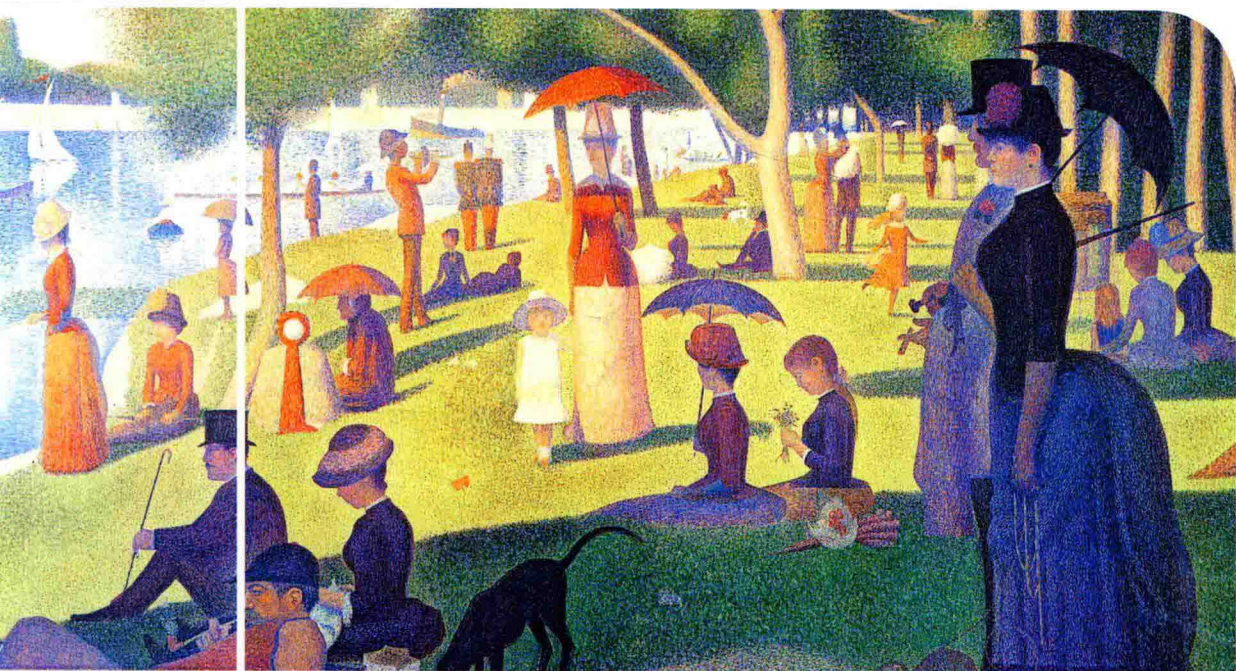




21世纪高等学校计算机
基础实用规划教材

C语言程序设计教程与项目实训

◎ 丁学钧 温秀梅 主编

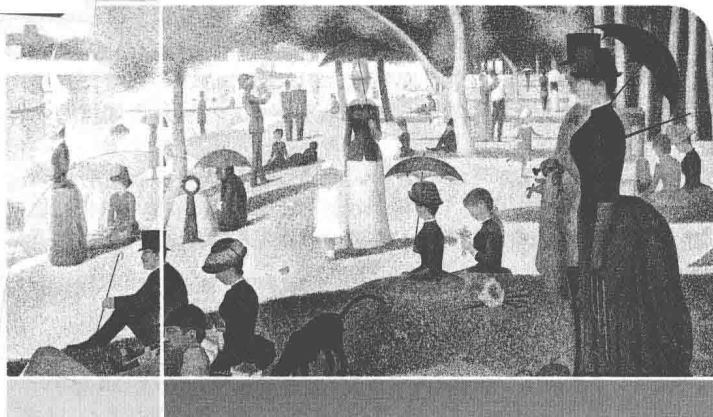


清华大学出版社





21世纪高等学校计算机
基础实用规划教材



C语言程序 设计教程与项目实训

◎ 丁学钧 温秀梅 主编

清华大学出版社

北京

内 容 简 介

本书共分3篇：**初级篇**（C语言基础知识及程序设计初步）、**提高篇**（C语言中的复杂数据类型与高级编程技术）和**实训篇**（基于C语言面向过程的项目实训）。

本书在内容编排上对C语言的知识不求全面系统，但求有理论、有深度、有层次，使其真正能够承担起第一门专业软件课程的重任，对大一新生有一定的专业引导性；特别是实训篇的内容在目前已有教材中未曾见到过。

本书中的所有程序都是在VC++ 6.0环境下编译调试通过的。本着少而精的原则，全书版面清晰、结构紧凑，知识信息含量高。本书可以作为高校（本科、专科）计算机及其相关专业的教材，也可以作为非计算机专业的教学参考书、培训班教材和自学教材。

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语言程序设计教程与项目实训/丁学钧，温秀梅主编. —北京：清华大学出版社，2016

（21世纪高等学校计算机基础实用规划教材）

ISBN 978-7-302-45469-4

I. ①C… II. ①丁… ②温… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字（2016）第274698号

责任编辑：黄 芝 王冰飞

封面设计：刘 键

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

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

印 刷 者：北京富博印刷有限公司

装 订 者：北京市密云县京文制本装订厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：19 字 数：459千字

版 次：2016年12月第1版 印 次：2016年12月第1次印刷

印 数：1~2000

定 价：39.50元

产品编号：060891-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和教学方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程（简称‘质量工程’）”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域，以公共基础课为主、专业基础课为辅，横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业，强调计算机在各专业中的应用。教材内容坚持基本理论适度，反映各层次对基本理论和原理的需求，同时加强实践和应用环节。

(2) 反映教学需要，促进教学发展。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略，突出重点，保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本，合理配套。基础课和专业基础课教材配套，同一门课程可以有针对性针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材、教学参考书，文字教材与软件教材的关系，实现教材系列资源配套。

(5) 依靠专家，择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材

建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主题。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材

联系人：魏江江 weijj@tup.tsinghua.edu.cn

前 言

在编写本书之前，本书作者已在高校从事多年的“高级语言（C、C++、C#）程序设计”、“数据结构”、“面向对象程序设计”及“软件项目开发”等计算机专业软件类课程的教学及研究工作。当站在软件工程及软件项目开发的高度，回过头来重新审视C语言，对于由该语言来承担计算机专业本科生软件系列第一门课程的重任，有了更深入的理解。但目前已有的关于C语言的教材大多只罗列了C语言的语法和应用实例，而不重视编程能力的培养，故组织编写了本书，旨在对C语言的知识体系重新梳理整合，在内容安排及教学深度等方面满足本科计算机及相关专业“C语言程序设计”课程的教学目标，使学生从专业的角度重视该课程并为后续课程的学习打下坚实的基础。

本书共分3篇，即**初级篇**（C语言基础知识及程序设计初步）、**提高篇**（C语言中的复杂数据类型与高级编程技术）和**实训篇**（基于C语言面向过程的项目实训）。

初级篇（第1~4章）介绍C语言的入门基础。我们将C语言的语法、基本数据类型、结构化编程思想、3种控制结构和数组都归属于C语言的入门基础。初学者通过本篇的学习，能够掌握C语言中最基础的主要知识点，有能力编制出结构清晰、代码规范的小程序，获得学习的自信。在C语言的学习过程中，学习并掌握数组是重中之重，数组也是初学者认识数据结构的起点，所以我们将数组归纳到了初级篇中。

提高篇（第5~8章）主要涉及函数、指针和结构体等知识。介绍利用函数组织程序结构，实现模块化设计的方法；利用数组及结构体描述大量数据，使用链表组织数据，使学习者有能力完成较大规模的程序。“指针”让C语言体现出紧凑和灵活的特点，这一点在语言中至关重要。有了数组、指针和函数的综合使用，C语言才更加鲜活。在提高篇中还将学习一些经典算法的思想和实现，使学生进一步地掌握调试程序的方法，能够在一定程度上掌握部分C语言的高级编程技术。

实训篇（第9~12章）是我们在这套教材中特别设计的“项目实训”部分，希望能够规范教师的教学指导与要求，通过一周的项目实训，使学生理论联系实际及编程的能力均能得到良好的训练。实训项目的目标是设计一个模拟的“银行储蓄业务系统”。学生在老师的指导下，学会利用C语言解决实际问题；帮助初学者真正提高程序设计能力，并初步接触软件工程的思想，做出一个小规模的综合性的应用项目。

本书在内容编排上对C语言的知识不求全面系统，但求有理论、有深度、有层次，使其真正能够承担起第一门专业软件课程的重任，对大一新生有一定的专业引导性；特别是实训篇的内容在目前已有教材中未曾见到过。

本书可以作为高校（本科、专科）计算机及其相关专业的教材，也可以作为非计算机专业的教学参考书、培训班教材和自学教材。

本书由河北建筑工程学院丁学钧、温秀梅任主编并完成统稿，宋淑彩、高丽婷任副主编。参加编写者有：李耀辉（第1~2章），赵巍（第3章），付江龙（第4章、附录A~D），宋淑彩（第5章），高丽婷（第6章），温秀梅（第7章），庞慧（第8章），董颖霞（第9章），丁学钧（第10~12章、附录E）。参加本书部分编写工作的还有李建华、周丽莉等。

在本书的大纲讨论和分工编写过程中，我们始终相互帮助，彼此鼓励，是一次非常难忘的经历。另外，本书在编写过程中引用了许多文献，谨向有关专家学者致以诚挚的谢意。

感谢您选择本书。由于我们的水平有限，书中难免有疏漏、错误和不妥之处，恳请广大读者和专家提出批评和修改意见。对于选用本书的教师，我们提供电子版课件，可从清华大学出版社网站（www.tup.com.cn）下载。

编者

2016年9月

目 录

初级篇 C 语言基础知识及程序设计初步

第 1 章 绪论	3
1.1 程序设计概述	3
1.1.1 程序设计的发展历程	3
1.1.2 结构化程序设计	4
1.1.3 面向对象程序设计	5
1.1.4 程序设计语言	6
1.1.5 语言处理程序	8
1.1.6 C 语言简介	8
1.2 算法及算法的表示	9
1.2.1 算法的概念	9
1.2.2 流程图表示法	9
1.2.3 简单算法设计	11
1.3 C 语言基础	13
1.3.1 字符集	13
1.3.2 标识符	13
1.3.3 关键字	14
1.3.4 运算符	14
1.3.5 分隔符	14
1.3.6 空白符	15
1.3.7 C 程序的基本结构	15
1.4 上机步骤与实验指导	18
1.4.1 程序的编辑、编译、运行和调试	18
1.4.2 Visual C++ 集成开发环境	19
1.4.3 编辑、编译、运行和调试程序	19
第 2 章 基本数据类型	24
2.1 数据类型概述	24
2.2 运算符和表达式	25

2.2.1	常量	25
2.2.2	变量	25
2.2.3	整型	26
2.2.4	实型	27
2.2.5	字符型	28
2.2.6	void 类型	29
2.2.7	运算符概述	30
2.2.8	赋值运算符和赋值表达式	30
2.2.9	算术运算符和算术表达式	32
2.2.10	关系运算符和关系表达式	34
2.2.11	逻辑运算符和逻辑表达式	34
2.2.12	条件运算符和条件表达式	35
2.2.13	逗号运算符和逗号表达式	35
2.2.14	类型转换	36
2.3	数据的输入与输出	37
2.3.1	字符输入输出函数	38
2.3.2	格式输出函数 printf	39
2.3.3	格式输入函数 scanf	41
2.3.4	简单程序举例	43
	习题	46
第 3 章	3 种程序控制结构	48
3.1	顺序结构	48
3.2	选择结构	49
3.2.1	if 语句实现选择结构	49
3.2.2	switch 语句实现选择结构	58
3.3	循环结构	61
3.3.1	while 语句	62
3.3.2	do···while 语句	64
3.3.3	for 语句	65
3.3.4	break 语句和 continue 语句	67
3.3.5	循环嵌套	69
3.4	程序结构综合举例	71
	习题	77
第 4 章	复杂数据类型初探——数组	79
4.1	复杂数据类型概述	79
4.2	一维数组	79
4.2.1	一维数组的定义	79

4.2.2	一维数组元素的引用	80
4.2.3	一维数组的初始化	81
4.2.4	一维数组程序举例	82
4.3	二维数组	85
4.3.1	二维数组的定义	85
4.3.2	二维数组元素的引用	85
4.3.3	二维数组的初始化	86
4.3.4	二维数组程序举例	87
4.4	字符串与数组	90
4.4.1	字符数组的定义	90
4.4.2	字符数组的初始化	90
4.4.3	字符数组的使用	92
4.4.4	字符串处理函数	93
4.4.5	字符数组程序举例	95
	习题	97

提高篇 C 语言中的复杂数据类型与高级编程技术

第 5 章	函数	101
5.1	函数的定义	101
5.2	函数的调用	104
5.2.1	函数原型声明	104
5.2.2	函数的调用	106
5.2.3	参数传递机制	107
5.2.4	函数返回值	111
5.3	函数的嵌套调用	112
5.4	递归函数	113
5.5	变量的作用域	117
5.5.1	局部作用域	118
5.5.2	全局作用域	118
5.6	变量的存储方式和生存期	121
	习题	127
第 6 章	指针	129
6.1	指针的基本概念	129
6.1.1	地址和指针的概念	129
6.1.2	指针变量的定义	130
6.1.3	指针变量运算符	131

6.1.4	指针的运算	135
6.2	指针与数组	138
6.2.1	指向数组的指针	138
6.2.2	指针与字符数组	141
6.2.3	多级指针与指针数组	144
6.2.4	指针与多维数组	149
6.2.5	数组指针	153
6.3	指针与函数	154
6.3.1	指针作为函数参数	154
6.3.2	指针作为函数的返回值——指针函数	157
6.3.3	指向函数的指针——函数指针	158
6.4	动态内存分配与指向它的指针变量	162
	习题	164
第 7 章	编译预处理	165
7.1	宏定义	165
7.1.1	无参宏	166
7.1.2	有参宏	168
7.2	文件包含	171
7.3	条件编译	172
	习题	175
第 8 章	复杂数据类型——结构体、共用体和枚举类型	176
8.1	结构体类型	176
8.2	结构体变量	178
8.2.1	结构体变量的定义	178
8.2.2	结构体变量的初始化	180
8.2.3	结构体变量成员的引用	181
8.3	结构体数组	184
8.3.1	结构体数组的定义	184
8.3.2	结构体数组的初始化	185
8.3.3	结构体数组应用举例	185
8.4	结构体指针	186
8.4.1	指向结构体变量的指针	187
8.4.2	指向结构体数组的指针	188
8.4.3	用结构体变量和结构体指针作为函数参数	189
8.5	用结构体指针处理链表	191
8.5.1	链表概述	191

8.5.2	创建单链表	193
8.5.3	输出链表	196
8.5.4	链表的删除操作	196
8.5.5	链表的插入操作	198
8.5.6	链表的综合操作	200
8.6	共用体类型	202
8.6.1	共用体的概念	202
8.6.2	共用体类型和共用体类型变量的定义	202
8.6.3	共用体成员的引用方式	204
8.6.4	共用体类型的特点和应用	204
8.7	枚举类型	207
8.7.1	枚举类型及枚举变量的定义	207
8.7.2	枚举元素的引用	208
	习题	209

实训篇 基于 C 语言面向过程的项目实训

第 9 章	项目实训概述及数据存储基础	213
9.1	概述	213
9.2	最基本的数据存储技术——文件	213
9.2.1	概述	213
9.2.2	文件类型	214
9.2.3	文件指针	214
9.2.4	文件的打开与关闭	214
9.2.5	文件的读写	215
9.2.6	文件的随机读写	218
第 10 章	需求分析与功能模块设计	220
10.1	需求分析	220
10.2	功能模块设计	220
10.2.1	功能模块划分	220
10.2.2	概要设计	221
第 11 章	详细设计与实现	223
11.1	概述	223
11.2	登录业务模块功能及主函数实现	223
11.2.1	系统登录及主函数功能流程图	223

11.2.2	代码实现步骤	223
11.3	系统菜单功能实现	226
11.3.1	功能需求描述	226
11.3.2	代码实现步骤	226
11.4	开户功能实现	227
11.4.1	功能需求描述	227
11.4.2	代码实现步骤	227
11.5	存款功能与取款功能的实现	231
11.5.1	存款功能需求描述	231
11.5.2	存款代码实现步骤	231
11.5.3	取款功能需求描述	232
11.6	文件的读写操作	234
11.6.1	文件的读写功能需求描述	234
11.6.2	文件的读写功能代码实现步骤	234
11.6.3	与读写文件相关的其他函数	235
11.7	查询功能实现	236
11.7.1	查询功能需求描述	236
11.7.2	查询功能代码实现步骤	237
11.8	销户功能实现	241
11.8.1	销户功能需求描述	241
11.8.2	销户功能代码实现步骤	241
第 12 章	项目测试与总结	244
12.1	项目测试	244
12.1.1	单元测试	244
12.1.2	集成测试	245
12.2	项目总结	246
12.2.1	实训目标与要求	246
12.2.2	实训过程与步骤	247
12.2.3	实训项目总结报告撰写要求	247
12.3	系统局限性分析与功能扩展的考虑	247
12.3.1	系统局限性分析	247
12.3.2	如何进行功能扩展的考虑	248
附录 A	标准字符 ASCII 表	251
附录 B	运算符和结合性	253
附录 C	常见库函数	255

附录 D 模拟考试题	261
模拟考试题一	261
模拟考试题一参考答案	271
模拟考试题二	272
模拟考试题二参考答案	283
附录 E 参考课时安排	285
参考文献	286

初 级 篇

C 语言基础知识及程序设计初步

本篇包括第 1~4 章，介绍 C 语言的入门基础。我们将 C 语言的语法、基本数据类型、结构化编程思想、三种控制结构和数组都归属于 C 语言的入门基础。初学者通过本篇的学习，能够掌握 C 语言中最基础的主要知识点，有能力编制出结构清晰、代码规范的小程序，获得学习中的自信。

在 C 语言的学习过程中，学习并掌握数组是重中之重，数组也是让初学者认识数据结构的起点，所以我们将数组归纳到了初级篇中。

最后强调：对于初学者来说，动手写程序比读程序更显得重要。通过编写代码、上机调试、再修改程序，这样反复的学习过程，体会知识点的连接与贯穿，编程经验就是这样慢慢地积累起来的。

1.1 程序设计概述

程序是由数据和处理数据的操作组成的。数据是操作的对象，操作的目的是对数据进行加工处理，以得到期望的结果。操作步骤也就是算法，即求解问题的方法和步骤。程序实际上就是用计算机语言描述的某一问题的解决步骤，是符合一定语法规则的符号序列。程序设计就是根据计算机要完成的任务，提出相应的需求，然后设计数据结构和算法，编写相应的程序代码，运行并测试代码的正确性，调试程序直到得到正确的运行结果。

1.1.1 程序设计的发展历程

回顾程序设计的发展史，大体上可划分为以下几个不同的时期。

20 世纪 50 年代的程序都是用指令代码或汇编语言编写的，这种程序的设计相当麻烦，编制和调试一个稍大一点的程序常常要花费很长时间，培养一个熟练的程序员更需经过长期训练和实习，这种局面严重地影响了计算机的普及应用。

20 世纪 60 年代高级语言的出现大大地简化了程序设计，缩短了解题周期，因此显示出强大的生命力。此后，编制程序已不再是软件专业人员才能做的事了，一般工程技术人员花上较短的时间学习，也可以使用计算机解题。这个时期，随着计算机的应用日益广泛地渗透到各学科和技术领域，发展出了一系列不同风格的、为不同目标服务的程序设计语言。其中较为著名的有 FORTRAN、COBOL、ALGOL、LISP、PL/1、PASCAL 等十几种语言。高级语言的蓬勃兴起，使得编译和形式语言理论相应地日趋完善，这是该时期的主要特征。但就整个程序设计方法而言，并无实质性的改进。

自 20 世纪 60 年代末到 70 年代初，出现了大型软件系统，如操作系统、数据库，这给程序设计带来了新的问题。大型系统的研制需要花费大量的资金和人力，可是研制出来的产品却可靠性差，错误多，且不易维护和修改。一个大型操作系统有时需要几千人/年的工作量，而所获得的系统又常常会隐藏着几百甚至几千个错误。当时，人们称这种现象为“软件危机”。

为了克服 20 世纪 60 年代出现的软件危机，1968 年北约组织提出“软件工程”的概念。对程序设计语言的认识从强调表达能力为重点转向以结构化和简明性为重点，将程序从语句序列转向相互作用的模块集合。1969 年，E.W.Dijkstra 首先提出了结构化程序设计的概念，他强调从程序结构和风格上来研究程序设计。在软件工程的迫切要求下，20 世纪 70 年代结构化语言获得了蓬勃发展并得到广泛的应用。使用结构化程序设计方法可显著地减少软件的复杂性，提高软件的可靠性、可测试性和可维护性。经过几年的探索和实践，结