



高等职业教育“十二五”规划教材

C语言程序设计

项目教程

郭运宏 李玉梅 ○ 主 编
谢文昌 马国峰 蔡小磊 ○ 副主编

- 程序源代码
- PPT教学课件
- 配备习题答案
- 任务驱动、项目载体
- 理论与实践紧密结合

配套资源下载地址: <http://www.tup.com.cn>

清华大学出版社



高等职业教育“十二五”规划教材

C 语言程序设计项目教程

郭运宏 李玉梅 主编

谢文昌 马国峰 蔡小磊 副主编

清华大学出版社

北 京

内 容 简 介

本书以 C 语言基础知识→核心技术→高级应用为主线,以项目为背景,采取任务驱动的方法来组织编写,全书深入浅出地讲解了 C 语言的各项技术,并以大量的实例来加深读者对知识的理解和运用。在编写的过程中,还特别注重知识的层次性和技能的渐进性。

本书共分 4 篇,第 1 篇为基础篇,包括第 1~5 章,以实用计算器项目为背景,系统介绍 C 语言的基本知识和程序控制结构;第 2 篇为提高篇,包括第 6~8 章,以学生成绩统计项目为背景,重点介绍 C 语言的函数、数组和指针等核心技术;第 3 篇为应用篇,包括第 9~10 章,以学生信息管理系统为背景,介绍结构体、文件等应用;第 4 篇为高级篇,包括第 11~12 章,重点介绍位运算、编译预处理等深层次的知识。本书所有的程序代码均在 VC++6.0 环境中调试通过。

本书适合作为高等职业院校、普通高等院校计算机及相关专业教材,也可作为程序开发人员和爱好者自学的参考用书。

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 语言程序设计项目教程/郭运宏,李玉梅主编. —北京:清华大学出版社,2012.8
高等职业教育“十二五”规划教材

ISBN 978-7-302-29286-9

I. ①C… II. ①郭… ②李… III. ①C 语言-程序设计-高等职业教育-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 152928 号

责任编辑:杜长清

封面设计:刘 超

版式设计:文森时代

责任校对:赵丽杰

责任印制:李红英

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

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:23 字 数:531 千字

版 次:2012 年 9 月第 1 版 印 次:2012 年 9 月第 1 次印刷

印 数:1~5000

定 价:39.00 元

前 言

目前,很多高校都选用 C 语言作为程序设计课程的学习语言,然而在教学实践中却发现传统的 C 语言教材比较注重知识的体系结构,并不能很好地将知识、技能与实际软件开发结合起来,学习起来难度较大,学生的学习积极性和主动性不能得到充分发挥。本书本着“任务驱动、项目载体”的教学原则,由长期从事 C 语言教学的老师精心编写,来解决这个问题。

1. 本书主要特色

(1) 理念先进

本书紧紧围绕培养高技能人才的目标,以项目为背景,以知识为主线,学、用结合,大胆进行“校企合作、工学结合、项目导向、任务驱动”的教学改革,选取合适的项目作为学习载体,分别以实用计算器、学生成绩统计和学生信息管理系统 3 个项目为背景,以项目的开发过程为主线,并将每个项目分解成多个任务,合理地安排到相关章节中,将知识的讲解贯穿于项目的开发过程中,通过对任务的分析和实现,引导学生由浅入深、由简到难地学习,使学生的编程能力在 3 个项目的实施中逐步得到提高,达到学以致用目的。

(2) 组织合理

本书按照工作过程系统化的思想组织内容,把基础知识和扩展知识结合,保证知识的覆盖面,在完成项目的过程中贯穿了 C 语言基本语法、函数、程序设计方法、数据类型、数组、指针、结构体、文件等知识点。

在课程内容的选择上,遵循学生能力培养的基本规律,以 3 个项目作为教学载体,通过学习情境的构建将传统的教学内容进行解构、重构,并将 C 语言程序设计知识、软件工程基础知识、数据结构基本知识融入到项目开发的过程中。

本书将函数的知识放在第一个学习情境中讲解,后面所有的学习情境都使用函数进行编码,强化结构化程序设计思想,并根据需要把软件工程中的开发过程、模块要求、测试技术以及数据结构中的基本知识与常用算法重构到 3 个项目中。随着项目的进展,知识由易到难,能力的培养由窄到宽,课程内容和项目开发内容相一致,理论与实践一体化,提高学生的编程能力和综合能力,为可持续发展奠定良好的基础。

(3) 通俗易懂

本书作为编程语言的入门教材,侧重于编程思想、编码规范的培养,提供大量的实例和各种类型的习题,增强动手能力,内容由浅入深逐步展开,力求通俗易懂。

编程是一门注重实践的技术,实践性的技术要在实践中提高。本书采用一种简单的、易于接受的风格,重点讲解结构化方法的编程思想、编程技巧、调试技巧,培养学生养成良好的编程风格。书中精心设计了大量的例题,对每个程序都进行了细致的解析,总结了

各种编程方法；采用示例法教学，根据示例编写每章的实验题目和习题，读者参照示例可以轻松完成，事半功倍，并可举一反三。

本书在培养学生编程能力的同时，注重对学生进行编程规范的训练，使学生养成良好的编程习惯和方法，遵守基本的编程约定，在编程规范方面实现与软件企业的无缝对接。本书提供的实例，列举了处理该类题目时容易出现的问题，有些实例还给出了不同的解决方法，以便学生更好地了解和掌握程序开发的灵活性。同时，每章节中和章后均附有各种类型的习题，便于读者自查学习效果。本书中的代码均在 VC++6.0 环境中调试通过。

2. 本书内容安排

（1）基础篇

基础篇包括第 1~5 章。以实用计算器项目为背景，主要介绍 C 语言的基本知识以及顺序、分支和循环 3 种程序控制结构。通过本篇的学习，读者应能利用 C 语言基础知识编写简单的 C 程序。

第 1 章介绍 C 语言的发展及特点、C 程序的基本结构及使用 VC++6.0 开发 C 语言程序的过程。第 2 章介绍 C 语言的基本数据类型、常量和变量、运算符和表达式、不同数据类型间的转换方法。第 3 章介绍输入/输出函数、算法和顺序结构程序设计。第 4 章介绍选择结构程序设计方法。第 5 章介绍循环结构程序设计方法。

（2）提高篇

提高篇包括第 6~8 章。以学生成绩统计项目为背景，主要介绍函数、数组和指针的内容。通过本篇的学习，读者应能灵活运用函数、数组和指针编写程序，解决科学计算和工程设计中的一般性问题。

第 6 章介绍 C 语言函数的定义和调用、函数间的数据传递、变量的作用域和存储类型、函数的嵌套和递归调用、编译预处理等内容。第 7 章介绍一维数组、字符数组和二维数组的概念、定义和使用方法。第 8 章介绍指针的基本概念、指针与数组、指针与字符串、指针变量作函数参数等内容。

（3）应用篇

应用篇包括第 9~10 章。以学生信息管理系统为背景，主要介绍结构体和文件的内容。通过本篇的学习，读者应具有利用 C 语言进行软件设计的能力。

第 9 章介绍结构体和共用体的概念、结构体数组、结构体指针的使用方法。第 10 章介绍文件的基本知识和文件操作方法。

（4）高级篇

高级篇包括第 11~12 章。主要介绍位运算和编译预处理的内容。通过本篇的学习，读者应进一步提高使用 C 语言的能力。

第 11 章介绍运算符、表达式和位运算的使用方法。第 12 章介绍宏定义、文件包含和条件编译等内容和方法。

本书由郑州铁路职业技术学院郭运宏、李玉梅担任主编，谢文昌、马国峰、蔡小磊担任副主编，全书由郭运宏统稿。

本书的出版得到了清华大学出版社的大力支持，在此表示衷心的感谢。本书在编写过程中，还得到了郑州大学王瑞民教授和复旦大学杨青骢博士的大力支持，在此一并表示感谢。由于水平和时间有限，书中难免有疏漏和不足之处，恳请读者批评指正。

编 者



目 录

第 1 篇 基 础 篇

实用计算器项目概述	2
第 1 章 C 语言概述	4
1.1 C 语言的发展及特点	5
1.1.1 程序设计语言	5
1.1.2 C 语言的发展	6
1.1.3 C 语言的特点	6
1.2 简单的 C 语言程序介绍	7
1.3 C 语言程序的开发过程	9
1.4 VC++6.0 集成开发环境	10
1.5 Turbo C 开发环境	12
1.6 GCC 介绍	14
1.7 本章小结	16
1.8 习题	16
第 2 章 项目中的数据类型和数据运算	18
2.1 任务一 项目中数据类型的定义	19
2.2 理论知识	20
2.2.1 C 语言的数据类型	20
2.2.2 常量和变量	21
2.2.3 运算符和表达式	28
2.2.4 数据类型转换	33
2.3 知识扩展	34
2.3.1 数值在计算机中的表示	34
2.3.2 标准数学函数的使用	36
2.4 本章小结	36
2.5 习题	37
第 3 章 项目主菜单的顺序执行设计	39
3.1 任务二 用输入/输出函数实现项目主菜单的顺序执行	40
3.2 理论知识	42
3.2.1 C 语言程序语句	42

3.2.2	格式化输入/输出函数	43
3.2.3	单字符输入/输出函数	48
3.2.4	算法与程序的 3 种基本结构	49
3.2.5	顺序结构程序设计	53
3.3	知识扩展	54
3.3.1	程序设计的步骤	54
3.3.2	结构化程序设计的标准	55
3.3.3	程序设计的风格	55
3.4	本章小结	56
3.5	习题	56
第 4 章	项目主菜单的选择执行设计	60
4.1	任务三 项目主菜单的选择执行设计	61
4.2	理论知识	63
4.2.1	if 语句	63
4.2.2	switch 语句	67
4.3	选择控制结构程序举例	70
4.4	本章小结	74
4.5	习题	74
第 5 章	项目主菜单的循环执行设计	79
5.1	任务四 项目主菜单的循环执行设计	80
5.2	理论知识	82
5.2.1	while 语句	82
5.2.2	do...while 语句	84
5.2.3	for 语句	86
5.3	知识扩展	89
5.3.1	循环的嵌套	89
5.3.2	辅助控制语句	90
5.4	循环控制结构程序举例	94
5.5	本章小结	97
5.6	习题	98

第 2 篇 提 高 篇

学生成绩统计项目概述	106
第 6 章 项目的整体框架设计	110
6.1 任务一 项目的整体框架设计	111

6.2	理论知识	114
6.2.1	结构化程序设计思想与函数的分类	114
6.2.2	函数的定义	115
6.2.3	函数的值	116
6.2.4	函数的调用	118
6.2.5	函数间的数据传递	121
6.2.6	变量的作用域	122
6.3	知识扩展	126
6.3.1	变量的存储类型	126
6.3.2	函数的嵌套调用和递归调用	128
6.3.3	编译预处理	131
6.4	本章小结	138
6.5	习题	139
第7章	项目中数组的应用	144
7.1	任务二 用一维数组实现项目中学生成绩的统计	145
7.2	理论知识	148
7.2.1	一维数组	148
7.2.2	一维数组作函数参数	156
7.3	任务三 用字符数组实现项目中的密码验证	159
7.4	理论知识——字符数组	161
7.5	知识扩展——二维数组	167
7.6	应用举例	171
7.7	本章小结	173
7.8	习题	173
第8章	项目中指针的应用	178
8.1	任务四 用指针实现项目中学生成绩的统计	179
8.2	理论知识	182
8.2.1	指针的概念	182
8.2.2	指针变量的定义与引用	183
8.2.3	变量的指针与指针变量	189
8.2.4	指针与一维数组	190
8.2.5	指针变量作函数参数	194
8.2.6	指针与字符串	198
8.3	知识扩展	201
8.3.1	指针与二维数组	201
8.3.2	指针数组和指向指针的指针	204



8.3.3 带参数的 main() 函数	208
8.3.4 返回指针值的函数	209
8.4 应用举例	211
8.5 本章小结	213
8.6 习题	214

第 3 篇 应用篇

学生信息管理系统项目概述	218
第 9 章 项目中结构体的应用	223
9.1 任务一 用结构体实现项目中学生信息的增加、浏览和修改	224
9.2 理论知识	227
9.2.1 结构体类型的定义	227
9.2.2 结构体变量的定义	230
9.2.3 结构体变量的引用	233
9.2.4 结构体数组	235
9.2.5 结构体指针	238
9.2.6 结构体类型的数据在函数间的传递	241
9.3 知识扩展——共用体	243
9.3.1 共用体类型和共用体变量的定义	243
9.3.2 共用体变量的引用	244
9.4 知识扩展——枚举类型和自定义类型	247
9.4.1 枚举类型	247
9.4.2 自定义类型	249
9.5 本章小结	251
9.6 习题	251
第 10 章 项目中文件的应用	254
10.1 任务二 项目中数据的存储	255
10.2 理论知识	257
10.2.1 文件的基本概念	257
10.2.2 文件的打开和关闭	259
10.2.3 文件的顺序读写	262
10.2.4 文件的定位及随机读写	269
10.2.5 文件的出错检测	271
10.3 本章小结	272
10.4 习题	272

第4篇 高级篇

第 11 章 运算符、表达式与位运算.....	276
11.1 运算符与表达式.....	277
11.2 赋值运算符与赋值表达式.....	278
11.3 逗号运算符与逗号表达式.....	281
11.4 条件运算符与条件表达式.....	283
11.5 自增/自减运算符.....	285
11.6 位运算.....	288
11.6.1 位运算概述.....	288
11.6.2 位段.....	295
11.7 表达式求解示例.....	297
11.8 本章小结.....	299
11.9 习题.....	299
第 12 章 编译预处理.....	304
12.1 宏定义.....	305
12.2 文件包含.....	309
12.3 条件编译.....	310
12.4 本章小结.....	312
12.5 习题.....	313
附录 I 常用字符与 ASCII 码对照表.....	317
附录 II C 语言关键字.....	318
附录 III 运算符的优先级和结合方向.....	319
附录 IV 常用 C 库函数.....	320
附录 V 学生信息管理系统源程序代码.....	324
附录 VI 常见编译错误信息.....	338
附录 VII 用 C 语言编写一个学生数据库系统.....	347
参考文献.....	353



第 1 篇 基础篇

本篇以实用计算器项目为背景，介绍 C 语言中的数据与运算、程序控制结构等编程基本要素。为了便于相关理论知识的学习，将项目分解为 4 个子任务，分别贯穿于第 2~5 章中进行分析和实现。

通过本篇的学习，读者应了解 C 语言的特点和基本编程方法，并能利用 C 语言基础知识编写简单的程序，解决日常生活和工作中的小问题。

实用计算器项目概述

一、任务描述

使学生理解和掌握项目所涉及的知识要点内容，培养学生编程的逻辑思维能力，初步掌握利用 C 语言进行软件开发的基本方法和步骤。

二、知识要点

该项目涉及的知识要点包括：C 语言的基本数据类型、常量和变量、运算符和表达式、输入/输出函数、顺序结构程序设计、选择结构程序设计和循环结构程序设计。这些知识内容将在第 2~5 章进行详细介绍。

三、任务分析

实现一个实用计算器，能够完成整型数据和实型数据的加、减、乘、除四则运算。

为方便用户使用，要求采用人机对话形式，首先提供系统操作主菜单，给出加、减、乘、除和退出 5 个选项，当用户选择某一选项后（退出选项除外），系统提示输入第一个运算数和第二个运算数，并给出运算结果。然后询问是否继续计算，如果输入字母 Y 或 y，重新返回主菜单；如果输入其他字母，则结束计算并退出系统。另外，为了使用方便，在主菜单中特设 0 选项，选择该选项也能正常退出系统。

四、界面设计

图 0-1 给出了从主菜单中选择加法运算后的运行界面，其他 3 种运算的运行界面与此基本相同，不再给出。

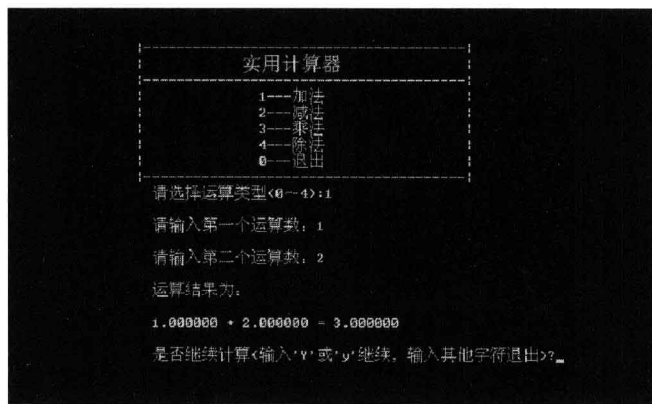


图 0-1 简易计算器运行界面

五、任务分解

为了便于相关理论知识的学习，将项目分解为4个子任务，每个子任务及其对应的章节如下。

- ☑ 第2章：任务一 项目中数据类型的定义
- ☑ 第3章：任务二 用输入/输出函数实现项目主菜单的顺序执行
- ☑ 第4章：任务三 项目主菜单的选择执行设计
- ☑ 第5章：任务四 项目主菜单的循环执行设计

第 1 章

C 语言概述

C 语言是规模小、效率高、功能强的专业编程语言，适用于编写各种系统软件和应用软件，近年来在国内外得到广泛推广和应用，成为当前最优秀的程序设计语言之一。本章首先介绍 C 语言的发展及特点，然后通过实例重点介绍 C 语言程序的基本结构和使用 VC++6.0 开发 C 语言程序的过程。



- 了解 C 语言的发展及特点
- 掌握 C 语言程序的基本结构
- 掌握使用 VC++6.0 开发 C 语言程序的过程

1.1 C语言的发展及特点

1.1.1 程序设计语言

计算机语言是指计算机能够接收和处理的具有一定格式的语言,是进行程序设计时最重要的工具之一。计算机语言分为低级语言和高级语言。

1. 低级语言

低级语言依赖于所在的计算机系统,也称为面向机器的语言。由于不同的计算机系统使用的指令系统可能不同,因此使用低级语言编写的程序移植性较差。低级语言主要包括机器语言和汇编语言。

机器语言是由二进制代码“0”和“1”组成的若干个数字串。用机器语言编写的程序称为机器语言程序,它能够被计算机直接识别并执行。但是,程序员直接编写或维护机器语言程序是很难完成的。

汇编语言是一种借用助记符表示的程序设计语言,其每条指令都对应着一条机器语言代码。汇编语言也是面向机器的,即不同类型的计算机系统使用的汇编语言不同。用汇编语言编写的程序称为汇编语言程序,它不能由计算机直接识别和执行,必须由“汇编程序”翻译成机器语言程序,才能够在计算机上运行。这种“汇编程序”称为汇编语言的翻译程序。汇编语言适用于编写直接控制机器操作的底层程序。汇编语言与机器联系仍然比较紧密,不容易使用。

2. 高级语言

高级语言编写的程序易读、易修改、移植性好。但使用高级语言编写的程序不能直接在机器上运行,必须经过语言处理程序的转换,才能被计算机识别。按照转换方式的不同,可将高级语言分为解释型和编译型两大类。

所谓解释型转换,是将编写的程序逐句翻译,翻译一句执行一句,即边翻译边执行,其中转换工作是由解释器自动完成的。常见的解释性语言包括 BASIC 语言和 Perl 语言。解释型转换方式的优点是比较灵活,可以动态地调整和修改程序;缺点是效率比较低,不能生成独立的可执行文件,即程序的运行不能脱离其解释器。

编译型语言编写的程序经过翻译等处理后,可以脱离其语言环境而独立地执行。C 语言、Pascal 语言等大多数编程语言都属于编译型语言。

高级语言按其发展过程,又可分为面向过程的语言、面向对象的语言和面向构件的语言 3 类。

面向过程的语言具有以下特点。

(1) 采用模块分解与功能抽象的方法,自顶向下,逐步求精。

(2) 按功能划分为若干个基本的功能模块,形成一个树状结构。各模块间的关系尽可能简单,功能上相对独立。每一个功能模块内部都是由顺序、选择或循环 3 种基本结构

组成。

面向过程的语言能有效地将一个比较复杂的任务分解成若干个易于控制和处理的子任务。任务的分解有利于程序的设计与维护。C 语言即属于面向过程的语言。

面向过程的程序是按照流水线方式执行的,即一个模块执行结束前,不能执行其他模块,也无法动态地改变程序的执行方向。而在实际处理事务时,总期望每发生一件事情就可以进行处理,即程序应该从面向过程改为面向具体的应用功能。20 世纪 80 年代初期,非过程化的程序设计语言开始出现。非过程化程序设计语言(即面向对象程序设计语言)的目标是实现软件的集成化,把相互联系的数据以及对数据的操作封装成通用的功能模块,各功能模块可以相互组合,完成具体的应用。各功能模块还可以重复使用,而用户不必关心其功能是如何实现的。C++、Java 等是典型的面向对象的语言。

面向构件的语言则提倡最大限度地进行资源共享和软件重用。面向构件编程也称为 Web 服务编程,其目标是将应用服务提供商开发出的各种构件放在自己的服务器上,供网络上的其他人使用。C#语言是面向构件的语言。

1.1.2 C 语言的发展

C 语言是 1972 年由美国的 Dennis M. Ritchie 设计开发的,由早期的编程语言 BCPL (Basic Combined Programming Language) 发展演变而来。早期的 C 语言主要用于 UNIX 操作系统,随着 UNIX 操作系统的广泛使用,C 语言也迅速得到推广,并出现了许多版本。由于没有统一的标准,这些 C 语言之间出现了一些不一致的地方,为了改变这一状况,美国国家标准协会(ANSI)根据 C 语言问世以来的各种版本,对 C 语言进行了改进和扩充,制定了 ANSI C 标准,成为现行的 C 语言标准。

目前,在计算机上广泛使用的 C 语言编译系统有 Borland C++、Turbo C、Microsoft Visual C++ (简称 VC++) 等。本书使用的开发环境是 VC++6.0 系统。

1.1.3 C 语言的特点

和其他语言相比,C 语言具有以下主要特点。

(1) C 语言简洁、紧凑

C 语言简洁、紧凑,而且程序书写形式自由,使用方便、灵活。

(2) C 语言是高、低级兼容语言

C 语言又称为中级语言,它介于高级语言和低级语言(汇编语言)之间,既具有高级语言面向用户、可读性强、容易编程和维护等优点,又具有汇编语言面向硬件和系统并可以直接访问硬件的功能。

(3) C 语言是一种结构化的程序设计语言

结构化语言的显著特点是程序与数据独立,从而使程序更通用。这种结构化方式可使程序层次清晰,便于调试、维护和使用。

(4) C 语言是一种模块化的程序设计语言

所谓模块化,是指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