



高等法律职业教育系列教材

C/C++ 程序设计

C/C++ CHENGXU SHEJI

主 编 ○ 李伶俐



中国政法大学出版社

T17312C
1562



高等法律职业教育系列教材

C/C++ 程序设计

C/C++ CHENGXU SHEJI

主 编 ○ 李伶俐

副主编 ○ 陈晓明 黄少荣

撰稿人 ○ (以撰写章节先后为序)

李伶俐 陈晓明 黄少荣

史聪慧 陈丽仪



中国政法大学出版社

2015 · 北京

声 明 1. 版权所有，侵权必究。
2. 如有缺页、倒装问题，由出版社负责退换。

图书在版编目（C I P）数据

C/C++程序设计 / 李玲俐主编. —北京：中国政法大学出版社，2015. 2
ISBN 978-7-5620-5895-3

I. ①C… II. ①李… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第024455号

出 版 者	中国政法大学出版社
地 址	北京市海淀区西土城路 25 号
邮 箱	fadapress@163.com
网 址	http://www.cuplpress.com (网络实名：中国政法大学出版社)
电 话	010-58908435 (第一编辑部) 58908334 (邮购部)
承 印	固安华明印业有限公司
开 本	787mm×1092mm 1/16
印 张	19.75
字 数	584 千字
版 次	2015 年 2 月第 1 版
印 次	2015 年 2 月第 1 次印刷
印 数	1~500 册
定 价	45.00 元

高等法律职业教育系列教材 审定委员会



主 任 张文彪

副主任 张友生 万安中

委 员 (按姓氏笔画排序)

王 亮 王冰路 刘 洁 刘晓辉

李雪峰 李忠源 陈晓明 周静茹

项 琼 盛永彬 黄惠萍

总序

Preface

高等法律职业化教育已成为社会的广泛共识。2008年,由中央政法委等15部委联合启动的全国政法干警招录体制改革试点工作,更成为中国法律职业化教育发展的里程碑。这也必将带来高等法律职业教育人才培养机制的深层次变革。顺应时代法治发展需要,培养高素质、技能型的法律职业人才,是高等法律职业教育亟待破解的重大实践课题。

目前,受高等职业教育大趋势的牵引、拉动,我国高等法律职业教育开始了教育观念和人才培养模式的重塑。改革传统的理论灌输型学科教学模式,吸收、内化“校企合作、工学结合”的高等职业教育办学理念,从办学“基因”——专业建设、课程设置上“颠覆”教学模式:“校警合作”办专业,以“工作过程导向”为基点,设计开发课程,探索出了富有成效的法律职业化教学之路。为积累教学经验、深化教学改革、凝塑教育成果,我们着手推出“基于工作过程导向系统化”的法律职业系列教材。

《国家(2010~2020年)中长期教育改革和发展规划纲要》明确指出,高等教育要注重知行统一,坚持教育教学与生产劳动、社会实践相结合。该系列教材的一个重要出发点就是尝试为高等法律职业教育在“知”与“行”之间搭建平台,努力对法律教育如何职业化这一教育课题进行研究、破解。在编排形式上,打破了传统篇、章、节的体例,以司法行政工作的法律应用过程为学习单元设计体例,以职业岗位的真实任务为基础,突出职业核心技能的培养;在内容设计上,改变传统历史、原则、概念的理论型解读,采取“教、学、练、训”一体化的编写模式。以案例等导出问题,根据内容设计相应的情境训练,将相关原理与实操训练有机地结合,围绕关键知识点引入相关实例,归纳总结理论,分析判断解决问题的途径,充分展现法律职业活动的演进过程和应用法律的流程。

法律的生命不在于逻辑,而在于实践。法律职业化教育之舟只有驶入法律实践的海洋当中,才能激发出勃勃生机。在以高等职业教育实践性教学改革为平台进行法律职业化教育改革的路径探索过程中,有一个不容忽视的现实问题:高等职业教育人才培养模式主要适用于机械工程设计等以“物”作为工作对象的职业领域,而法律职业教育主要针对的是司法机关、行政机关等以“人”作为工作对象的职业领域,这就要求在法律职业教育中对高等职业教育人才培养模式

进行“辩证”地吸纳与深化，而不是简单、盲目地照搬照抄。我们所培养的人才不应是“无生命”的执法机器，而是有法律智慧、正义良知、训练有素的有生命的法律职业人员。但愿这套系列教材能为我国高等法律职业化教育改革作出有益的探索，为法律职业人才的培养提供宝贵的经验、借鉴。



2010年11月15日

前言

Foreword

C语言是应用十分广泛的计算机语言，具有功能强大、使用灵活方便、应用面广、目标程序效率高、可移植性好等特点。C++是一门高效实用的程序设计语言，从C进化而来，是C语言的超集，它既可以进行过程化程序设计，也可以进行面向对象程序设计，实现了类的封装、数据隐藏、继承和多态，使得其代码容易维护并具有高度可用性，所以深受广大程序员的青睐。C/C++已经成为计算机类和非计算机类本科生、高职高专生的必修课程。

本书针对读者的特点和认知能力，在结构组织上，以项目任务为中心，全书分为12个单元，每个单元包含若干个项目，每个项目分解成若干个任务，每个任务又有多个实例，将枯燥的理论学习与编程转换为基于 workflow 导向的任务驱动模式；在内容上，力求准确精要、层次清晰、通俗易懂。本书特点如下：

1. 单元一详细描述了算法的流程，以及编程工具Microsoft Visual C++ 6.0的使用方法。
2. 前十个单元主要介绍C语言，作为C++面向对象程序设计语言的基础。单元十一是C++对C的扩充内容，包括C++的特点、C++标准设备输入/输出函数、函数重载等。单元十二是面向对象程序设计基础，其内容包括类和对象、构造函数和析构函数、友元、继承和派生、多态性和虚函数等，为学习者进一步学习C++和C#奠定基础。
3. 本书以“经典实例→算法思路→程序实现→相关知识→引申拓展”的方式组织内容，然后通过单元习题加以巩固，符合学生的认知过程，把任务驱动、案例教学和启发式教学等多种教学方法融入其中。
4. 综合实训（学生成绩管理系统）突出以实践技能为核心，注重全面提高学生的职业实践技能和专业素养。

本书是主编及教学团队在总结多年教学实践经验的基础上编写而成的，适于作为高职高专计算机专业和相关专业的程序设计基础课程教材，也可作为“全国计算机水平考试二级”的辅导教材，或作为自学C/C++程序设计的参考用书。

本书由李伶俐任主编，陈晓明和黄少荣为副主编，史聪慧和陈丽仪参与编写。全书分工如下：第1~5、8~9、11~12单元以及综合实训由李伶俐编写，第6单元由陈晓明编写，第7单元由黄

少荣编写，第10单元由史聪慧编写，附录部分由陈丽仪编写。全书由李玲俐策划并确定框架结构，最后由李玲俐和陈晓明定稿。

本书在编写过程中，得到了教务及科研部门的支持和兄弟院校同类专业老师的帮助，在此向所有为本书做出贡献的同志致以衷心的感谢。

由于作者水平及视界所限，书中难免会有错误和疏漏之处，恳请广大读者给予批评指正。

编 者

2015年1月

Abstract 内容简介

本书是根据高职高专培养目标和信息技术类教学大纲的要求，依据高职院校培养实际要求编写的程序设计类C语言教材。该书比较全面、系统、循序渐进地介绍了C/C++语言的基本概念、各种语法成分及其在程序设计中的应用，并提供了大量实例和习题，引导读者掌握一般程序设计的方法和基本的编程方法。同时，根据现行教学需求，本书还将《全国高等学校计算机水平考试C++二级考试大纲》中有关C/C++语言程序设计的要求及考点融入其中。

本书共有12个单元以及综合实训，主要内容包括程序设计与C语言的基本概念、数据类型、运算符和表达式、选择和循环结构程序设计、数组、函数、存储类别和作用域、编译预处理、指针、结构体和共用体、文件，以及C++流的概念、面向对象类和对象的概念、C++面向对象基础等。

任务二 掌握变量的定义和赋值	25
任务三 掌握整型变量的定义和使用	26
任务四 掌握实型变量的定义和使用	27
任务五 掌握字符型变量的定义和使用	28
项目四 掌握不同类型数据间的转换	30
任务一 掌握自动类型转换	30
任务二 掌握强制类型转换	30
项目五 掌握运算符和表达式的使用	32
任务一 了解算术运算符和算术表达式	32
任务二 了解赋值运算符和赋值表达式	34
任务三 了解关系运算符和关系表达式	36
任务四 了解逻辑运算符和逻辑表达式	37
任务五 了解条件运算符和条件表达式	38
任务六 了解逗号运算符和逗号表达式	39
单元习题	40
单元三 C程序设计的三种基本结构	42
项目一 掌握顺序结构程序设计	42
任务一 了解C语句	42
任务二 熟悉库函数和头文件	43
任务三 掌握字符数据的输入与输出	44
任务四 掌握格式的输入和输出	45
任务五 顺序结构程序设计实例	52
项目二 掌握选择结构程序设计	54
任务一 掌握if语句的用法	54
任务二 掌握switch语句的用法	57
任务三 学会用if语句和switch语句编程	58
项目三 掌握循环结构程序设计	60
任务一 了解goto语句和if语句构成循环	60
任务二 掌握while循环语句	61
任务三 掌握do...while循环语句	64
任务四 掌握for循环语句	65
任务五 掌握循环嵌套的使用形式	68
项目四 掌握辅助控制语句的使用	69
任务一 掌握break语句的使用	69
任务二 掌握continue语句的使用	71
单元习题	72
单元四 数 组	74
项目一 一维数组的定义和引用	74
任务一 掌握一维数组的定义	74
任务二 掌握一维数组的引用	75
任务三 掌握一维数组的初始化	75
任务四 一维数组应用实例	76

项目二 二维数组的定义和引用	78
任务一 掌握二维数组的定义	78
任务二 掌握二维数组的引用	79
任务三 掌握二维数组的初始化	79
任务四 二维数组应用实例	80
项目三 字符数组和字符串	82
任务一 掌握字符数组的定义、初始化和引用	82
任务二 了解字符串和字符串结束标志	84
任务三 掌握字符数组的输入输出	85
任务四 了解常用的字符串处理函数	88
任务五 字符数组应用实例	89
单元习题	93
单元五 函 数	95
项目一 了解函数	95
项目二 掌握函数的定义和函数的说明	96
任务一 掌握函数的定义方法	96
任务二 掌握函数的声明方法	97
项目三 掌握函数的参数和返回值	99
任务一 掌握形式参数和实际参数的用法	99
任务二 掌握函数的返回值	101
项目四 函数的调用	103
任务一 掌握函数调用的一般形式	103
任务二 掌握函数调用的方式	103
项目五 函数的嵌套调用	106
项目六 函数的递归调用	108
项目七 数组作为函数的参数	112
任务一 数组元素作为函数的参数	112
任务二 数组名作为函数的参数	114
单元习题	115
单元六 存储类别和作用域	117
项目一 掌握变量的作用域	117
任务一 掌握局部变量的概念和使用	117
任务二 掌握全局变量的概念和使用	119
项目二 了解变量的存储类别	124
任务一 了解auto变量	124
任务二 了解register变量	124
任务三 掌握extern变量	125
任务四 掌握static变量	127
项目三 内部函数和外部函数	129
任务一 掌握内部函数的用法	129
任务二 掌握外部函数的用法	129
单元习题	130

单元七 编译预处理	133
项目一 掌握文件包含	133
项目二 了解宏定义	136
任务一 了解不带参数的宏定义	136
任务二 了解带参数的宏定义	138
项目三 条件编译	141
单元习题	145
单元八 指针	147
项目一 理解地址和指针的概念	147
项目二 掌握变量的指针和指向变量的指针变量	149
任务一 掌握指针变量的定义	149
任务二 掌握指针变量的引用	150
任务三 掌握指针变量作为函数参数	153
项目三 掌握指针与数组的使用	155
任务一 理解指向数组元素的指针	155
任务二 掌握通过指针来引用数组元素	156
任务三 掌握数组名作为函数参数	159
任务四 了解指向多维数组的指针和指针变量	165
项目四 掌握指针与字符串的使用	171
任务一 掌握字符串的表示形式	171
任务二 掌握字符串指针作函数参数的使用	175
任务三 对使用字符指针变量和字符数组的讨论	179
项目五 掌握指针与函数的使用	181
任务一 掌握用函数指针变量调用函数	181
任务二 用指向函数的指针作函数参数	183
项目六 掌握函数返回指针值的方法	185
项目七 掌握用指针数组处理字符串	186
单元习题	189
单元九 结构体和共用体	191
项目一 了解结构体类型与定义	191
项目二 掌握结构体类型变量的使用	192
任务一 掌握定义结构体类型变量的方法	192
任务二 掌握结构体类型变量的引用	195
任务三 掌握结构体类型变量的初始化	195
项目三 了解结构体数组	196
任务一 了解结构体数组的定义	196
任务二 了解结构体数组的初始化	197
任务三 结构体数组应用实例	198
项目四 掌握指向结构体类型数据的指针	200
任务一 掌握指向结构体变量的指针的使用	200
任务二 掌握指向结构体数组的指针的用法	201
任务三 用结构体变量和指向结构体的指针作为函数参数	203

项目五 理解动态存储分配	205
项目六 了解链表的基本操作	207
任务一 了解链表的概念	207
任务二 如何建立动态链表	209
任务三 对动态链表的基本操作	210
项目七 了解共用体的定义和使用	217
任务一 了解共用体类型的定义	217
任务二 了解共用体变量的定义和引用方式	218
任务三 共用体变量应用实例	219
项目八 了解枚举类型的用法	221
项目九 学会用typedef定义类型	224
单元习题	226
单元十 文 件	228
项目一 了解C语言的文件的概念	228
项目二 了解文件类型指针	229
项目三 掌握文件的打开和关闭	230
任务一 掌握文件的打开 (fopen函数)	230
任务二 掌握文件的关闭 (fclose函数)	232
项目四 了解文件的读写函数	232
任务一 了解fputc函数和fgetc函数 (putc函数和getc函数)	232
任务二 了解fread函数和fwrite函数	236
任务三 了解fprintf函数和fscanf函数	239
任务四 了解其他读写函数	242
项目五 了解文件的定位	243
任务一 了解rewind函数	243
任务二 了解fseek函数和随机读写	244
任务三 了解ftell函数	245
项目六 了解出错的检测函数	245
项目七 文件输入输出小结	246
单元习题	246
单元十一 C++对C的扩充	248
项目一 了解C++的特点	248
项目二 了解最简单的C++程序	249
项目三 掌握C++的输入输出	249
任务一 掌握用cout进行输出	250
任务二 掌握用cin进行输入	250
项目四 掌握函数的重载的用法	251
项目五 了解带缺省参数的函数	254
项目六 理解变量的引用类型	255
任务一 了解引用的概念	255
任务二 引用的简单使用	256
任务三 掌握引用作为函数参数的用法	256

项目七 理解内联函数的概念和使用	258
项目八 掌握作用域运算符的使用	259
项目九 了解动态分配和撤销内存的运算符new和delete	260
单元习题	261
单元十二 C++面向对象基础	263
项目一 了解面向对象程序设计方法	263
项目二 掌握类和对象	264
任务一 理解对象和类的概念	264
任务二 掌握类的定义方法	264
任务三 掌握对象的定义和成员访问	266
任务四 定义类的成员函数	267
任务五 C++类的实现实例	268
项目三 掌握构造函数和析构函数的使用	268
任务一 掌握构造函数的使用	269
任务二 了解析构函数的使用	271
项目四 用const限定类的成员函数	272
项目五 了解静态成员	273
任务一 了解静态数据成员	274
任务二 了解静态成员函数	275
项目六 了解友元函数与友元类	276
任务一 了解友元函数	276
任务二 了解友元类	278
项目七 掌握继承和派生	279
任务一 理解继承与派生的概念	279
任务二 掌握建立派生类的方法	280
任务三 掌握公用派生类的用法	281
任务四 了解派生类的构造函数	283
项目八 了解多态性和虚函数	285
任务一 了解多态性的概念	285
任务二 了解虚函数的定义和使用	285
任务三 了解纯虚函数	287
单元习题	288
综合实训 学生成绩管理系统	292
附录A 常用字符与ASCII代码对照表	293
附录B C语言中的关键字	294
附录C 运算符和结合性	295
附录D C库函数	297
参考文献	302

单元一

程序设计与C语言概述

 导 读

计算机是一种具有一定存储能力、在程序控制下自动工作的电子设备。为了使计算机发挥作用，需要为它编写各类不同的程序。编写程序时，不仅要认真考虑程序的数据结构和算法，还要采用正确的方法进行程序设计，并且用一种程序设计语言来表示。这一切，对于学习、掌握C语言，都是至关重要的。

 知识与能力目标

- ◇ 了解程序和程序设计语言的概念
- ◇ 理解算法的表示方法和流程图的作用
- ◇ 了解C语言的历史背景和特点
- ◇ 了解C语言的程序格式和结构特点
- ◇ 熟悉Visual C++ 6.0开发环境的使用与程序调试

项目一 了解程序和程序设计语言

任务一 了解程序的概念

程序是指存储在计算机内部存储器中，可以连续执行的一条条指令的集合。一个程序应该包括两个方面的内容：

1. 数据结构（Data Structure）。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。
2. 算法（Algorithm）。对操作的描述，即操作步骤，是用来解决“做什么”和“怎么做”的问题。

著名科学家沃思（Niklaus Wirth）提出一个公式：

$$\text{程序} = \text{数据结构} + \text{算法}$$

实际上，程序设计人员在设计一个程序时，除了需要考虑数据结构和算法两个因素外，还应当采用结构化程序设计方法进行程序设计，并且用一种计算机语言表示。因此，可以这样表示：

$$\text{程序} = \text{数据结构} + \text{算法} + \text{程序设计方法} + \text{语言工具和环境}$$

任务二 了解程序设计语言

程序设计语言（Programming Language）是计算机所能识别的代码，计算机代码通常能够向计算机描述清楚“做什么”、用“什么做”这两个问题。计算机程序就是用计算机语言书写的、能完成一定功能的代码序列。随着计算机技术的发展，计算机语言也不断地向语言更加丰富、语句更容易理解的方向发展，以扩大计算机的应用范围。

程序设计语言按照语言级别可以分为低级程序设计语言和高级程序设计语言。

低级程序设计语言提供的语句是计算机所能进行的基本操作，如数据传送指令、算术运算指令、逻辑运算指令、串操作指令、控制转移指令、条件转移指令、控制指令、位操作指令等。这些操作和日常用语差别很大，理解它们需要对计算机结构有一定的了解。低级程序设计语言包括机器语言和汇编语言。

高级程序设计语言是接近于自然语言或数学语言的计算机语言，本书介绍的C语言就是高级语言。高级语言不再面向机器，而是面向解题的过程，因而又称为算法语言或过程语言。利用高级语言编写程序，编程者不需要掌握过多的计算机专业知识，而将主要精力放在算法描述上。

对于计算机本身来说，它并不能直接识别由高级语言编写的程序，它只能接收和处理由0和1的代码构成的二进制指令或数据，这种形式的指令是面向机器的，也称为机器语言。计算机所能直接接受的是二进制信息，因此，利用高级语言编写的程序应转变为机器语言，才能在计算机上运行。

利用高级语言编写程序的过程是：借助每种语言提供的各自的编辑软件生成各自的高级语言源程序，利用各自的翻译程序（编译或解释程序）将高级语言源程序自动翻译成目标程序（.obj文件），再将目标程序与高级语言提供的各种库函数进行连接，生成一个可执行文件（.exe文件）。整个过程可以用图1-1表示。

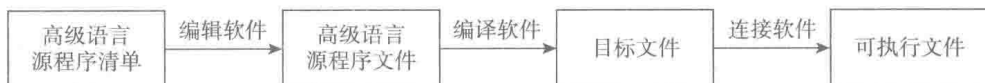


图1-1 高级语言编写程序的过程

项目二 了解算法

算法是程序设计的灵魂，实际上，程序中的操作语句就是算法的体现。

任务一 了解算法的概念

算法是指为解决某个特定问题而采取的方法和步骤。算法是指令的有限序列，使得给定类型的问题通过有限的指令序列，在有限的时间内被求解。比如，高职院校的学生要报考专升本，首先要填写报名表，交报名费用，拿到准考证，然后按照规定的时间到指定的地点参加考试，得到录取通知书后，到指定的高等院校报到注册。这些步骤是按一定的顺序进行的，各个步骤缺一不可，它们之间的次序也不能颠倒。可见，算法体现了人们解决某一类问题时的思维方法和过程，描述了人类解决某类问题所依据的规则和操作。

任务二 简单算法举例

【实例1-1】求 $1 \times 2 \times 3 \times 4 \times 5$ 。

可以用最原始的方法进行。其步骤如下：

- （1）先求 1×2 ，得到结果2。
- （2）将步骤（1）得到的乘积2再乘以3，得到结果6。
- （3）将步骤（2）得到的乘积6再乘以4，得到24。
- （4）将步骤（3）得到的乘积24再乘以5，得到120，这就是最后的结果。

这样的算法虽然是正确的，但太繁琐。如果要求 $1 \times 2 \times 3 \times 4 \times \cdots \times 1000$ ，则要写999个步骤，显然是不可取的。而且每次都直接使用上一步骤的数值结果（如2、6、24等），也不方便。应当找到一种通用的表示方法。

可以设两个变量，一个变量代表被乘数，一个变量代表乘数。不另设变量存放乘积结果，而直接将每一步骤的乘积放在被乘数变量中。设p为被乘数，i为乘数，用循环算法来求结果，可以将算法改写如下：

S1：使 $p=1$ 。