语言 (Combined Language) 是一种比较流行的计算机程序设计语言。它既具有高级语言的特点, 又具有汇编语言的特点。它可以作为系统程序设计语言, 编写系统软件; 也可以作为应用程序设计语言, 编写应用软件。因此, 它的应用范围广泛, 不仅用在软件开发上, 在其他计算机应用领域也都需要用到 C 语言, 如单片机及嵌入式系统开发。

### 1.1 程序与程序设计

从自然语言角度来讲, 程序是对解决某个问题的方法步骤的描述; 从计算机角度来讲, 程序是用计算机语言描述的解决问题的方法步骤。程序的执行过程是问题的解决过程, 程序是有始有终的, 每个步骤都能操作, 所有步骤执行完, 程序对应的问题就能得到解决。因此, 要想解决问题, 首先要写出解决问题的正确步骤。

例如, 求一元二次方程  $ax^2+bx+c=0$  (设  $a \neq 0$ ) 实数根的步骤如下。

第一步 获得系数  $a, b, c$ 。

第二步 计算  $d = b^2 - 4ac$ 。

第三步 若  $d > 0$

计算:  $x_1 = (-b + \sqrt{d}) / (2a), x_2 = (-b - \sqrt{d}) / (2a)$

输出: 两个实根  $x_1$  和  $x_2$ , 转第六步。

第四步 若  $d < 0$

输出: 没有实根, 转第六步。

第五步 计算:  $x_1 = x_2 = (-b) / (2a)$

输出: 两个相同的实数根  $x_1$ , 转第六步。

第六步 结束。

上述步骤就是求一元二次方程实数根的程序。

程序设计 (Programming) 就是分析解决问题的方法和步骤, 并将其记录下来的过程。在描述问题求解步骤时, 就需要使用程序设计语言, 所谓程序设计语言是用于书写计算机程序的语言。语言的基础是一组记号和一组规则, C 语言就是一种很好的程序设计语言。程序设计过程一般包括分析、设计、编码、测试、排错等不同阶段。

## 1.2 C 语言的发展和特点

### 1.2.1 C 语言的发展

#### 1. C 语言产生的背景

以前操作系统及其他系统软件主要用汇编语言编写，由于汇编语言依赖于机器硬件，程序的可读性和可移植性都很差。为了提高可读性和可移植性，最好使用高级语言，可是一般的高级语言又难以实现汇编语言能直接对硬件进行操作的功能，所以人们设想能否找到一种既具有高级语言的特点、又具有低级语言特点的语言，集它们的优点于一身，于是，C 语言应运而生。

#### 2. C 语言的发展

C 语言是在 B 语言的基础上发展起来的，而 B 语言的产生是以 BCPL 为基础的。C 语言的发展过程如图 1.1 所示。

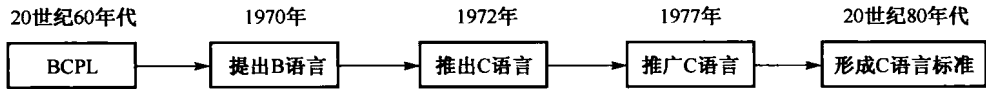


图 1.1 C 语言的发展过程

① BCPL：它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序。1963 年英国剑桥大学推出了 CPL（Combined Programming Language，复合编程语言）。CPL 在 ALGOL 60 的基础上接近了硬件一些，但规模比较大，难于实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 进行简化，推出了 BCPL（基本复合编程语言）。

② B 语言：1970 年美国贝尔实验室的 Ken Thompson 对 BCPL 进一步简化，设计出很简单而又很接近硬件的 B 语言，并用 B 语言写了第一个 UNIX 操作系统，在 PDP-7 计算机上实现。1971 年又在 PDP-11/20 计算机上实现了 B 语言并用它编写了 UNIX 操作系统。但 B 语言过于简单，功能有限。

③ 推出 C 语言：1972—1973 年，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。最初的 C 语言是一种描述和实现 UNIX 操作系统的工作语言。1973 年，K. Thompson 和 D. M. Ritchie 合作，把 UNIX 程序的 90% 以上用 C 语言改写，形成了 UNIX 第 5 版。直到 1975 年，UNIX 第 6 版发布后，C 语言的突出优点才引起人们的普遍注意。

④ 推广 C 语言：1977 年出现了不依赖于具体计算机的 C 语言编译文本——可移植 C 语言编译程序，使 C 程序移植到其他计算机时所需做的工作大大简化，这也推动了 UNIX 操作系统迅速在各种计算机上实现。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广。1978 年以后，C 语言先后移植到大、中、小、微型机上，已独立于 UNIX 和 PDP 了。现在 C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

目前广泛流行的各种版本的 C 语言编译系统虽然基本相同，但也有一些差别。在微机使用的 Microsoft C、Turbo C、Quick C、Borland C 等，它们的不同版本略有差异，因此读者需了解所用的计算机系统的 C 编译的特点和规定。本书中，用 Microsoft Visual C++ 6.0 作为 C 语言程序的编译程序。

⑤ C 语言的标准化工：C 语言的标准化工作是从 20 世纪 80 年代初期开始。1983 年，美

国家标准学会 (ANSI) 根据已有的各种 C 语言版本提出了对 C 语言的扩充和发展方案, 颁布了 C 语言的新标准 ANSI C。

### 1.2.2 C语言的特点

同其他程序设计语言相比, C 语言之所以能够存在和发展, 并具有很强的生命力, 是因为它有如下主要特点。

#### 1. 语言简洁、紧凑, 使用方便、灵活

C 语言一共只有 32 个关键字、9 种控制语句, 压缩了一切不必要的成分, 程序书写形式自由, 语句简洁。

#### 2. 运算符丰富, 适用的范围也很广泛

C 语言共有 34 种运算符, 它把括号、赋值符号、强制类型转换符号等都作为运算符处理, 从而使 C 语言的运算符类型极其丰富, 表达式类型多样化, 灵活使用各种运算符可以实现用其他高级语言难以实现的运算和操作。

#### 3. 数据结构丰富, 具有现代化语言的各种数据结构

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。这些丰富的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。尤其是指针类型的数据, 使用起来灵活多变。

#### 4. 具有结构化的控制语句

如 if...else 语句、switch 语句、while 语句、do...while 语句、for 语句等, 这些语句可以实现程序中所有的控制结构。另外, 函数是 C 语言程序的基本单位, 将函数作为程序模块的基本单元, 以实现程序的模块化。C 语言是结构化的理想语言。

#### 5. 编程限制少, 程序设计自由度大

一般的高级语言语法规则和检查比较严格, 几乎能检查出所有的语法错误。而 C 语言允许程序的编写有较大的自由度, 因此放宽了语法检查。编写者应当仔细检查程序, 保证其正确性, 而不要过分依赖编译软件去查错。“限制”和“灵活”是一对矛盾。限制严格, 就失去灵活性; 而强调灵活, 就必然放松限制。这一点使得 C 语言比其他语言对程序编写者的要求更高。例如, 对数组下标越界不进行检查, 由程序编写者自己保证程序的正确性。对变量类型的使用比较灵活, 例如, 整型数据与字符型数据及逻辑型数据可以通用, 使得某些运算变得更加简单、直接。

#### 6. 可直接对硬件操作

C 语言允许直接访问物理地址, 能进行位操作, 能实现汇编语言的大部分功能, 可以直接对硬件进行操作。这个特点使得 C 语言既具有高级语言的功能, 又具有许多低级语言的特点, 可以用来编写系统程序。

#### 7. 目标代码质量好, 程序执行效率高

C 语言生成的目标代码一般只比汇编语言生成的目标代码的效率低 10%~20%。

#### 8. 程序的可移植性好

与汇编语言相比, 用 C 语言编写的程序基本上不用修改就能用于各种型号的计算机和操作系统, 使程序具备了很好的移植性。

以上介绍的是 C 语言的一般特点，至于其内部的其他特点，将结合以后各章节内容逐一进行介绍。正是 C 语言的这些优点，使得它的应用非常广泛。许多大的软件都用 C 语言编写，这主要是由于 C 语言的可移植性好和对硬件的控制能力高，表达和运算能力强。许多以前只能用汇编语言处理的问题现在可以改用 C 语言来处理了。

总之，C 语言对编程者要求较高。由于使用 C 语言编写程序的限制很少、灵活性大、功能强，可以编写出任何类型的程序，所以学习和使用 C 语言的人越来越多。

### 1.3 C 语言的程序结构及特点

本节通过几个简单的 C 语言程序，认识与体会 C 语言程序的结构和特性。

**【例 1.1】** 输出一行字符串。

```
#include <stdio.h>          /*编译预处理*/
main()                      /*主函数*/
{
    printf("This is a C program.\n"); /*语句*/
}
```

此程序的功能是在屏幕上输出下面的一行信息：

This is a C program.

其中，main 是主函数，每一个 C 语言程序都必须有一个 main 函数，表示整个 C 语言程序的入口。函数体用大括号“{ }”括起来。本例中主函数内只有一个函数调用语句，printf 是 C 语言的输出函数，其功能是将双引号中的字符串原样输出。“\n”是换行控制符，即在“**This is a C program.**”输出之后回车换行。每一个语句后要有一个分号(;)，表示语句结束。

**【例 1.2】** 求两数之和。

```
main()
{
    int a,b,sum;          /*定义变量 a, b, sum*/
    a=123;                /*给变量 a 赋值, 123 为十进制常量*/
    b=0456;               /*给变量 b 赋值, 456 为八进制常量*/
    sum=a+b;              /*计算 a+b 的和, 并将结果赋给 sum 变量*/
    printf("sum is %d\n",sum); /*输出结果, 以十进制形式输出*/
}
```

本程序的功能是求两个整数的和，并将其输出。/\*.....\*/表示注释部分，其作用是提高程序的可读性。注释只是给人看的，对编译和运行不起作用。注释可以加在程序中的任何位置。printf 函数中的“sum is %d\n”是“输出格式字符串”，其中%d 是格式字符，用来指定输出时的数据类型和格式，“%d”表示“十进制整数类型”。在执行输出时，此位置代表一个十进制整型数值。printf 函数中最右端的 sum 是要输出的变量，现在它的值为 425，因此执行该程序的结果是输出如下一行信息：

sum is 425

**【例 1.3】** 输入 a, b 两个值，输出其中大者。

```
main()                      /*主函数*/
{
    int max();              /*函数声明*/
    int a,b,c;             /*定义变量*/
    scanf("%d,%d",&a,&b); /*输入两个整型数给变量 a 和 b*/
}
```

```

    c=max(a,b),          /*调用 max 函数, 将返回值赋给 c*/
    printf("max=%d",c); /*输出 c 的值*/
}
int max(x,y)           /*max 函数的定义, 函数返回值为整型, x,y 为形式参数*/
int x,y;              /*对形参 x,y 进行类型说明*/
{                      /*定义 max 中用到的变量 z*/
    int z;
    if (x>y) z=x;
    else z=y;
    return(z);        /*将 z 的值作为函数 max 返回值带回调用处*/
}

```

本程序由一个主函数 main 和一个被调用函数 max 组成。说明 C 语言源程序由函数构成, 包含一个主函数和若干个其他函数。函数可以是系统定义的库函数(如主函数中的 printf、scanf 函数, 只能调用, 无须定义), 也可以是用户自己定义的函数(如此例中的 max 函数)。

max 函数的功能是将 x 和 y 中的较大者赋给变量 z。return 语句将 z 的值返回给主函数 main。返回值通过函数名 max 带回到 main 函数的调用处。在 main 函数中, 调用了系统函数 scanf, 其作用是通过键盘输入 a 和 b 的值。“&”的含义是“取地址”, 表示将输入的值放到 a、b 所代表的地址单元中。

main 函数的第六行调用了 max 函数, 在调用时将实际参数 a 和 b 的值分别传送给 max 函数中的形式参数 x 和 y。经过执行 max 函数得到一个返回值, 把这个值赋给变量 c, 然后输出 c 的值。此程序的执行结果如下:

```

10,20✓           (输入 10, 20 并按回车键, 符号✓表示回车)
max=20          (输出 c 变量的值)

```

通过对以上几个例子的分析, 可以总结出 C 语言程序结构有如下特点。

① C 语言程序由函数构成。一个 C 语言程序至少要包括一个主函数, 即 main 函数, 可以包含一个 main 函数和若干个其他函数。因此, 函数是 C 语言程序的基本单位。被调用的函数可以是系统提供的库函数, 如 printf 和 scanf 函数, 也可以是用户自定义的函数。C 语言中的函数相当于其他语言中的子程序。C 语言用函数来实现特定的功能, C 语言的系统函数库十分丰富, 编译系统能够提供三百多个库函数。C 语言的这种特点易于实现程序的模块化。

② 每个函数都由两部分组成: 函数的说明部分和函数体。函数的一般形式为:

**函数类型 函数名 (形参表)**

**形参类型说明;**

{

**函数体;**

}

其中, 说明部分包括函数名、函数类型、形参表、形参类型说明。一个函数名后面必须跟一对圆括弧, 可以没有参数, 如 main(); 函数体即函数说明部分下面的大括弧“{ }”内的部分, 如果一个函数中有多对大括弧, 则最外层的一对大括弧为函数体的范围。函数体一般包括变量的定义部分和执行部分。

③ main 函数是整个 C 语言程序的入口。一个 C 语言程序总是从 main 函数开始执行, 也在 main 中结束, 其他函数通过调用得以执行。main 函数可以在程序最前面, 也可以在程序最后, 或在一些函数之前、另一些函数之后。

④ 程序书写格式自由, 一行内可以写几个语句, 一个语句也可以分开写有多行上。各

语句之间用分号 (;) 分隔。分号是 C 语句的必要组成部分。语句结束标志分号不可省略,即使是程序的最后一个语句,也必须有分号。注意,复合语句可以不用分号作为语句结束符。

⑤ C 语言本身没有输入/输出语句。其输入和输出功能是由库函数 `scanf` 和 `printf` 等来实现的,即 C 语言对输入/输出实行“函数化”。

⑥ 可以用 `/*……*/` 对 C 语言程序中的任何部分进行注释,以提高程序的可读性。

## 习 题 1

### 一、填空题

1. 一个 C 语言源程序中至少应包括一个\_\_\_\_\_。
2. 在一个 C 语言源程序中,注释部分两侧的分界符分别是\_\_\_\_\_和\_\_\_\_\_。
3. 一个 C 语言程序的执行从\_\_\_\_\_函数开始,到\_\_\_\_\_函数结束。
4. 在 C 语言程序中,输入操作是由库函数\_\_\_\_\_完成的,输出操作是由库函数\_\_\_\_\_完成的。

### 二、选择题

1. C 语言属于 ( )。  
A. 机器语言                      B. 低级语言                      C. 中级语言                      D. 高级语言
2. C 语言程序能够在不同的操作系统下运行,这说明 C 语言具有很好的 ( )。  
A. 适应性                          B. 移植性                          C. 兼容性                          D. 操作性
3. 一个 C 语言程序是由 ( ) 组成的。  
A. 一个主程序和若干子程序                      B. 函数  
C. 若干过程                          D. 若干子程序
4. C 语言规定,在一个源程序中, `main` 函数的位置 ( )。  
A. 必须在最开始                      B. 必须在系统调用的库函数的后面  
C. 可以任意                          D. 必须在最后
5. C 语言程序的执行,总是起始于 ( )。  
A. 程序中的第一条可执行语句                      B. 程序中的第一个函数  
C. `main` 函数                          D. 包含文件中的第一个函数
6. 以下叙述不正确的是 ( )。  
A. 一个 C 语言源程序可由一个或多个函数组成  
B. 一个 C 语言源程序必须包含一个 `main` 函数  
C. C 语言程序的基本组成单位是函数  
D. 在 C 语言程序中,注释说明只能位于一条语句的后面
7. 下面对 C 语言特点的描述,不正确的是 ( )。  
A. C 语言兼有高级语言和低级语言的双重特点  
B. C 语言既可以用来编写应用程序,又可以来编写系统软件  
C. C 语言的可移植性较差  
D. C 语言是一种结构式模块化程序设计语言
8. C 语言程序的注释 ( )。  
A. 由 `/*` 开头, `*/` 结尾  
B. 由 `/*` 开头, `/*` 结尾



- C. 由“//”开头  
 D. 由“/\*”或“//”开头
9. C语言程序的语句都以（ ）结尾。  
 A. “.”                      B. “;”                      C. “,”                      D. 都不是
10. 用C语言编写的代码程序（ ）。  
 A. 可立即执行                      B. 是一个源程序  
 C. 经过编译即可执行                      D. 经过编译解释才能执行

### 三、实验题

1. 编写一个简单的C语言程序完成如下功能，在屏幕上显示如下信息：

```
This is my first C program!
```

2. 上机调试例 1.1、例 1.2、例 1.3。

3. 上机调试程序，体会'\n'的作用，要求输出以下格式：

```
*****
Very good!
*****
```

程序清单如下：

```
main()
{   printf("*****\n");
    printf("Very good!\n");
    printf("*****\n");
}
```

4. 编写一个程序，输入 a、b、c 三个值，输出其中最大者。上机调试并执行。

程序清单如下：

```
main()
{
    int a,b,c,max;
    printf("请输入三个数 a, b, c:\n");
    scanf("%d,%d,%d",&a,&b,&c);
    max=a;
    if(max<b)   max=b;
    if(max<c)   max=c;
    printf("最大数为:%d",max);
}
```

5. 上机调试程序，体会 scanf 函数的作用，注意输入数据的格式。分析下列程序，从键盘输入两个字母并按回车键，写出其运行结果。

```
#include <stdio.h>
main()
{
    char c1,c2;
    scanf("%c%c",&c1,&c2);
    c1=c1+1;
    c2=c2-1;
    printf("c1=%c,c2=%c",c1,c2);
}
```

## 第 2 章 基本数据类型与运算

数据类型定义了一个对象（如常量和变量）应具有何种数值及对其可进行什么样的运算。常量和变量是 C 语言程序中最基本的数据处理对象，在使用变量前应先定义变量。变量定义包含了变量名、变量数据类型、变量的存储类别等内容。本章介绍前两个内容，变量的存储类别在第 7 章中介绍。运算是程序的基本功能，运算用运算符来描述，运算符决定运算对象应该进行何种运算。表达式就是用运算符将运算对象连接起来的式子，它的运算结果是一个值。

### 2.1 类型的重要性

在编写程序时必须要做好两件事：一是描述数据；二是描述数据的加工方法。前者通过数据定义语句实现，后者通过若干执行语句完成。数据的加工是建立在数据描述基础上的，数据描述的好坏直接影响数据加工的质量。例如，学生成绩处理程序中应该包含姓名、性别、年龄、计算机成绩、外语成绩和数学成绩等数据，其中姓名应该由多个字符组成，在 C 语言中称为字符串。性别可以由单个字符组成（如用 M 表示男性，F 表示女性），在 C 语言中称为字符型。字符串或字符型数据不能进行加、减等算术运算。年龄应该为整型，各门课程的成绩应该为实型，它们能进行算术运算。因此，对于一个具体问题，在程序设计中，先要解决的是分解步骤，确定运算，然后确定运算对象的类型。

### 2.2 C 语言的基本数据类型

在 C 语言中，任何数据的表现形式都有两种：常量或变量。无论常量还是变量，都必须有各自的数据类型。在一个具体的 C 语言系统中，每个数据类型都有固定的表示方式，如整型（int）、字符型（char）等。这种表示方式实际上确定了常量或变量可以表示的数据范围、在内存中的存放形式及所占内存空间的大小。

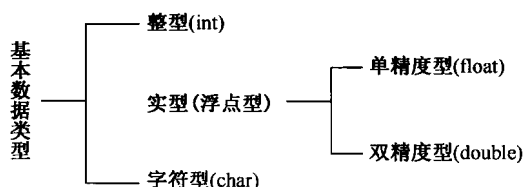


图 2.1 C 语言的基本数据类型

C 语言中有 4 种基本数据类型，每一种基本数据类型都用一个关键字表示，如图 2.1 所示。

其中，char（字符型）、int（整型）、float（单精度浮点型）和 double（双精度浮点型）都是关键字。在程序中用关键字表示对应的类型。

另外，还有 4 个修饰词可以出现在上述几个基本类型之前，从而改变原来的含义，它们是：short（短型）、long（长型）、signed（有符号）和 unsigned（无符号）。例如，short int 表示短整型；unsigned char 表示无符号字符型；long int 表示长整型；unsigned short int 表示无符号短整型。

在计算机中，由于机器字长的位数通常是固定的（如 16 位或 32 位），因此，计算机中数的表示范围（允许取值范围）是有限的。另外，实际的数有正数和负数。正号（+）或负号（-）在机器里用一位二进制数来表示，称为符号位。通常这个符号位放在二进制数的最

高位。一般规定 0 代表正号 (+), 1 代表负号 (-), 按该格式存放的数据称为符号数或机器数 (用 `signed` 修饰)。不设符号位的数据称为无符号数 (用 `unsigned` 修饰)。因此, 一个无符号变量只能存放不带符号的数据, 即无符号变量只能是一个大于等于零的数。尽管无符号变量存放数的范围大小和有符号数的范围大小一样, 但无符号变量无负数, 正数的范围比有符号数正数范围大了一倍多。例如, 整型数 (`int`) 在 TC (Turbo C) 系统中用 2 字节存放, 则 `[signed] int` 变量数的范围为  $-32\ 768 \sim 32\ 767$ 。而 `unsigned int` 变量数的范围为  $0 \sim 65\ 535$ 。同样, 整型数 (`int`) 在 VC 6.0 系统中用 4 字节存放, 则 `[signed] int` 变量数的范围为  $-2\ 147\ 483\ 648 \sim 2\ 147\ 483\ 647$ , 而 `unsigned int` 变量数的范围为  $0 \sim 4\ 294\ 967\ 295$ 。

修饰符 `long` 一般指存储空间比 `int` 型扩大了一倍, 而 `short` 一般指存储空间缩小到 `int` 型的一倍。但不同 C 语言编译系统的具体规定是不同的。例如, TC 中的 `int` 与 `short int` 所占用内存的位数一样 (都为 2 字节)。表 2.1 给出了 TC 系统和 VC 6.0 系统中的基本数据类型、字宽和范围。

表 2.1 基本数据类型、字宽和范围

类 型	类型名 (类型说明符)	所 占 位 数		VC 系统数的范围
		TC	VC	
字符型 (2 种)	<code>[signed] char</code>	8	8	$-128 \sim 127$
	<code>unsigned char</code>	8	8	$0 \sim 255$
整型 (6 种)	<code>[signed] int</code>	16	32	$-2\ 147\ 483\ 648 \sim 2\ 147\ 483\ 647$
	<code>[signed] short [int]</code>	16	16	$-32\ 768 \sim 32\ 767$
	<code>[signed] long [int]</code>	32	32	$-2\ 147\ 483\ 648 \sim 2\ 147\ 483\ 647$
	<code>unsigned [int]</code>	16	32	$0 \sim 4\ 294\ 967\ 295$
	<code>unsigned short [int]</code>	16	16	$0 \sim 65\ 535$
	<code>unsigned long [int]</code>	32	32	$0 \sim 4\ 294\ 967\ 295$
实型 (3 种)	<code>float</code>	32	32	可精确到小数点后 7 位有效数字
	<code>double</code>	64	64	可精确到小数点后 16 位有效数字
	<code>long double</code>	80	64	可精确到小数点后 16 位有效数字

表注: ① 在表中出现的 “[ ]”, 表示该部分可以省略。如 `int` 与 `signed int` 都是表示有符号整型变量的说明符; `long` 与 `long int` 都是表示有符号长整型变量的说明符。

② C 语言标准没有规定各类数据所占用的内存位数。所以不同 C 语言编译系统的各类数据所占用的内存位数是不同的。上机使用时, 应注意使用 C 语言编译系统的具体规定。

注意, 在后面的描述中都是针对 VC 6.0 系统来讲的。

## 2.3 标志符

程序中有许多需要命名的对象, 以便在程序的其他地方使用。那么, 如何表示在不同地方使用同一个对象呢? 最基本的方式就是为对象命名, 通过名字在程序中建立对象的定义与对象使用之间的关系。为此, 每种程序语言都规定了在程序中描述名字的规则, 这些名字包括: 变量名、常数名、数组名、函数名等, 通常被统称为标志符。

C 语言规定, 标志符由字母、数字和下画线 (`_`) 3 种字符组成, 它的第一个字符必须是字母或下画线。这里要说明的是, 为了标志符构造和阅读的方便, C 语言把下画线作为一个特殊字符使用, 它可以出现在标志符字符序列里的任何地方, 特别是它可以作为标志符的第一个字符出现。C 语言还规定, 标志符中, 同一个字母的大写与小写被看做是不同的字符。这样, `a` 和 `A`、`AB` 和 `Ab` 是互不相同的标志符。下面是合法的 C 语言标志符: