

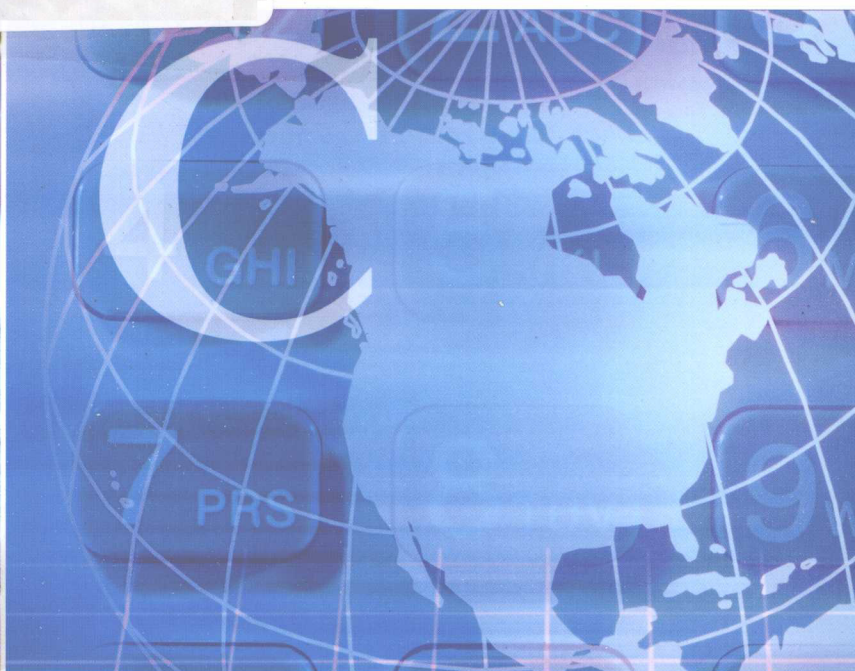
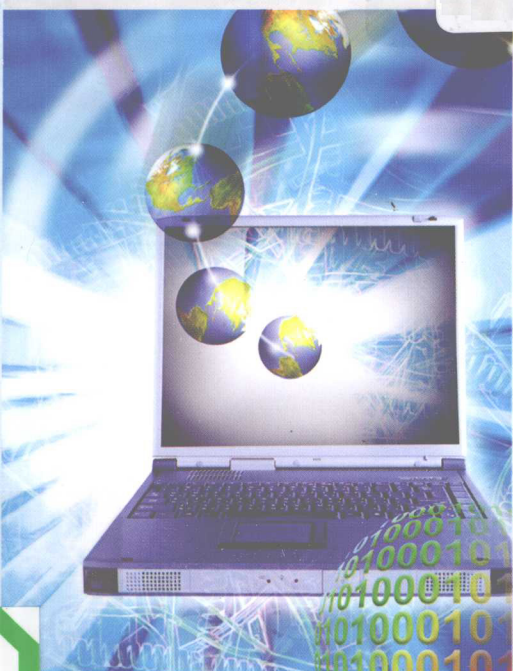


软件职业技术学院“十一五”规划教材

C 语言程序设计

项目引导教程

主 编 杨 威
副主编 张 宇 袁晓曦 陈 娜
主 审 王路群



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书是根据全国计算机等级考试(二级)大纲进行编写,以用C语言编程实现一个完善的学生管理系统为主线,内容涵盖了C语言中各种数据结构、运算符和表达式、常用语句、函数、编译预处理、数组、链表、文件操作等。书中通过翔实的案例对C语言的各项重点进行了详尽的介绍。每章中都附有大量的真实程序案例,并将项目贯穿始终。

本书内容选择恰当、逻辑层次清晰、文字简明易懂。适用于高职高专以及各类高等院校的C语言程序设计教材,也可作为各种成人教育和计算机培训教材,还可供自学者参考。

本书配有电子教案和素材文件,读者可以从中国水利水电出版社网站和万水书苑免费下载,网址为:<http://www.waterpub.com.cn/softdown/>和<http://www.wsbookshow.com>。

图书在版编目(CIP)数据

C语言程序设计项目引导教程 / 杨威主编. — 北京:中国水利水电出版社, 2010.7
软件职业技术学院“十一五”规划教材
ISBN 978-7-5084-7704-6

I. ①C… II. ①王… III. ①王 审 主
C语言—程序设计—高等学校:技术学校—教材 IV.
①TP312

中国版本图书馆CIP数据核字(2010)第138125号

策划编辑:杨庆川 责任编辑:杨元泓 加工编辑:韩莹琳 封面设计:李 佳

书 名	软件职业技术学院“十一五”规划教材 C语言程序设计项目引导教程
作 者	主 编 杨 威 副主编 张 宇 袁晓曦 陈 娜 主 审 王路群
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (营销中心) 82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16开本 16.25印张 392千字
版 次	2010年8月第1版 2010年8月第1次印刷
印 数	0001—3000册
定 价	28.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

序

随着信息技术的广泛应用和互联网的迅猛发展,以信息产业发展水平为主要特征的综合国力竞争日趋激烈,软件产业作为信息产业的核心和国民经济信息化的基础,越来越受到世界各国的高度重视。中国加入世贸组织后,必须以积极的姿态,在更大范围和更深程度上参与国际合作和竞争。在这种形势下,摆在我们面前的突出问题是人才短缺,计算机应用与软件技术专业领域技能型人才的缺乏尤为突出,无论从数量还是质量方面,都远不能适应国内软件产业的发展和信息化建设的需要。因此,深化教育教学改革,推动高等职业教育与培训的全面发展,大力提高教学质量,是迫在眉睫的重要任务。

2000年6月,国务院发布《鼓励软件产业和集成电路产业发展的若干政策》,明确提出鼓励资金、人才等资源投向软件产业,并要求教育部门根据市场需求进一步扩大软件人才培养规模,依托高等学校、科研院所,建立一批软件人才培养基地。2002年9月,国务院办公厅转发了国务院信息化工作办公室制定的《振兴软件产业行动纲要》,该《纲要》明确提出要改善软件人才结构,大规模培养软件初级编程人员,满足软件工业化生产的需要。教育部也于2001年12月在35所大学启动了示范性软件学院的建设工作,并于2003年11月启动了试办示范性软件职业技术学院的建设工作。

示范性软件职业技术学院的建设目标是:经过几年努力,建设一批能够培养大量具有竞争能力的实用型软件职业技术人才的基地,面向就业、产学结合,为我国专科层次软件职业技术人才培养起到示范作用,并以此推动高等职业技术教育人才培养体系与管理体制和运行机制的改革。要达到这个目标,建立一套适合软件职业技术学院人才培养模式的教材体系显得尤为重要。

高职高专的教材建设已经走过了几个发展阶段,由最开始本科教材的压缩到加大实践性教学环节的比重,再到强调实践性教学环节,但是学生在学习时还是反映存在理论与实践的结合问题。为此,中国水利水电出版社在经过深入调查研究后,组织了一批长期工作在高职高专教学一线的老师,编写了这套“软件职业技术学院‘十一五’规划教材”,本套教材采用项目驱动的方法来编写,即全书所有章节都以实例作引导来说明各知识点,各章实例之间并不是孤立的,每个实例都可以作为最终项目的一个组成部分;每一章章末还配有实习实训(或叫实验),这些实训组合起来就是一个完整的项目。

采用这种方式编写的图书与市场上同类教材相比更具优越性,学生不仅仅学到了知识点,还通过项目将这些知识点连成一条线,开拓了思路,掌握了知识,达到了面向岗位的职业教育培训目标。

本套教材的主要特点有:

- (1) 课程主辅分明——重点突出,教学内容实用。
- (2) 内容衔接合理——完全按项目运作所需的知识体系结构设置。

(3) 突出实习实训——重在培养学生的专业能力和实践能力，力求缩短人才与企业间的磨合期。

(4) 教材配套齐全——本套教材不仅包括教学用书，还包括实习实训材料、教学课件等，使用方便。

本套教材适用于广大计算机专业和非计算机专业的大中专院校的学生学习，也可作为有志于学习计算机软件技术与开发的工程技术人员的参考教材。

编委会

2006年7月

前 言

C 语言已成为目前应用最广泛、影响力最大的计算机程序设计语言，也是计算机专业的一门必修课。C 语言概念简洁，运算符丰富，表达能力强，数据类型多，是一种十分灵活的编程语言。C 语言既具有高级语言方便、易读等优点，又具有低级语言的高效、能直接对硬件进行操作等特点。因此，它非常适合编写应用程序，也适用于系统程序的编写。

C 语言是一种结构化编程语言。目前全国计算机等级考试以及 NIT 等考试都有对 C 语言的重点考查。本书遵循由教育部颁布的《职业院校计算机和软件专业领域技能型紧缺人才培养培训指导方案》的精神，根据全国计算机等级考试（二级）考试大纲编写而成。本书既可以作为计算机专业技能入门教材，也可以作为全国计算机等级考试（二级）的培训教材。同时，本书与以往 C 语言类教材最大的不同之处：以项目为引导。全书都是基于一个项目——编写一个完善的学生管理系统的基础上编写的。书中案例丰富，偏重于实用。以项目为引导，以案例为学习材料是本书奉行的原则。

全书共分为 10 章，各章内容简介如下。

- 第 1 章：C 语言概述。介绍 C 语言的发展和开发环境，为初学者打开一扇门。
- 第 2 章：基本数据类型和表达式。介绍有关数据类型和数据表达方面的基本知识。
- 第 3 章：输入/输出函数。专门介绍有关输入/输出函数的使用知识。
- 第 4 章：控制语句。介绍有关程序流程控制方面的知识。
- 第 5 章：函数。系统介绍函数的定义、调用和程序设计方法。
- 第 6 章：编译预处理指令。介绍有关编译过程、宏定义、库的引用知识。
- 第 7 章：数组。介绍数组的定义、引用和初始化等。
- 第 8 章：指针。介绍指针的概念、指针变量的定义、赋值和引用等。
- 第 9 章：结构体和链表。介绍结构体、共同体、链表的特点、定义和引用。
- 第 10 章：文件操作。介绍文件的概念、文件操作的方法等。

本书作者常年从事 C 语言开发和教学，经验丰富。全书围绕一个项目，深入浅出、逻辑清晰、案例丰富。本书具有以下特点：

(1) 以项目为引导，以教会读者开发动手能力为目标，从实用角度来讲授有关 C 语言的相关知识。

(2) 内容上多采用案例教学，让读者能做到举一反三。

(3) 以项目为主线，使读者既能掌握基本知识和技能，又获得实际编程能力。

(4) 面向全国计算机等级考试，案例多为考试中出现过的类似问题，很有针对性，是备战计算机二级的不二选择。

本书由王路群教授担任主审，杨威担任主编，张宇、袁晓曦、陈娜担任副主编。鄢军霞、库波、谢日星、江骏、肖奎、张松慧参加编写，杨威统编全稿，鄢军霞负责全书的校

对工作。

本书图片处理工作得到了冯奕先生的大力支持，在此表示真诚的感谢。

由于时间仓促，编者水平有限，书中难免有错误和不妥之处，敬请各位专家与读者指正，以求共同进步，感激不尽。

编者

2010年6月

一、前言
二、第一章 绪论
三、第二章 基础理论
四、第三章 应用案例
五、第四章 实验设计
六、第五章 结果分析
七、第六章 结论与展望
八、参考文献
九、附录
十、索引

C 语言程序设计项目引导教程

教学大纲

一、目的与任务

本课程为计算机软件技术专业的必修课，也可以是计算机及应用专业、电子技术专业及其他相关专业的选修课。

通过对本课程的学习，可以使学生从零开始逐渐全面地了解 C 语言的相关知识，了解使用 C 语言进行程序设计的全套方法，全面掌握 C 语言的编程技巧，从而为学生在将来可能从事计算机软件开发工作打下一个良好的专业基础。

二、教学内容

“C 语言程序设计”是一门侧重于实践的课程，学生需要通过努力，多动手、多实践，才能熟练掌握这门基础而又有实用价值的计算机语言。为了指导学生更好地将课堂所学到的知识应用到开发过程中去，本教材专门选定了用一个 C 语言开发学籍管理系统的项目，并将其根据每章的教学内容加以分解，然后由教师带领学生动手进行项目开发。通过理论联系实际的学习，学生最终能熟练掌握每章的知识重点，并加以应用。在完成了全书的学习之后，学生不仅能掌握有关 C 语言的知识，还实际参与到了一次项目的开发，得到了锻炼。本教材讲解详细，适用于教师根据不同的教育实际情况组织实践或演示。

三、教学基本要求

学习完本课程，学生应该掌握：

- 1) C 语言概述。
- 2) 基本数据类型和表达式。
- 3) 输入/输出函数。
- 4) 控制语句。
- 5) 函数。
- 6) 编译预处理指令。
- 7) 数组。
- 8) 指针。
- 9) 结构体和链表。
- 10) 文件操作。

四、与其他课程的关系

本课程的先行课程包括：

1) 计算机数学基础。

2) 计算机应用基础。

五、实践环节

本教材要求在教会学生书本知识的基础上，培养学生实践动手和熟练应用理论知识的能力。书中所使用的实例是经过检验能够编译运行通过的。为了更好地指导学生学学习，建议教师在授课前首先进行试验，然后采取“先演示，后动手”的方式组织实践工作。

课程设计的目的在于综合运用所学习到的 C 语言相关知识，让学生制作一个学籍管理系统的项目，结合项目开发的过程，一边学习理论知识，一边进行项目开发。课程考查方式最好综合学生在每章后的子项目开发表现和最终项目开发结果来给出分数，让学生在做的过程当中，综合运用课堂所学知识，从而达到巩固的目的。

六、学时分配

“C 语言程序设计”课程计划学时共 72 学时，其中授课 52 学时，项目开发 20 学时，详见表 1。

表 1 “C 语言程序设计项目引导教程”课程授课学时分配表

课程内容	学时数
1. C 语言概述	4 学时
2. 基本数据类型和表达式	10 学时
3. 输入/输出函数	2 学时
4. 控制语句	6 学时
5. 函数	6 学时
6. 编译预处理指令	6 学时
7. 数组	6 学时
8. 指针	4 学时
9. 结构体和链表	2 学时
10. 文件操作	6 学时
11. 项目开发	20 学时

教材特色

一、教材特点

“C 语言程序设计”课程是一门实践性很强的软件开发专业课。《C 语言程序设计项目引导教程》是该课程的配套教材。本教材具有以下特色和特点。

1. 针对高职高专特点，难易适中，语言精当

本教材由浅入深地向学生讲解了 C 语言这一基础而又重要的计算机高级语言。考虑到高

职高专学生的实际要求,将重点放在了理论联系实际进行项目开发上,并以让高职高专的学生熟练掌握 C 语言开发技能为终极目标。本书语言十分简洁而准确,难度适中,非常适合高职高专学生作为软件技术专业课程学习的教材,也十分适合电脑爱好者作为业余学习的参考资料。

2. 组织结构逻辑性强,由浅入深

由于本课程为理论课,为了帮助学生更好地学习相关内容,本教材在各章节之间的联系、知识点与知识点之间的联系等逻辑结构上倾注了很大精力。本教材的内容组织原则是由浅入深,由难到易,层层拨开,逻辑严谨。教师和学生可以按照目录章节安排,顺序往下学习,就会得到满意的结果。

3. 内容翔实,知识面覆盖广

C 语言程序设计是一门基础性学科,涉及数制和数据类型、内存知识、运算表达、程序结构、库和宏定义、文件操作、数组和指针等方面内容。本教材在编写的过程中,一直注意内容的丰富,力图做到内容涵盖 C 语言程序设计的各个重要方面。

4. 项目引导,贴近学生实际

由于本课程是一门强调动手的学科,因此设定了一个用 C 语言开发学生学籍管理系统的项目,并将其根据每章所讲解的内容分解成若干个子项目。学生每学完一章,就要做一个项目实训,每个实训都是整个项目组成部分之一。这样当全书学习完成之后,学生既巩固了每章的讲解内容,又参与了一次项目开发的全过程,十分符合高职高专教育的特点和要求。

二、编写方法

本教材内容详尽、结构清晰、通俗易懂,大量使用了图表对内容进行表述和归纳,并对重点内容给出了详细的讲解,既突出基础性内容,又重视实践性应用,专业性、针对性和实用性较强。在对实例的分析上力求把复杂的问题简单化,很多内容和方法都是作者在长期软件开发工作过程中积累出来的。

三、内容大纲及重点、难点的分析

● 第一部分:基础知识

这一部分包含八大章,是学习 C 语言编程的必备知识。在其中详细讲解了数据类型、数值转换、运算符、表达式、常用语句、条件转换、循环、函数、预处理、指针、数组等相关内容。

第 1 章 C 语言概述

教学目的:通过对本章的学习,使学生了解 C 语言的基本概念,数制和码制的具体内容,并对用 C 语言进行程序开发过程有一个大致的了解。

本章重点:数制转换、码制转换。

本章难点:数制转换、码制转换。

本章编写说明:本章内容属于概论部分,数制和码制的问题对以后的学习是至关重要的,因为要让学生有较为牢固的数学知识,故采取多举例的方法来进行说明,便于学生能够熟练掌握几种数制和码制的转换。同时用一个简单易懂的小程序实例,让学生对用 C 语言开发程序有一个感官的认识。

第2章 基本数据类型和表达式

教学目的：通过对本章的学习，使学生了解 C 语言中使用的数据类型、存储方式、各种运算符、常量和变量的关系等内容，为今后的学习扫清障碍，打下基础。

本章重点：数据类型、常量和变量、运算符、数据类型转换。

本章难点：本章的难点就是本章的重点。其中又以数据类型转换为难中之难，要学习好 C 语言，必须下苦功学习好本章的所有知识，甚至超越书本。

本章编写说明：本章涉及的是 C 语言最基础的知识，难度不高，但学生必须能熟练掌握，不能出错，学习效果要求很高。在教材编写上，作者尽量采取循序说明，详加解释的方法，让学生在理解上的困难降到最低。

第3章 输入/输出函数

教学目的：这是学生最早学习的函数，也是 C 语言中使用频率最高的两类函数。通过对本章的学习，学生应了解 C 语言的函数使用方式。

本章重点：熟练掌握输入/输出函数。

本章难点：数量地使用输入/输出函数。

本章编写说明：本章正式进入 C 语言程序编写的学习。输入/输出函数在 C 语言中使用频率最高，而且通过对本章的学习，学生要能熟练使用函数，知道如何去查函数参数并使用。

第4章 控制语句

教学目的：通过对本章的学习，学生应掌握程序结构的设计方法，以及几种跳转和循环结构的使用方法。

本章重点：程序结构设计、选择、循环。

本章难点：循环结构的数量使用。

本章编写说明：本章站在宏观的角度，告诉学生如何在编写程序之前，对程序的结构进行规划和设计以及优化。同时要学习的是选择结构和循环结构，这是两种非常常用的条件判断结构，具有很强的逻辑性和很高的工作效率。学生在学习完本章之后，应能熟练使用这些设计方法和跳转来优化程序结构。

第5章 函数

教学目的：通过对本章学习，学生要能深入地理解函数和参数之间的关系，并能做到根据需要准确无误地调用和编写各种函数。

本章重点：函数的调用和声明、变量的存储。

本章难点：参数调用、递归问题、变量的存储和类型。

本章编写说明：函数是 C 语言中实现具体功能的模块，是组成一个程序的重要零件。函数之间是通过参数来传递信息，因此参数调用方法就显得十分重要。参数的调用和变量存储也是一个难点。作者在编写本章的时候，十分注重讲解和实例的结合，力图给读者一个较为明确的指导方向。

第6章 编译预处理指令

教学目的：通过对本章的学习，学生要深入了解编译的原理和过程，同时学会使用宏定义来减轻编程压力。

本章重点：编译原理和宏定义。

本章难点：编译原理和宏定义。

本章编写说明：本章首先站在理论的角度，对编译、连接的过程进行了详细的讲解，然后教授学生使用宏定义的方式来减轻编程中的重复性劳动。

第7章 数组

教学目的：本章的编写目的在于让学生学会使用数组这一重要的数据存储和传递的手段。

本章重点：熟练使用数组的各种应用方法。

本章难点：熟练使用数组的各种应用方法。

本章编写说明：数组是在编写 C 语言程序中经常要用到的一种数据存储和传递的手段，而且在各种类 C 语言中都有包含。学生要在本章的学习中做到熟练掌握各种数组的使用方法。

第8章 指针

教学目的：本章的编写目的在于让学生学会使用指针这一重要的地址传递手段。

本章重点：熟练使用指针的各种应用方法。

本章难点：熟练使用指针的各种应用方法。

本章编写说明：数组是在编写 C 语言程序中经常要用到的一种地址传递手段，而且在各种类 C 语言中都有包含。学生要在本章的学习中做到熟练掌握各种指针的使用方法。

● 第二部分：扩展知识

这一部分有两章，是在第一部分介绍基础知识之后，更加深入地教授学生有关 C 语言的高级技巧和知识，内容包含结构体、共用体及文件操作。

第9章 结构体和链表

教学目的：本章详细地向学生介绍了结构体、共用体、枚举数据类型和链表这几方面的知识，属于高级编程技巧，目的在于让学生的编程手段向高层次提升。

本章重点：结构体、共用体、结构体指针、链表。

本章难点：结构体、共用体、结构体指针、链表。

本章编写说明：本章属于高级编程技巧，里面所涉及的内容都将对学生的编程手段有很大的提升作用，也能让学生在学完本章后，在用 C 语言编程时能站在一个更高的角度。

第10章 文件操作

教学目的：通过对本章的学习，学生能掌握如何用 C 语言来操作文件和文件系统。

本章重点：用 C 语言来操作文件和文件系统。

本章难点：用 C 语言来操作文件和文件系统。

本章编写说明：本章向学生详细讲解了如何用 C 语言来操作文件和文件系统，至此学生所编写的程序就不再是独立存在的，而是拓展到了一个更广阔的空间，调用手段将更加的灵活，功能也将更加的强大。

目 录

序	1
前言	2
第1章 C语言概述	1
1.1 程序设计语言的发展过程	1
1.1.1 C语言的发展	2
1.1.2 C语言的主要特点	2
1.2 C语言的基本组成	3
1.2.1 程序设计的相关基础知识	3
1.2.2 C语言的组成	4
1.2.3 函数和库的简介	6
1.3 开发环境	8
1.3.1 Turbo C 2.0 集成开发环境	8
1.3.2 Visual C++ 6.0 可视化开发环境	14
相关知识链接	15
项目实训	16
习题一	17
第2章 基本数据类型和表达式	18
2.1 基本数据类型	18
2.2 常量和变量	19
2.2.1 标识符	19
2.2.2 常量	20
2.2.3 变量	21
2.3 整型数据	22
2.3.1 整型常量	22
2.3.2 整型变量	23
2.4 实型数据	27
2.4.1 实型常量	27
2.4.2 实型变量	28
2.5 字符型数据	31
2.5.1 字符常量	31
2.5.2 字符变量	33
2.5.3 putchar()函数的使用	35
2.6 不同数据类型间的转换	36
2.6.1 自动类型转换	36
2.6.2 强制类型转换	37
2.7 运算符和表达式	38
2.7.1 算术运算	39
2.7.2 自增、自减运算	40
2.7.3 赋值运算	41
2.7.4 逗号运算	43
2.7.5 关系运算	44
2.7.6 逻辑运算	45
2.7.7 条件运算	47
2.7.8 位运算	48
2.7.9 运算符和表达式的综合应用	50
相关知识链接	56
项目实训	56
习题二	56
第3章 输入/输出函数	58
3.1 格式化输出函数	58
3.1.1 printf()函数的一般格式	58
3.1.2 printf()函数的应用举例	60
3.1.3 printf()函数的使用说明	63
3.2 格式化输入函数	64
3.2.1 scanf()函数的一般格式	64
3.2.2 scanf()函数的应用举例	65
3.2.3 scanf()函数的使用说明	67
3.3 字符输出函数	68
3.3.1 putchar()函数的一般格式	69
3.3.2 putchar()函数的应用举例	69

3.3.3	putchar()函数的使用说明	70	5.3.3	函数的参数和返回值实例	113
3.4	键盘输入函数	70	5.4	函数的调用和返回值	115
3.4.1	getchar()函数的一般格式	70	5.4.1	函数的调用形式	115
3.4.2	getchar()函数的应用举例	71	5.4.2	函数调用的方式	116
3.4.3	getchar()函数的使用说明	71	5.4.3	函数调用的实例	119
	相关知识链接	72	5.5	函数的嵌套调用和递归调用	120
	项目实训	73	5.5.1	函数的嵌套调用	120
	习题三	73	5.5.2	函数的递归调用	122
第4章	控制语句	74	5.5.3	函数的嵌套和递归调用实例	125
4.1	控制语句概述	74	5.6	局部变量和全局变量	127
4.2	顺序结构程序设计	75	5.6.1	局部变量	127
4.3	选择结构程序设计	75	5.6.2	全局变量	129
4.3.1	if语句	76	5.6.3	变量的存储类别	131
4.3.2	if-else语句	76	5.6.4	局部变量和全局变量的实例	136
4.3.3	if语句嵌套	77		相关知识链接	137
4.3.4	switch语句	78		项目实训	137
4.3.5	选择结构程序设计举例	81		习题五	138
4.4	循环语句	83	第6章	编译预处理指令	139
4.4.1	while循环语句	83	6.1	编译预处理指令概述	139
4.4.2	for循环语句	85	6.2	#define宏定义指令	139
4.4.3	do-while循环语句	89	6.2.1	不带参数的宏定义	139
4.4.4	break和continue语句	92	6.2.2	带参数的宏定义	140
4.4.5	循环语句的嵌套	95	6.2.3	删除宏定义	141
4.4.6	循环结构程序设计举例	97	6.3	#include文件包含指令	141
	相关知识链接	99	6.4	条件编译指令	142
	项目实训	100		相关知识链接	144
	习题四	101		项目实训	146
第5章	函数	102		习题六	146
5.1	函数概述	102	第7章	数组	147
5.2	函数的定义	105	7.1	一维数组	147
5.2.1	无参数函数定义	105	7.1.1	一维数组的定义	147
5.2.2	有参数函数定义	106	7.1.2	一维数组的初始化	148
5.2.3	空函数	107	7.1.3	一维数组元素的引用	149
5.2.4	函数定义的实例	107	7.1.4	一维数组作为函数参数	156
5.3	函数的参数和返回值	109	7.2	多维数组	158
5.3.1	形式参数和实际参数	109	7.2.1	多维数组的定义	158
5.3.2	函数的返回值	111	7.2.2	多维数组的初始化	159

7.2.3	多维数组的引用	160	9.3.1	指向结构体变量的指针	195
7.2.4	多维数组作为函数参数	162	9.3.2	指向结构体数组的指针	196
7.3	字符数组和字符串	163	9.3.3	结构体指针作为函数的参数	197
7.3.1	字符数组的定义和初始化	163	9.4	链表	198
7.3.2	字符数组元素的引用	164	9.4.1	结构的嵌套	198
7.3.3	字符数组的单个字符处理	166	9.4.2	动态存储分配	199
7.3.4	字符数组的整体处理	166	9.4.3	链表的概述	200
7.4	常用的字符串处理函数	169	9.4.4	链表的创建	200
相关知识链接		175	9.4.5	链表的遍历	202
项目实训		176	9.4.6	链表结点的插入	203
习题七		176	9.4.7	链表结点的删除	204
第 8 章 指针		177	9.5	共用体和枚举类型简介	205
8.1	地址的概念	177	9.5.1	共用体	205
8.1.1	内存地址	177	9.5.2	枚举类型	206
8.1.2	变量地址	177	相关知识链接		208
8.1.3	变量值的存取	178	项目实训		209
8.2	指针的定义及运算	178	习题九		209
8.2.1	指针的定义	178	第 10 章 文件操作		210
8.2.2	指针运算符	179	10.1	文件的概念	210
8.2.3	指针的赋值与运算	179	10.1.1	文件、文件系统与文件名	210
8.3	指针与数组	181	10.1.2	文件的分类	211
8.3.1	用指针处理数组元素	181	10.1.3	文件的访问方式	211
8.3.2	指针数组	184	10.2	文件类型指针	213
8.4	指针和函数	185	10.3	文件的打开与关闭	213
8.4.1	指针作为函数的参数	185	10.3.1	文件的打开函数 fopen()	213
8.4.2	返回指针的函数	186	10.3.2	文件的关闭函数 fclose()	214
8.4.3	指向函数的指针	187	10.4	文件的读/写	214
相关知识链接		189	10.4.1	fgetc()函数和 fputc()函数	215
项目实训		190	10.4.2	fread()函数和 fwrite()函数	218
习题八		190	10.4.3	fprintf()函数和 fscanf()函数	221
第 9 章 结构体和链表		191	10.4.4	fgets()和 fputs()函数	226
9.1	结构体类型和结构体变量	191	10.5	文件定位函数	226
9.1.1	结构体类型和结构体变量的定义	191	10.5.1	rewind()函数	226
9.1.2	结构体变量的初始化和引用	192	10.5.2	fseek()函数	227
9.1.3	结构体变量作为函数的参数	193	10.5.3	ftell()函数	228
9.2	结构体数组	194	10.6	出错检测	228
9.3	指向结构体类型数据的指针	195	10.6.1	ferror()函数	228

10.6.2 clearerr()函数	228
相关知识链接	228
项目实训	230
习题十	230
891	112
901	112
900	112
914	112
905	112
915	112
916	112
917	112
918	112
919	112
920	112
921	112
922	112
923	112
924	112
925	112
926	112
927	112
928	112
929	112
930	112
931	112
932	112
933	112
934	112
935	112
936	112
937	112
938	112
939	112
940	112
941	112
942	112
943	112
944	112
945	112
946	112
947	112
948	112
949	112
950	112
951	112
952	112
953	112
954	112
955	112
956	112
957	112
958	112
959	112
960	112
961	112
962	112
963	112
964	112
965	112
966	112
967	112
968	112
969	112
970	112
971	112
972	112
973	112
974	112
975	112
976	112
977	112
978	112
979	112
980	112
981	112
982	112
983	112
984	112
985	112
986	112
987	112
988	112
989	112
990	112
991	112
992	112
993	112
994	112
995	112
996	112
997	112
998	112
999	112
1000	112
1001	112
1002	112
1003	112
1004	112
1005	112
1006	112
1007	112
1008	112
1009	112
1010	112
1011	112
1012	112
1013	112
1014	112
1015	112
1016	112
1017	112
1018	112
1019	112
1020	112
1021	112
1022	112
1023	112
1024	112
1025	112
1026	112
1027	112
1028	112
1029	112
1030	112
1031	112
1032	112
1033	112
1034	112
1035	112
1036	112
1037	112
1038	112
1039	112
1040	112
1041	112
1042	112
1043	112
1044	112
1045	112
1046	112
1047	112
1048	112
1049	112
1050	112
1051	112
1052	112
1053	112
1054	112
1055	112
1056	112
1057	112
1058	112
1059	112
1060	112
1061	112
1062	112
1063	112
1064	112
1065	112
1066	112
1067	112
1068	112
1069	112
1070	112
1071	112
1072	112
1073	112
1074	112
1075	112
1076	112
1077	112
1078	112
1079	112
1080	112
1081	112
1082	112
1083	112
1084	112
1085	112
1086	112
1087	112
1088	112
1089	112
1090	112
1091	112
1092	112
1093	112
1094	112
1095	112
1096	112
1097	112
1098	112
1099	112
1100	112
1101	112
1102	112
1103	112
1104	112
1105	112
1106	112
1107	112
1108	112
1109	112
1110	112
1111	112
1112	112
1113	112
1114	112
1115	112
1116	112
1117	112
1118	112
1119	112
1120	112
1121	112
1122	112
1123	112
1124	112
1125	112
1126	112
1127	112
1128	112
1129	112
1130	112
1131	112
1132	112
1133	112
1134	112
1135	112
1136	112
1137	112
1138	112
1139	112
1140	112
1141	112
1142	112
1143	112
1144	112
1145	112
1146	112
1147	112
1148	112
1149	112
1150	112
1151	112
1152	112
1153	112
1154	112
1155	112
1156	112
1157	112
1158	112
1159	112
1160	112
1161	112
1162	112
1163	112
1164	112
1165	112
1166	112
1167	112
1168	112
1169	112
1170	112
1171	112
1172	112
1173	112
1174	112
1175	112
1176	112
1177	112
1178	112
1179	112
1180	112
1181	112
1182	112
1183	112
1184	112
1185	112
1186	112
1187	112
1188	112
1189	112
1190	112
1191	112
1192	112
1193	112
1194	112
1195	112
1196	112
1197	112
1198	112
1199	112
1200	112

附录 A ASCII 码表	231
附录 B C 语言数据类型	235
附录 C 常用 Turbo C 库函数	236
参考文献	241
401	237
601	237
701	237
801	237
901	237
1001	237
1101	237
1201	237
1301	237
1401	237
1501	237
1601	237
1701	237
1801	237
1901	237
2001	237
2101	237
2201	237
2301	237
2401	237
2501	237
2601	237
2701	237
2801	237
2901	237
3001	237
3101	237
3201	237
3301	237
3401	237
3501	237
3601	237
3701	237
3801	237
3901	237
4001	237
4101	237
4201	237
4301	237
4401	237
4501	237
4601	237
4701	237
4801	237
4901	237
5001	237
5101	237
5201	237
5301	237
5401	237
5501	237
5601	237
5701	237
5801	237
5901	237
6001	237
6101	237
6201	237
6301	237
6401	237
6501	237
6601	237
6701	237
6801	237
6901	237
7001	237
7101	237
7201	237
7301	237
7401	237
7501	237
7601	237
7701	237
7801	237
7901	237
8001	237
8101	237
8201	237
8301	237
8401	237
8501	237
8601	237
8701	237
8801	237
8901	237
9001	237
9101	237
9201	237
9301	237
9401	237
9501	237
9601	237
9701	237
9801	237
9901	237
10001	237
10002	237
10003	237
10004	237
10005	237
10006	237
10007	237
10008	237
10009	237
10010	237
10011	237
10012	237
10013	237
10014	237
10015	237
10016	237
10017	237
10018	237
10019	237
10020	237
10021	237
10022	237
10023	237
10024	237
10025	237
10026	237
10027	237
10028	237
10029	237
10030	237
10031	237
10032	237
10033	237
10034	237
10035	237
10036	237
10037	237
10038	237
10039	237
10040	237
10041	237
10042	237
10043	237
10044	237
10045	237
10046	237
10047	237
10048	237
10049	237
10050	237
10051	237
10052	237
10053	237
10054	237
10055	237
10056	237
10057	237
10058	237
10059	237
10060	237
10061	237
10062	237
10063	237
10064	237
10065	237
10066	237
10067	237
10068	237
10069	237
10070	237
10071	237
10072	237
10073	237
10074	237
10075	237
10076	237
10077	237
10078	237
10079	237
10080	237
10081	237
10082	237
10083	237
10084	237
10085	237
10086	237
10087	237
10088	237
10089	237
10090	237
10091	237
10092	237
10093	237
10094	237
10095	237
10096	237
10097	237
10098	237
10099	237
10100	237

第 1 章 C 语言概述

教学要求

- 掌握 C 语言的基本结构和 Turbo C 的基本使用方法
- 理解 C 语言的运行环境
- 了解 C 语言的发展及主要特点

1.1 程序设计语言的发展过程

目前我们所使用的计算机，其实是一种很机械的设备，具体表现在其所能识别的数值只有“0”和“1”这两个二进制数。计算机的所有运算、判断都是基于二进制的，其所存储、使用的所有指令、数据也都是些“0”和“1”组成的二进制数。在计算机最开始的年代，人类就是使用这种方法向计算机发出各种计算指令，获取所需的结果，而所使用的二进制指令就被称为机器语言。机器语言是最简单也是最直接的计算机语言，能被计算机直接读懂并依照执行。但是机器语言有三个问题：第一，脱离了自然语言，从而难以为人所阅读；第二，大量的二进制数串“0”和“1”在编写过程中极易出错；第三，面向不同的计算机系统结构，几乎没有通用性可言。

为了克服机器语言的这三大缺陷，人类继而在其基础上编制出了汇编语言。所谓汇编语言，就是将具有一定意义的机器语言代码用助记符来替代，以提高程序的可阅读性和正确率。汇编语言中的指令和机器语言中的指令是一一对应的关系。相对机器语言，汇编语言的优势非常明显，效率也很高，但是仍然具有难以移植的缺点，这是由于汇编语言是针对某一具体计算机而言的。同时，汇编语言要求开发人员具有相当的计算机系统结构水平，因此难以大面积的推广。

计算机发展的目的是为了帮助人类解决各种烦琐的计算，如果不能达到这一目的，甚至需要投入比计算本身更多的编程成本，则是与其开发初衷相违背的。因此，人类继续在汇编语言的基础上开发出更贴近自然语言、移植性更强的计算机语言，即所谓的高级语言。“高级”是相对汇编语言和机器语言这两种基础语言而言的。高级语言不能被计算机执行，而是需要首先进行编译（其实汇编语言也需要编译），成为机器语言之后方可被执行。所以，高级语言脱离了具体机器，其通用性和可移植性大大增强。程序员只需要使用高级语言编写好源代码，再使用相应的编译器进行编译，即可得到可执行文件。不同的高级语言有自己不同的编译器。

早期的高级语言种类繁多，如 COBOL、BASIC 等。随着计算机理论的发展，又出现了结构化程序设计语言，如 Pascal 等。而 20 世纪 80 年代在出现了以 C++ 语言为代表的面向对象的程序设计语言，而 90 年代则更发展出了 Java 这一网络化编程语言。计算机语言的发展为计算机普及和应用做出了重大的贡献，而 C 语言则是所有高级语言中最具有代表性的一种，C++、Java、C# 等语言都是在其基础上发展而来的。因此学好 C 语言是学好后续程序开发语言的基础。