

# C语言程序设计教程

## (第2版)

易云飞 主 编  
万 励 唐凤仙 姜 林 副主编



高等学校计算机专业教材精选·算法与程序设计

# C语言程序设计教程

## (第2版)

易云飞 主 编  
万 励 唐凤仙 姜 林 副主编

清华大学出版社  
北 京

## 内 容 简 介

本书讲授 C 语言程序设计的基本思想、方法和解决实际问题的技巧,特点是尽可能把概念、知识点与案例相结合,进行案例教学,并且每一章都精心设计了习题与上机实验,突出实用性、操作性,每一章的安排尽量做到从简单到复杂,在编著教材时尽量结合任务驱动法和建构主义理论教学法组织各个知识点。

全书共分 13 章,主要内容包括 C 语言程序设计概述、数据类型、运算符、表达式、顺序结构、选择结构、循环结构、数组、函数、指针、复合数据类型、文件、位运算、编译预处理,以及 C 语言的实际应用等。

本书可作为高等学校各专业 C 语言程序设计课程的教材,也可以作为等级考试的辅导教材以及计算机爱好者自学用书和各类工程技术人员的参考书。

本书配有电子教案,并提供程序源代码,以方便读者自学。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

C 语言程序设计教程/易云飞主编.--2 版.--北京:清华大学出版社,2014

高等学校计算机专业教材精选·算法与程序设计

ISBN 978-7-302-35068-2

I. ①C… II. ①易… 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 006760 号

责任编辑:汪汉友

封面设计:傅瑞学

责任校对:李建庄

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:21.25 字 数:529 千字

版 次:2008 年 5 月第 1 版 2014 年 3 月第 2 版 印 次:2014 年 3 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:053555-01

# 出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行了大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并力图努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析问题、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,我们相信它能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

# 前 言

C 语言是一种在世界范围内被普遍使用的、优秀程序设计语言,是现代最流行的通用程序设计语言之一。它功能丰富,使用灵活,可移植性好,深受广大用户欢迎。C 语言的数据类型丰富,既具有高级程序设计语言的优点,又具有低级程序设计语言的特点;既可以用来编写系统程序,又可以用来编写应用程序。因此,C 语言正在被迅速地推广和普及。

本教材通过大量实例,系统地介绍了 C 语言的语法结构。全书共分 13 章,第 1 章为 C 语言程序设计概述,主要内容包括 C 语言发展的历史背景,C 语言的特点,C 语言程序的结构及在 Visual C++ 集成环境下运行 C 程序的步骤与方法。第 2 章为数据类型,运算符和表达式,主要内容包括 C 语言的基本数据类型的表示形式、存储格式、相关的运算以及各种表达式。第 3 章为顺序结构,主要内容包括 C 语句、数据的输入与输出及顺序程序设计。第 4 章为选择结构,主要内容包括关系运算符和关系表达式、逻辑运算符与逻辑表达式、选择结构语句及选择结构程序设计举例。第 5 章为循环结构,主要内容包括 while 型循环、do...while 型循环、for 型循环、循环的嵌套、其他控制语句及循环结构综合实例。第 6 章为数组,主要内容包括一维数组、二维数组、多维数组介绍、字符数组及数组综合实例。第 7 章为函数,主要内容包括函数概念引入、库函数、函数定义、函数间的参数传递、函数调用、数组与函数变量的作用域、变量的存储类型及函数的存储分类。第 8 章为指针,主要内容包括指针的含义、指针变量、指针与数组、指针与函数、指针与字符串、字符数组与字符指针的区别、指向指针的指针、指针类型小结及指针综合实例。第 9 章为复合数据类型,主要内容包括结构体、用结构指针操作链表、共用体、枚举类型及类型定义。第 10 章为文件,主要内容包括 C 文件的概述、文件的打开与关闭、文件的读写及文件的定位与出错检测。第 11 章为位运算,主要内容包括二进制位运算概述、位运算符及位段。第 12 章为编译预处理,主要内容包括文件包含、宏定义及条件编译。第 13 章为 C 语言的实际应用,主要讲解了两个实例:学生成绩管理系统和个人理财管理系统的设计与实现。

本教材文字精练,例题简单,容易理解,并配备了各种类型的练习,部分章节结合了任务驱动教学法和建构主义理论组织各知识点,便于学习掌握。对于 C 语言中重要和较难理解、容易出错的内容,书中均特别加以强调和说明。此外,在介绍 C 语言的语法结构的同时,也强调了计算机算法和结构化设计方法的概念和作用。本书可作为高等学校 C 语言程序设计课程的教材,也可作为计算机等级考试的辅导教材。

本书以 ANSI 标准 C 语言为背景,有关内容不依赖于任何具体的 C 系统。本书所有的程序实例都在 Visual C++ 6.0 中调试过,读者也可自由选用其他符合 ANSI 标准的 C 系统编程环境作为学习工具。

本书由易云飞担任主编,万励、唐凤仙和姜林任副主编。全书由易云飞组织、确定框架结构,并统编定稿。本书第 1 章和第 7 章由唐凤仙编写,第 2 章和第 13 章由姜林编写,第 3 章和第 9 章由黄华编写,第 4 章和第 12 章由万励编写,第 5 章和第 6 章由易云飞和罗富贵共同编写,第 8 章由李海英编写,第 10 章和第 11 章由喻飞编写。林晓东对全书的实例进行

了验证。林晓东、林叶川、梁国海、农光福、唐杰和吴勇兵完成了校正工作。另外,本书在编写过程中,得到武汉大学董文永教授、中南民族大学覃俊教授的指导和支持,得到兄弟高校从事计算机教育的老师的关心和帮助,教研室的同仁也提出了许多宝贵意见,在此一并表示衷心的感谢。

本书配有电子教案,并提供程序源代码,以方便读者自学,请到(网址由出版社定)下载。

限于作者水平,书中难免存在不当之处,恳请广大读者批评指正。任何批评和建议请发至:gxyiyf@163.com。

编 者  
2014年1月

# 目 录

|   |    |
|---|----|
| 第 1 章 C 语言程序设计概述 .....                  | 1  |
| 1.1 C 语言的发展及其特点 .....                   | 1  |
| 1.1.1 C 语言发展历程 .....                    | 1  |
| 1.1.2 C 程序的特点 .....                     | 2  |
| 1.1.3 C 语言与 C++ 语言交融发展 .....            | 3  |
| 1.2 C 语言程序的组成与结构 .....                  | 3  |
| 1.3 C 程序的上机步骤与方法 .....                  | 6  |
| 1.3.1 有关概念 .....                        | 6  |
| 1.3.2 运行 C 程序的一般步骤 .....                | 7  |
| 1.3.3 上机运行 C 程序的方法 .....                | 8  |
| 1.3.4 Visual C++ 6.0 环境运行 C 程序的步骤 ..... | 8  |
| 1.4 如何学好 C 语言 .....                     | 11 |
| 习题 .....                                | 11 |
| 第 2 章 数据类型,运算符和表达式 .....                | 13 |
| 2.1 C 语言字符集与标识符 .....                   | 13 |
| 2.1.1 C 语言字符集 .....                     | 13 |
| 2.1.2 标识符 .....                         | 14 |
| 2.2 常量和变量 .....                         | 15 |
| 2.2.1 常量 .....                          | 15 |
| 2.2.2 变量 .....                          | 16 |
| 2.3 基本数据类型 .....                        | 17 |
| 2.3.1 C 语言的数据类型 .....                   | 17 |
| 2.3.2 整型数据 .....                        | 17 |
| 2.3.3 实型数据 .....                        | 20 |
| 2.3.4 字符型数据 .....                       | 21 |
| 2.3.5 变量赋初值 .....                       | 23 |
| 2.4 运算符和表达式 .....                       | 23 |
| 2.4.1 运算符和表达式概述 .....                   | 23 |
| 2.4.2 算术运算符及算术表达式 .....                 | 25 |
| 2.4.3 关系运算符及关系表达式 .....                 | 28 |
| 2.4.4 逻辑运算符及逻辑表达式 .....                 | 29 |
| 2.4.5 条件运算符及条件表达式 .....                 | 31 |
| 2.4.6 赋值运算符及赋值表达式 .....                 | 33 |

|              |                          |           |
|--------------|--------------------------|-----------|
| 2.4.7        | 逗号运算符及逗号表达式 .....        | 35        |
| 2.4.8        | 求字节运算符 .....             | 36        |
| 2.4.9        | 类型转换 .....               | 36        |
| 习题           | .....                    | 37        |
| <b>第 3 章</b> | <b>顺序结构 .....</b>        | <b>40</b> |
| 3.1          | 概述 .....                 | 40        |
| 3.2          | 简单顺序语句 .....             | 40        |
| 3.2.1        | 表达式语句 .....              | 40        |
| 3.2.2        | 空语句 .....                | 40        |
| 3.2.3        | 复合语句 .....               | 41        |
| 3.3          | C 语言数据的输入输出 .....        | 41        |
| 3.3.1        | 字符输入输出函数 .....           | 42        |
| 3.3.2        | 格式输入输出 .....             | 43        |
| 3.4          | 综合实例 .....               | 51        |
| 习题           | .....                    | 53        |
| <b>第 4 章</b> | <b>选择结构 .....</b>        | <b>56</b> |
| 4.1          | if 语句 .....              | 56        |
| 4.1.1        | if 语句的基本形式 .....         | 56        |
| 4.1.2        | if 语句嵌套 .....            | 60        |
| 4.2          | switch 语句 .....          | 62        |
| 4.3          | 程序应用 .....               | 65        |
| 习题           | .....                    | 72        |
| <b>第 5 章</b> | <b>循环结构 .....</b>        | <b>75</b> |
| 5.1          | 问题的提出 .....              | 75        |
| 5.2          | while 循环结构 .....         | 76        |
| 5.2.1        | while 语句的基本形式 .....      | 76        |
| 5.2.2        | while 语句的应用 .....        | 76        |
| 5.3          | do...while 循环结构 .....    | 79        |
| 5.3.1        | do...while 语句的基本形式 ..... | 79        |
| 5.3.2        | do...while 语句的应用 .....   | 80        |
| 5.4          | for 循环结构 .....           | 82        |
| 5.4.1        | for 语句的基本形式 .....        | 82        |
| 5.4.2        | for 语句的应用 .....          | 82        |
| 5.5          | 转向语句 .....               | 86        |
| 5.5.1        | break 语句 .....           | 86        |
| 5.5.2        | continue 语句 .....        | 87        |



|              |                     |            |
|--------------|---------------------|------------|
| 5.5.3        | goto 语句 .....       | 88         |
| 5.6          | 程序应用 .....          | 89         |
|              | 习题 .....            | 93         |
| <b>第 6 章</b> | <b>数组 .....</b>     | <b>95</b>  |
| 6.1          | 问题的提出 .....         | 95         |
| 6.2          | 数组的基本概念 .....       | 95         |
| 6.3          | 一维数组 .....          | 96         |
| 6.3.1        | 一维数组的声明 .....       | 96         |
| 6.3.2        | 一维数组的引用 .....       | 97         |
| 6.3.3        | 一维数组的初始化 .....      | 98         |
| 6.3.4        | 一维数组的应用 .....       | 98         |
| 6.4          | 二维数组 .....          | 103        |
| 6.4.1        | 二维数组的声明 .....       | 103        |
| 6.4.2        | 二维数组的引用 .....       | 104        |
| 6.4.3        | 二维数组的初始化 .....      | 104        |
| 6.4.4        | 多维数组 .....          | 105        |
| 6.4.5        | 二维数组的应用 .....       | 105        |
| 6.5          | 字符数组 .....          | 109        |
| 6.5.1        | 字符数组的声明 .....       | 109        |
| 6.5.2        | 字符数组的引用 .....       | 109        |
| 6.5.3        | 字符数组的初始化 .....      | 110        |
| 6.5.4        | 字符串变量 .....         | 111        |
| 6.5.5        | 字符串变量的输入输出 .....    | 111        |
| 6.5.6        | 字符串函数 .....         | 113        |
| 6.5.7        | 字符数组的应用 .....       | 115        |
|              | 习题 .....            | 117        |
| <b>第 7 章</b> | <b>函数 .....</b>     | <b>120</b> |
| 7.1          | 概述 .....            | 120        |
| 7.2          | 函数定义的形式 .....       | 120        |
| 7.3          | 函数参数和函数的返回值 .....   | 122        |
| 7.3.1        | 形式参数和实际参数 .....     | 122        |
| 7.3.2        | 函数的返回值 .....        | 123        |
| 7.4          | 函数的调用 .....         | 125        |
| 7.4.1        | 函数调用的一般形式 .....     | 125        |
| 7.4.2        | 函数调用的方式 .....       | 127        |
| 7.4.3        | 对被调函数的声明和函数原型 ..... | 127        |
| 7.5          | 函数的嵌套调用 .....       | 130        |

|              |                        |            |
|--------------|------------------------|------------|
| 7.6          | 函数的递归调用 .....          | 131        |
| 7.7          | 数组作为函数参数 .....         | 136        |
| 7.7.1        | 数组元素作为函数实参 .....       | 136        |
| 7.7.2        | 一维数组名作为函数参数 .....      | 137        |
| 7.7.3        | 二维数组名作为函数参数 .....      | 140        |
| 7.8          | 变量的作用域 .....           | 141        |
| 7.8.1        | 局部变量 .....             | 141        |
| 7.8.2        | 全局变量 .....             | 142        |
| 7.8.3        | 变量的优先级 .....           | 144        |
| 7.9          | 变量的存储类别 .....          | 144        |
| 7.9.1        | auto 变量 .....          | 145        |
| 7.9.2        | 用 static 声明的局部变量 ..... | 145        |
| 7.9.3        | register 变量 .....      | 147        |
| 7.9.4        | 用 extern 声明外部变量 .....  | 148        |
| 7.9.5        | 用 static 声明外部变量 .....  | 151        |
| 7.9.6        | 存储类别小结 .....           | 151        |
| 7.10         | 内部函数和外部函数 .....        | 153        |
| 7.10.1       | 内部函数 .....             | 153        |
| 7.10.2       | 外部函数 .....             | 153        |
|              | 习题 .....               | 155        |
| <b>第 8 章</b> | <b>指针 .....</b>        | <b>158</b> |
| 8.1          | 数据、地址与指针 .....         | 158        |
| 8.1.1        | 数据存储与地址指向 .....        | 158        |
| 8.1.2        | 地址与指针 .....            | 159        |
| 8.2          | 指针变量 .....             | 160        |
| 8.2.1        | 指针变量的声明与赋值 .....       | 160        |
| 8.2.2        | 通过指针访问变量 .....         | 163        |
| 8.2.3        | 指针变量的应用 .....          | 164        |
| 8.3          | 指针与数组 .....            | 166        |
| 8.3.1        | 数组名与数组首地址 .....        | 166        |
| 8.3.2        | 指针的运算 .....            | 167        |
| 8.3.3        | 通过指针访问一维数组 .....       | 169        |
| 8.3.4        | 通过指针访问多维数组 .....       | 172        |
| 8.4          | 指针与字符串 .....           | 175        |
| 8.4.1        | 通过指针访问字符数组 .....       | 175        |
| 8.4.2        | 字符指针作为函数参数传递 .....     | 177        |
| 8.5          | 指针数组 .....             | 179        |
| 8.6          | 指针函数 .....             | 181        |

|               |                           |            |
|---------------|---------------------------|------------|
| 8.7           | 函数指针 .....                | 182        |
| 8.8           | 多级指针 .....                | 184        |
| 8.9           | void 指针类型 .....           | 185        |
| 8.9.1         | void 的含义 .....            | 185        |
| 8.9.2         | void 使用规则 .....           | 186        |
| 8.10          | 带参数的 main() 函数 .....      | 189        |
| 8.11          | 指针的内存动态分配 .....           | 190        |
| 8.12          | 指针的应用举例 .....             | 193        |
|               | 习题 .....                  | 196        |
| <b>第 9 章</b>  | <b>复合数据类型 .....</b>       | <b>200</b> |
| 9.1           | 结构体数据类型 .....             | 200        |
| 9.1.1         | 结构体变量的使用 .....            | 201        |
| 9.1.2         | 结构体数组 .....               | 205        |
| 9.1.3         | 结构体指针 .....               | 208        |
| 9.1.4         | 结构体及指向结构体的指针作为函数的参数 ..... | 210        |
| 9.1.5         | 结构体综合举例 .....             | 211        |
| 9.2           | 链表 .....                  | 213        |
| 9.2.1         | 链表与数组的主要区别 .....          | 213        |
| 9.2.2         | 链表的操作 .....               | 213        |
| 9.2.3         | 链表应用举例 .....              | 216        |
| 9.3           | 共用体定义和共用体变量声明 .....       | 217        |
| 9.3.1         | 共用体定义 .....               | 217        |
| 9.3.2         | 共用体类型变量 .....             | 218        |
| 9.3.3         | 共用体类型变量的引用 .....          | 218        |
| 9.4           | 枚举数据类型 .....              | 220        |
| 9.4.1         | 枚举类型的定义和枚举变量的说明 .....     | 220        |
| 9.4.2         | 枚举类型变量的赋值和使用 .....        | 220        |
| 9.5           | 类型说明 .....                | 222        |
|               | 习题 .....                  | 223        |
| <b>第 10 章</b> | <b>文件 .....</b>           | <b>228</b> |
| 10.1          | 文件概述 .....                | 228        |
| 10.2          | FILE 结构类型 .....           | 228        |
| 10.3          | 文件的操作 .....               | 229        |
| 10.3.1        | 文件的打开 .....               | 229        |
| 10.3.2        | 文件的关闭 .....               | 231        |
| 10.3.3        | 文件的读写 .....               | 231        |
| 10.3.4        | 文件缓冲区操作 .....             | 237        |

|               |                  |     |
|---------------|------------------|-----|
| 10.3.5        | 文件的随机读写          | 238 |
| 10.3.6        | 文件的监测            | 242 |
| 10.4          | C库文件             | 243 |
|               | 习题               | 245 |
| <b>第 11 章</b> | <b>位运算</b>       | 246 |
| 11.1          | 整数的计算机表示         | 246 |
| 11.2          | 位运算符             | 248 |
| 11.2.1        | 取反运算符 $\sim$     | 248 |
| 11.2.2        | 按位与运算符 $\&$      | 249 |
| 11.2.3        | 按位或运算符 $ $       | 249 |
| 11.2.4        | 按位异或运算符 $\wedge$ | 250 |
| 11.2.5        | 左移运算符 $\ll$      | 251 |
| 11.2.6        | 右移运算符 $\gg$      | 251 |
| 11.2.7        | 位运算与赋值运算的结合      | 251 |
| 11.2.8        | 位运算举例            | 251 |
|               | 习题               | 252 |
| <b>第 12 章</b> | <b>编译预处理</b>     | 254 |
| 12.1          | 宏声明              | 254 |
| 12.1.1        | 不带参数的宏声明         | 254 |
| 12.1.2        | 带参数的宏声明          | 256 |
| 12.2          | 文件包含             | 262 |
| 12.3          | 条件编译             | 265 |
|               | 习题               | 269 |
| <b>第 13 章</b> | <b>C 语言的实际应用</b> | 271 |
| 13.1          | C 语言程序设计步骤       | 271 |
| 13.2          | 实际应用中常见模块设计      | 272 |
| 13.2.1        | 输入模块设计           | 272 |
| 13.2.2        | 选择菜单模块设计         | 275 |
| 13.2.3        | 数据结构设计           | 277 |
| 13.2.4        | 功能模块设计           | 277 |
| 13.3          | 实际应用举例           | 280 |
| 13.3.1        | 学生成绩管理           | 280 |
| 13.3.2        | 个人理财管理           | 295 |

|      |                            |     |
|------|----------------------------|-----|
| 附录 A | ASCII 编码表 .....            | 319 |
| 附录 B | ctype. h 文件中所包含的字符函数 ..... | 321 |
| 附录 C | math. h 文件中所包含的数学函数 .....  | 322 |
| 附录 D | C 语言运算符优先级详细列表 .....       | 324 |

# 第 1 章 C 语言程序设计概述

C 语言是一种通用的程序设计语言,由于它很适合用来编写编译器和操作系统,因此称为“系统编程语言”,它既具有一般高级语言特性,又具有低级语言特性。通过本章学习,了解 C 语言发展的历史背景,C 语言的特点,重点掌握 C 语言程序的结构和在 Visual C++ 集成环境下运行 C 程序的步骤与方法。

## 1.1 C 语言的发展及其特点

### 1.1.1 C 语言发展历程

C 语言在 1983 年进行了标准化工作,颁布了第一份标准版本,自此以后,C 语言就在编程语言中占据着龙头的位置。

C 语言的发展过程经历了:ALGOL 60→CPL→BCPL→B→C→标准 C→ANSI C→ISO C。

ALGOL 60:是 1960 年出现的一种面向问题的高级语言。ALGOL 60 离硬件较远,不适合用来编写系统程序。

CPL(Combined Programming language,组合编程语言):1963 年英国的剑桥大学推出了 CPL,CPL 是在 ALGOL 60 基础上更接近于硬件的一种语言。但 CPL 规模大,实现困难。

BCPL(Basic Combined Programming language,基本的组合编程语言):1967 年英国剑桥大学的 Martin Richards 对 CPL 语言做了简化,推出了 BCPL 语言。

B 语言:1970 年,UNIX 的开山鼻祖美国贝尔实验室的 Ken Thompson 设计出了既简单又很接近硬件的 B 语言(取 BCPL 的第一个字母),BCPL(Basic Combined Programming language,基本的组合编程语言),B 语言诞生后,UNIX 开始用 B 语言改写(原来的 UNIX 操作系统是 1969 年由美国的贝尔实验室的 K. Thompson 和 D. M. Ritchie 开发成功的,是用汇编语言写的)。

C 语言:1972 年,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)。C 语言既保持了 BCPL 和 B 语言的优点(精练、接近硬件),又克服了它们的缺点(过于简单、数据无类型等)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。

标准 C:以 1978 年发表的 UNIX 第 7 版中的 C 编译程序为基础,Brian W. Kernighan 和 Dennis M. Ritchie(合称 K&R)合著了影响深远的名著《The C Programming Language》,这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础,它被称为标准 C。

ANSI C:1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充,制定了新的标准,称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展。

K & R 在 1988 年修改了他们的经典著作《The C Programming Language》，按照 ANSI C 标准重新写了该书。1987 年，ANSI 又公布了新标准——87 ANSI C。

ISO C: 1990 年，国际标准化组织 ISO(International Standard Organization)接受 87 ANSI C 为 ISO C 的标准(ISO 9899—1990)。目前流行的 C 编译系统都是以它为基础的。

1999 年推出的 C99 在保留 C 语言特征的基础上，增加了面向对象的新特征，使 C 语言从面向过程的语言发展成为面向对象的语言。C 语言是 C++ 的基础，掌握了 C 语言，再学习 C++，将会驾轻就熟。

目前在微型计算机上常用的有 Microsoft C、Turbo C、Quick C、BORLAND C 等多种版本。这些不同版本的 C 编译系统所实现的语言功能和语法规则基本相同，但在某些方面又略有差异，因此读者应了解所用的 C 语言编译系统的特点(可以参阅有关手册)。本书基本上以 ANSI C 为基础来讲授，上机环境使用在 Windows 操作系统上方便使用的 Visual C++ 6.0 可视化集成开发工具。

### 1.1.2 C 程序的特点

C 语言发展迅速，成为最受欢迎的高级语言之一，主要因为它功能强大，与其他早期语言相比，它具有一系列优点。

#### 1. C 语言简洁，使用灵活，便于学习和应用

C 语言只有 32 个关键字，9 种控制语句，书写形式直观、精炼。

#### 2. 运算符丰富

除了最基本的加、减、乘、除、取模等运算外，C 把括号、赋值、强制类型转换等都作为运算符处理，共有 34 种运算符。从而使 C 的运算类型极其丰富，表达式类型多样化。

#### 3. 数据类型丰富

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。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，使程序效率更高。

#### 4. C 语言是结构化语言

具有结构化的流程控制语句(如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句)。用函数作为程序的模块单位，便于实现程序的模块化。

#### 5. 可直接与机器硬件打交道，直接访问内存地址

C 语言允许直接访问内存地址，能进行位操作，能实现汇编语言的大部分功能，兼有高级和低级语言的特点。C 语言的这种双重性，使它既是成功的系统描述语言(如编写 UNIX 操作系统)，又是通用的程序设计语言。

#### 6. 运行程序质量高，程序执行效率高

用 C 语言编写的程序经编译生成的目标代码效率接近于汇编语言程序，只比汇编程序生成的目标代码效率低 10%~20%，可以开发出执行速度很快的程序。

#### 7. C 语言适用范围广，可移植性好

C 编译程序 80%以上的代码是公共的，因此稍加修改就能移植到各种不同型号的计算机上。

当然，C 语言也有自身的不足，和其他高级语言相比，其语法限制不太严格，例如：对变

量的类型约束不严格,影响程序的安全性,对数组下标越界不作检查;编程自由度大,编译程序查错能力有限,对不熟练的程序员带来一定困难。

### 1.1.3 C语言与C++语言交融发展

由于C语言是面向过程的结构化和模块化的程序设计语言,当处理的问题比较复杂、规模庞大时,就显现出一些不足,由此面向对象的程序设计语言C++应运而生。C++的基础是C,它保留了C的所有优点,增加了面向对象机制,并且与之完全兼容。绝大多数C语言程序设计可以不加修改直接在C++环境中运行。

## 1.2 C语言程序的组成与结构

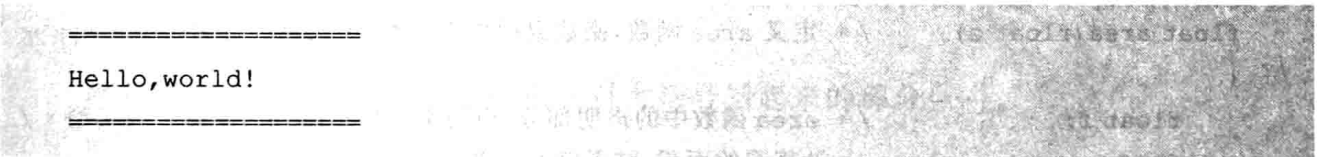
用C语言语句编写的程序称为C程序或C源程序。下面通过介绍3个简单的C语言程序例子来介绍C语言程序的基本组成与结构。

**例 1-1** 一个简单的输出程序。

程序代码如下:

```
#include<stdio.h> /* 文件包含,标准输入输出头文件 */
void main() /* 主函数 */
{ /* 函数体开始 */
    printf ("=====\n"); /* 输出语句 */
    printf ("Hello,world!\n"); /* 输出语句 */
    printf ("=====\n"); /* 输出语句 */
}
```

运行结果:



说明:

- (1) main 是主函数的函数名,每一个C程序都必须有且仅有一个main函数。
- (2) void 是函数的返回值类型,该例无返回值,用void表示。
- (3) {}是函数开始和结束的标志,不可省略,{}之间为函数体部分,该例有3行。
- (4) printf()是C语言中的输出函数,双引号内的字符串原样输出,\n是换行符。
- (5) 每个C语句以分号结束。
- (6) 使用标准库函数时应在程序开头一行写: #include<stdio.h>。
- (7) /\* ... \*/表示注释。注释内容只是给别人容易看懂,对编译和运行不起作用。所以可以用汉字或英文字符,可以出现在一行中的最右侧,也可以单独成为一行。

**例 1-2** 已知3个整型数4、8、10,按公式  $s=(a+b) * c$  计算,并输出s的值。

程序代码如下:

```
#include<stdio.h> /* 标准输入输出头文件 */
```



```

void main()
{
    int a,b,c,s;          /* 声明 4 个整型变量 */
    a=4;
    b=8;
    c=10;                /* 变量赋初值 */
    s=(a+b)*c;          /* 算术运算并赋值 */
    printf("s=%d\n",s); /* 输出结果 s 的值 */
}

```

运行结果：

```
s=120
```

**例 1-3** 包含自定义函数的 C 程序,输入圆的半径,求圆的面积。

程序代码如下：

```

#include<stdio.h>      /* 编译预处理 */
#define PI 3.1416     /* 声明符号常量 PI */
void main()           /* 主函数 */
{
    float area(float r); /* 对被调用函数 area 的声明 */
    float s,r;          /* 声明实型变量 s,r */
    printf("r=");      /* 输出提示信息 r= */
    scanf("%f",&r);    /* 输入变量 r 的值 */
    s=area(r);         /* 调用 area 函数,将得到的返回值赋给 s */
    printf("面积 s=%f\n",s); /* 输出 s 的值 */
}
float area(float a)   /* 定义 area 函数,函数返回值为实型,形式参数 a 为实型 */
{
    float f;          /* area 函数中的声明部分,声明本函数中用到的变量 f 为实型 */
    f=PI*a*a;        /* 计算圆的面积,结果赋给 f 变量 */
    return (f);      /* 将 f 的值返回给主调用函数 */
}

```

运行结果：

```
r=10 ✓
面积 s=314.160000
```

说明：本程序包括 main 函数和被调用函数 area 两个函数。area 函数的作用是求圆的面积并赋给 f。return 语句将 f 的值返回给主调用函数 main。area 函数是用户自定义的函数,在主函数中对被调用函数 area 进行声明(称为函数原型声明)。

### 1. 结构

程序中 scanf 函数的作用是输入 r 的值。&r 中的 & 的含义是“取地址”,此 scanf 函数的作用是将数值输入到变量 r 的地址所标志的单元中。