



普通高等教育
软件工程

“十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

高等学校精品资源共享课程（省级）主讲教材
高等学校教学成果一等奖（省级）建设成果

高级语言程序设计

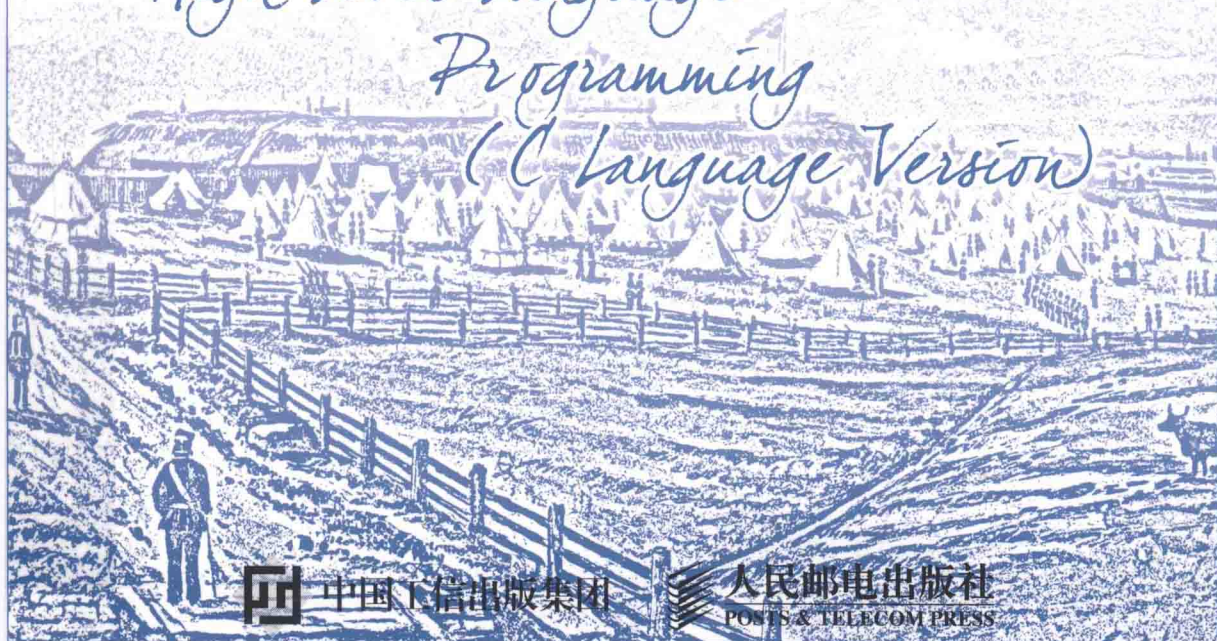
（C语言版）

——基于计算思维能力培养

揭安全 ◎ 著

王明文 ◎ 主审

*High-level Language
Programming
(C Language Version)*



中国通信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



普通高等教育
软件工程

12th Five-Year Plan Textbooks
of Software Engineering

课程（省级）主讲教材
一等奖（省级）建设成果

高级语言程序设计

（C语言版）

——基于计算思维能力培养

揭安全 ◎ 著

王明文 ◎ 主审

*High-level Language
Programming
(C Language Version)*

人民邮电出版社

北京



图书在版编目 (C I P) 数据

高级语言程序设计 : C语言版 : 基于计算思维能力
培养 / 揭安全著. — 北京 : 人民邮电出版社, 2015.8
普通高等教育软件工程“十二五”规划教材
ISBN 978-7-115-39460-6

I. ①高… II. ①揭… 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 (2015) 第136426号

内 容 提 要

本书是一本以 C 语言为描述语言, 介绍结构化程序设计方法的教材。全书共 10 章, 内容包括: 程序设计引论, C 数据类型、运算符与表达式, 算法与简单 C 语言程序设计, 程序基本控制结构, 函数及其应用, 数组及其应用, 指针及其应用, 结构体及其应用, 文件与数据存储, C 综合性程序设计案例分析。

全书理论联系实际, 将 C 语言的语法融入问题求解方法的学习中。本书案例选取贴近生活, 通俗易懂, 紧密联系应用实践, 易于激发读者的学习兴趣; 内容组织由浅入深, 重点突出, 例题与实验设计举一反三, 采用图文并茂的方式来解析教学重点与难点, 使读者对知识点的学习不但知其然, 而且知其所以然; 教学设计符合信息化学习模式的需求, 循序渐进地介绍模块化程序设计方法, 潜移默化地提高读者问题求解与结构化程序设计的能力, 培养计算思维能力。

本书适合作为高等学校计算机相关专业的程序设计基础课程教材, 也可作为非计算机专业 C 语言程序设计课程教材, 还可供从事计算机相关工作的科技人员、程序设计爱好者及各类自学人员参考。

-
- ◆ 著 揭安全
 - 主 审 王明文
 - 责任编辑 邹文波
 - 责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京中新伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
 - 印张: 23 2015 年 8 月第 1 版
 - 字数: 605 千字 2015 年 8 月北京第 1 次印刷
-

定价: 48.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前言

现在人类社会已进入到运行在软件上的信息化时代,这个时代背景下的软件开发涉及 PC 应用程序、工程计算和嵌入式系统、信息管理和移动互联网等众多领域。程序设计是从事软件开发工作的必备技能,程序设计的大部分工作是在寻找和优化问题的解决方案,它是一种创造性与高智力劳动。学习程序设计本身是一项具有挑战性的工作。“高级语言程序设计”(“程序设计基础”、“C 语言程序设计”)作为大学计算机相关专业的第一门程序设计课程,承担了培养学生解决问题与程序设计的基本能力的任务。然而,一些学生在学习这门课程的过程中,过度纠结于程序设计语言的语法细节,忽略了程序设计方法的学习,完全没有体会到程序设计的乐趣;还有些同学甚至对学习程序设计产生恐惧心理,从而导致专业兴趣全无。因此,作为第一门程序设计课程,如何培养学生的程序设计学习兴趣至关重要。

早在 2006 年,美国 Carnegie Mellon 大学的 Jeannette M. Wing 教授就提出了 Computational Thinking (计算思维)的概念,她认为计算思维就是运用计算机科学的知识分析问题、设计方案、解决实际问题的思维活动。近年来,计算思维引起了国内外计算机教育界的广泛关注,将计算思维贯穿于理论教学和实践应用,将有助于促进知识向能力的转化。计算思维的提出,对程序设计教学提出了更高的要求。因此,“高级语言程序设计”作为计算机相关专业的核心课程,同时还承担着培养学生计算思维能力的重任。

岁月如梭,今年正好是作者从事高等学校计算机教育的第十八年。十八年来的学习和教学经历让我体会到,一本“好”的程序设计书对学习者的严重程度。“若能回到二十多年前,我需要一本什么样的程序设计书来开启我学习程序设计的大门呢?这本书不应只是一本程序设计语言的语法手册,也不应只是包含大量编程技巧而难以理解的程序代码的堆积,它应该是真正教授程序设计方法和思维的,应该通俗易懂,而又不乏兴趣的……”带着这些思考,写一本程序设计书的想法在脑海里渐渐呈现。参加工作以来,我曾担任了八年的全国计算机等级考试(NCRE)考务工作,目睹过很多不得要领而无法通过全国计算机等级考试的学生。那时真想将自己积累的一点程序设计与教学经验融入一本教材,来帮助他们轻松学习程序设计,但终因时间问题,最终没有成文。近几年,在承担教学研究课题的过程中,我也在反思如何通过程序设计课程来培养学生的计算思维能力,如何帮助新步入教坛的年轻教师快速提升教学设计能力。这些思考让我决定着手写这本书的想法更加清晰。2014 年我有机会到美国 California State University Fullerton 进行访学,近距离接触了美国高校计算机教育,并有机会将自己多年来的想法整理成文,于是一年后有了大家手中的这本书。

任何程序设计都是基于一种程序设计语言的,在程序设计语言百花齐放的今天,已没有哪一种语言能够包打天下。众所周知,目前许多主流的程序设计语言都是在 C 语言基础上发展起来的。本书以 C 语言作为描述语言,以应用为导向,以

程序设计方法教学为主线，力图通过课程学习，使读者能将普适的程序设计方法应用于实际的问题求解。这也正是本书取名《高级语言程序设计（C 语言版）——基于计算思维能力培养》而非《C 程序设计语言》的原因了。

本书共 10 章，内容包括：程序设计引论，C 数据类型、运算符与表达式，算法与简单 C 语言程序设计，程序基本控制结构，函数及其应用，数组及其应用，指针及其应用，结构体及其应用，文件与数据存储，C 综合性程序设计案例分析。

概括起来，全书有以下特点。

（1）理论联系实际，强化计算思维能力的培养。

本书语言语法介绍以够用、实用和应用为原则，将 C 语言的语法融入问题求解中；从实际应用案例中抽取教学要素，重点强化模块化程序设计方法与基本算法的学习。全书从数据组织的维度介绍了基本数据类型以及数组、指针和结构体等构造数据类型在数据处理中的应用；从算法维度将“迭代”“穷举”“递归”“分治”“检索”与“排序”等算法融入实际应用问题的求解过程。让读者在学习的过程中潜移默化地提高计算思维能力。

（2）案例选取贴近生活，有助于提高学习兴趣。

全书通过贴近生活的案例来分析问题的本质，如程序设计语言为何要区分不同的数据类型，如何在程序设计中选择正确的数据类型，如何存储大规模数据等，书中用了通俗易懂的例子来进行说明。同时，从读者熟悉的应用软件中抽取教学案例，如网银认证的验证码，信息加密，计算器的进位制转换，手机通信录查询等案例都贴近生活，突出了应用导向，有助于提高学生学习兴趣。

（3）内容呈现直观、形象，知识点讲解深入浅出，通俗易懂。

全书以图文并茂的方式深入剖析相关知识的底层原理，使读者对课程难点做到不但知其然，而且知其所以然。例如，通过递归调用图来说明递归程序的执行原理；通过详细的图示来说明指针、参数传递、链表等难点，并拓展了指针在生成动态不规则二维数组等方面的高级应用，这可为学生今后深入理解 Java 等新型程序设计语言的引用数据类型奠定基础。

（4）内容编排体现“以学为中心”的教学思想。

随着大规模在线教育课程（MOOC）和微课等新型教学媒体的出现，学生的学习呈现碎片化学习特点。本书在内容组织上进行了全新设计，将精选的相关练习与实验穿插到章节中间，题型涵盖全国计算机等级考试二级 C 语言程序设计全部题型，使读者在学习完相关内容后能够及时巩固并拓展所学知识，做到举一反三，满足“翻转课堂”等新型教学范式的教学需求。全书还以小贴士的形式提供了大量的相关拓展知识，以开阔读者的视野。

（5）C 标准与时俱进，程序代码规范统一。

教材内容一方面符合当前全国计算机等级考试二级 C 语言程序设计考试大纲的要求，同时还拓展了部分 C99 与 C11 标准，以提升编写 C 语言程序的灵活性。例如，通过引入 bool 类型，使逻辑表达更加简洁，符合现代程序设计的规范。从第 5 章开始，大部分问题求解都以模块化的方式进行设计，有助于引导学生掌握模块化程序设计的思想。书中所有程序源代码均通过了调试。

（6）注重实践环节，设计了多层次实验体系。

正如我们不可能只通过书本知识来获得驾驶和游泳技能一样，程序设计实践是学习程序设计的最佳途径。为此，本书设计了验证型、设计型和综合设计型多层次实验体系，从多维度强化实践环节。建议读者在学习完每章节知识点后，完成相应的练习和实验题，在实践中达到提高程序设计能力的目的。作为本书的重要特色，在第 5 章和第 10 章分别提供了两个阶段性的综合设计案例。其中第 10 章详细分析了一个“基于用户角色的图书管理系统”的设计与实现方法，详细说明

了需求方案、设计目标、设计任务、模块划分、功能实现等环节的设计方法,将程序编写和软件工程原理的阐述有机地贯穿在一起,起到事半功倍的效果,可为学校开展课程设计提供良好的借鉴。

(7) 提供了丰富的辅助教学资源。

面向教师提供了精心设计的教学课件、程序源代码、实验案例、实验指导、习题解答、实验参考答案、测试样卷、重点难点解决办法及教学设计模板等资料。其中教学课件应用了动画、仿真等形式突破教学难点,可有效提升课堂教学效果,实验案例可直接应用于与课程同步的实验教学。

面向学生提供了 Visual C++6.0、Code::Blocks 集成开发工具的使用详解、程序源代码、实验案例、教学课件、全国计算机等级考试二级 C 语言程序设计模拟题、实验报告模板、课程设计报告模板等资料,可为学习者提供学习指导。第 10 章的综合课程设计案例源代码提供了分阶段的项目源代码(共 10 个阶段),可使读者理解在软件工程思想指导下的渐进式项目开发过程。上述资料都可以从出版社网站下载(www.ptpress.com.cn)或向编者发送邮件索取。

全书由揭安全著,王明文教授在百忙之中仔细审阅了全部初稿,并提出了许多宝贵的意见和建议,在此表示衷心的感谢!

本书可作为大学计算机科学与技术、软件工程、网络工程、信息管理、电子商务、物联网工程等专业“高级语言程序设计”“程序设计基础”课程的教材,也可作为大学公共计算机基础——“C 语言程序设计”课程的教材,内容符合全国计算机等级考试二级 C 语言程序设计大纲要求,还可供参加全国计算机等级考试(NCRE)的同学或从事 C 语言程序开发的工程技术人员参考。

2004 年本人参编的《数据结构》(C 语言版)(第 1 版)在人民邮电出版社出版,2012 年该书第 2 版被评为普通高等学校“十二五”国家级本科规划教材,现已出版第 3 版。在本书撰写、校审及出版过程中,人民邮电出版社给予了大力的支持,在此一并表示衷心的感谢!

因作者水平有限,书难免存在不妥之处,欢迎读者给我发送邮件,对本书和教学辅助资源提出意见和建议。编者 E-mail: jieanquan@163.com、jaq@jxnu.edu.cn。

作 者

2015 年 4 月

学习与教学建议

章名	基本内容	教学建议	45 学时	60 学时
第1章 程序设计引论	计算科学与问题求解★ 程序、程序设计及程序设计语言 C 语言简介 C 语言程序开发工具与程序开发步骤★	本章内容重在引导,使学生理解计算科学与问题求解的基本概念,了解程序、程序设计与程序设计语言的基本概念,了解 C 语言的历史及其应用领域。掌握 C 语言程序常用的开发工具与 C 语言程序开发步骤。	2 学时	2 学时
第2章 数据类型、运算符与表达式	数据类型的概念与分类★ 常量与变量★ 算术运算符★ 关系运算符★ 赋值运算符★ ++和--运算符★ 表达式的类型转换	本章的重点是理解数据类型的概念及分类,各种数据类型变量的表示范围。 正确使用常量与变量,掌握算术运算、关系运算、赋值运算和++、--等常用运算符的使用方法,并理解运算符的优先级与结合性。	3 学时	3 学时
第3章 算法与简单 C 语言程序设计	字符输入/输出 格式输入/输出★ 算法的概念及其描述方法★ 顺序程序设计举例★	本章的重点是掌握键盘数据输入、输出函数的使用方法。 理解算法在问题求解中的作用,了解算法的表示方法。 掌握简单顺序程序设计的一般方法。	2 学时	3 学时
第4章 程序基本控制结构	逻辑运算符与逻辑表达式★ if 分支语句★ 条件表达式 switch 语句★☆ while 循环语句★☆ for 循环语句★☆ do while 循环语句★ break 语句★☆ continue 语句★☆ 多重循环及其应用★☆ 循环程序设计方法★☆	本章是全书的重点之一。 要求在正确掌握分支与循环控制语句的基础上学习分支与循环程序设计方法。 重点掌握计数法、标记法等循环控制技术,并通过数列求和、素数判断等典型例题介绍迭代、递推、穷举等循环程序设计技巧。 引导学生学会用程序调试工具分析程序执行情况。同时正确理解 break 与 continue 等程序跳转语句的使用场合。 通过输出九九乘法表等案例介绍多重循环程序的设计方法。掌握利用分支与循环进行问题求解的方法,培养计算思维能力。	11 学时	14 学时
第5章 函数及其应用	函数的定义和调用★ 库函数与自定义函数◎ 函数的参数传递方式★☆ 函数的嵌套调用★ 递归函数★☆ 变量的作用域与生存期 函数综合应用举例◎	本章是全书的重点之一。 要求学生领会函数的定义方法,有参函数与无参函数,有返回值与无返回值函数的使用场合。 重点理解函数的参数传递方式,通过汉诺塔等典型问题介绍递归程序设计方法。 理解模块化程序设计的基本思想,并能在实际中熟练应用。	6 学时	7 学时

章名	基本内容	教学建议	45 学时	60 学时
第6章 数组及其应用	一维数组的定义与使用★	本章是全书的重点之一。	10 学时	14 学时
	向函数传递一维数组★☆	要求重点掌握如何应用数组来组织大规模数据，并掌握基于数组的数据插入、删除、查找、排序（选择与冒泡）等常用数据处理算法，并能熟练应用。		
	基于数组的常用算法及其应用★☆	理解数组作为函数参数的使用方法，掌握字符串及常用的字符串函数的使用方法。		
	二维数组★ 向函数传递二维数组★☆ 字符串及字符数组★ 常用的字符串函数 基于数组的递归算法◎☆	通过基于数组的递归算法进一步巩固递归程序设计技巧。		
第7章 指针及其应用	指针变量的定义与初始化	本章是全书的重点与难点。	6 学时	8 学时
	间接寻址运算符★	要求学生理解指针的本质，理解指针的定义与使用方法。重点掌握指针在作为函数参数、访问一维数组、访问字符串等三个方面的应用。		
	指针与函数★ 指针和一维数组★ 字符指针★ 行指针与列指针★☆ 利用列指针设计通用二维数组处理函数◎☆ 指针数组◎	行指针与列指针是本章的难点，需重点讲授。引导学生了解指针在动态内存分配等方面的应用。		
	动态内存分配 二级指针与指向函数的指针◎	60 学时的教学计划还要求学生掌握二级指针、指针数组的定义与使用方法，为“数据结构”等后续课程的学习奠定基础。		
第8章 结构体及其应用	结构体类型与结构体变量的定义与使用★	本章的教学重点是引导学生掌握如何利用结构体（数组）存储大规模复杂数据对象，并编写相应的数据处理算法。	4 学时	5 学时
	指向结构体的指针★	要求学生掌握结构体类型及变量的定义，并理解通过指针引用结构体变量的方法。		
	向函数传递结构体 结构体数组★ 单链表的定义◎ 基于单链表的查找、插入和删除等基本算法◎☆	理解如何应用动态内存分配构造单链表，理解基于单链表的基本算法。		
第9章 文件与数据存储	文件的分类、文件指针	本章的学习重点是理解为何要用文件来存储数据，如何应用文件存储数据。 掌握两种文件的数据读写方法，重点讲授格式化读/写函数的使用方法。可引导学生通过编程实践来熟悉文件的读写操作。	1 学时	2 学时
	文件的打开与关闭★			
	文件检测函数 字符读/写函数 字符串读/写函数 格式化读/写函数★ 数据块读/写函数 文件的随机读写◎☆			

续表

章名	基本内容	教学建议	45 学时	60 学时
第 10 章 C 综合性程序设计案例分析	软件开发过程概述◎ C 综合性程序设计案例分析◎	对非计算机相关专业本章不作要求。 对计算机相关专业教师可以用 2 学时讲授如何结合软件工程知识，利用所学程序设计方法来开展综合性课程设计。 本章案例可供开展课程设计的学校师生参考。	0 学时	2 学时

说明：

★为重点内容，☆为难点内容，◎为 45 学时的教学计划自学内容。前 9 章涵盖全国计算机等级考试二级 C 语言程序设计大纲内容。表中所列学时为理论学时，建议同步实验为 30 学时，另应辅助 30 学时课外实验，若开展课程设计则另需 20 学时课外实验。

目 录

前言	I
----	---

学习与教学建议	V
---------	---

第1章 程序设计引论	1
------------	---

1.1 计算科学与问题求解	1
1.2 程序与程序设计	2
1.2.1 程序	2
1.2.2 程序设计	3
1.3 程序设计语言	4
1.3.1 程序设计语言分类	4
1.3.2 程序的执行方式	5
1.4 C语言简介	7
1.5 C语言程序开发工具	9
1.6 C语言程序开发步骤	14
本章小结	16
实验一	16

第2章 数据类型、运算符与表达式	18
------------------	----

2.1 C程序基本结构	18
2.2 C程序中常见的符号	20
2.3 数据类型	21
2.4 常量	25
2.4.1 整型常量	25
2.4.2 实型常量	26
2.4.3 字符常量	26
2.4.4 字符串常量	28
2.4.5 宏常量	28
练习2.1	28
2.5 变量	29
2.5.1 变量的声明	29

2.5.2 变量初始化	30
2.5.3 变量的访问与使用	31
练习2.2	32
2.6 运算符与表达式	33
2.6.1 算术运算符	33
2.6.2 运算符的优先级与结合性	34
2.6.3 关系运算符	35
2.6.4 复合赋值运算符	36
2.6.5 ++和--运算符	37
2.7 表达式的类型转换	38
2.7.1 赋值表达式的类型转换	38
2.7.2 强制类型转换	39
2.8 const 常量	40
练习2.3	41
本章小结	41
实验二	42

第3章 算法与简单C语言程序设计	43
------------------	----

3.1 C语句	43
3.2 C程序输入/输出操作的实现	44
3.2.1 字符输入/输出	44
3.2.2 格式输入/输出	46
练习3.1	52
3.3 算法	52
3.3.1 问题求解过程中算法的作用	52
3.3.2 算法的特点	54
3.3.3 算法的描述	55
3.4 简单程序设计举例	60
练习3.2	63
本章小结	63
实验三	63

第4章 程序基本控制结构.....64

4.1 逻辑运算符与逻辑表达式.....64

4.1.1 逻辑运算符.....64

4.1.2 逻辑表达式.....65

练习 4.1.....66

4.2 选择控制结构.....67

4.2.1 if 单分支语句.....67

4.2.2 if 双分支语句.....73

4.2.3 if 多分支语句.....74

4.3 条件表达式.....78

练习 4.2.....78

4.4 switch 多分支语句.....79

练习 4.3.....84

4.5 循环控制语句.....84

4.5.1 while 循环语句.....84

4.5.2 for 循环语句.....89

4.5.3 do while 循环语句.....94

练习 4.4.....95

4.6 程序跳转语句.....96

4.6.1 break 语句.....96

4.6.2 continue 语句.....98

4.6.3 goto 语句.....99

练习 4.5.....100

4.7 多重循环及其应用.....101

练习 4.6.....104

4.8 循环程序设计方法.....105

4.8.1 迭代法.....105

4.8.2 穷举法.....107

本章小结.....110

实验四.....110

第5章 函数及其应用.....112

5.1 C 函数概述.....112

5.2 C 语言函数的定义和调用.....113

5.2.1 C 语言函数的定义.....113

5.2.2 return 语句.....115

5.2.3 函数调用.....116

5.2.4 函数声明.....118

练习 5.1.....119

5.3 引用库函数与自定义函数.....120

5.3.1 库函数分类.....120

5.3.2 #include 指令.....120

5.4 函数参数传递方式.....121

练习 5.2.....123

5.5 函数嵌套调用.....124

5.5.1 嵌套调用的概念.....124

5.5.2 模块化设计基本方法.....124

练习 5.3.....128

5.6 递归函数及其应用.....129

5.6.1 递归的概念.....129

5.6.2 递归程序分析.....131

练习 5.4.....133

5.6.3 递归程序应用.....134

练习 5.5.....137

5.7 变量的作用域与生存期.....138

5.7.1 局部变量.....138

5.7.2 全局变量.....139

5.7.3 变量的存储类型.....141

练习 5.6.....144

5.8 函数综合应用——趣味算术游戏.....145

练习 5.7.....152

本章小结.....152

实验五.....152

第6章 数组及其应用.....153

6.1 一维数组.....153

6.1.1 一维数组的定义与引用.....153

6.1.2 一维数组的初始化.....156

练习 6.1.....158

6.2 向函数传递一维数组.....159

练习 6.2.....163

6.3 基于数组的常用算法及其应用.....164

6.3.1 顺序查找.....164

6.3.2 数据删除.....165

6.3.3 数据插入.....166

6.3.4 寻找最大值.....167

练习 6.3.....169

6.3.5 数据排序.....169

6.3.6 数据倒置.....174

6.3.7 二分查找	176	7.7 指针和二维数组	230
6.3.8 一维数组应用实例	178	7.7.1 列指针	230
练习 6.4	180	7.7.2 行指针	231
6.4 二维数组	181	7.7.3 深入理解二维数组作函数参数的本质*	233
6.4.1 二维数组的定义、引用及初始化	182	7.7.4 巧用列指针设计通用二维数组处理函数*	233
6.4.2 二维数组应用实例	184	练习 7.5	235
练习 6.5	187	7.8 指针的高级应用	236
6.5 向函数传递二维数组	188	7.8.1 指针数组及其应用	236
练习 6.6	191	7.8.2 动态内存分配	240
6.6 字符串及字符数组	191	7.8.3 二级指针*	243
6.6.1 字符串	191	7.8.4 指向函数的指针及其应用*	246
6.6.2 字符数组的初始化	191	练习 7.6	248
6.6.3 字符数组的输入/输出	192	本章小结	249
6.6.4 字符串处理函数	194	实验七	249
6.6.5 字符串应用实例	197	第 8 章 结构体及其应用	250
练习 6.7	202	8.1 为何要用结构体	250
6.7 基于数组的递归算法	203	8.2 结构体类型与结构体变量	250
练习 6.8	205	8.2.1 结构体类型的声明	250
本章小结	205	8.2.2 结构体变量的定义	252
实验六	206	8.2.3 对结构体变量的操作	253
第 7 章 指针及其应用	207	8.2.4 结构体变量的初始化	254
7.1 指针的本质	207	练习 8.1	255
7.2 指针变量的定义与初始化	208	8.3 指向结构体的指针	256
7.3 间接寻址运算符	210	练习 8.2	257
练习 7.1	211	8.4 向函数传递结构体	257
7.4 指针与函数	212	8.4.1 值传递	257
7.4.1 传值调用与传地址调用	212	8.4.2 地址传递	258
7.4.2 指针作函数参数的应用实例	214	练习 8.3	259
练习 7.2	217	8.5 结构体数组	260
7.5 指针和一维数组	218	8.5.1 结构体数组的定义	260
7.5.1 指针的算术运算与关系运算	218	8.5.2 结构体数组的初始化与引用	260
7.5.2 应用指针访问一维数组	220	8.5.3 结构体数组的应用	263
7.5.3 深入理解一维数组作函数参数的本质	222	练习 8.4	266
练习 7.3	224	8.6 动态数据结构——单链表	267
7.6 字符指针	225	8.6.1 单链表的定义	267
7.6.1 使用字符指针指示字符串常量	225	8.6.2 在单链表插入新结点	268
7.6.2 利用字符指针访问字符串变量	227	8.6.3 建立单链表	270
练习 7.4	229		

8.6.4 单链表的遍历	272
8.6.5 在单链表中查找结点	273
8.6.6 在单链表中删除结点	273
练习 8.5	276
本章小结	277
实验八	277

第 9 章 文件与数据存储

9.1 引例——学生文件信息的创建	278
9.2 文件概述	279
9.2.1 流的概念	279
9.2.2 文件的分类	279
9.2.3 文件指针	281
9.3 文件的打开和关闭	281
9.3.1 文件的打开	281
9.3.2 文件的关闭	283
练习 9.1	284
9.4 文件检测函数	284
9.4.1 检测文件末尾函数	284
9.4.2 检测出错函数	284
9.5 文件的读/写操作	285
9.5.1 字符读/写函数	285
9.5.2 字符读/写函数应用实例	285
9.5.3 字符串读/写函数	286
9.5.4 字符串读/写函数应用实例	287
9.5.5 格式化读/写函数	287
9.5.6 格式化读/写函数应用实例	288
9.5.7 数据块读/写函数	290
9.5.8 数据块读/写函数应用实例	290
练习 9.2	292
9.6 文件的随机读写	293
9.6.1 文件的定位	294
9.6.2 文件随机读写应用实例	295
练习 9.3	299
9.7 利用位运算对文件数据加密*	299
9.7.1 位运算	299
9.7.2 数据文件加密实例	301

本章小结	303
实验九	303

第 10 章 C 综合性程序设计案例 分析*

10.1 软件开发过程概述	304
10.2 基于用户角色的图书管理系统案例 分析	306
10.2.1 问题描述与需求分析	307
10.2.2 总体设计与详细设计	307
10.2.3 编码	314
10.2.4 测试与运行效果	338
10.3 课程设计的实施建议与选题设计	340
10.3.1 课程设计实施建议	340
10.3.2 课程设计选题设计	340
本章小结	341
实验十*	341

附录 A C 语言中的关键字

附录 B 常用字符与 ASCII 代码 对照表

附录 C 运算符的优先级和结合性

附录 D 常用的 C 语言库函数

D.1 数学函数	345
D.2 字符函数和字符串函数	346
D.3 动态存储分配函数	347
D.4 其他常用函数	347

附录 E 全国计算机等级考试二级 C 语言程序设计考试大纲

附录 F Visual C++6.0 使用 指南

参考文献

第 1 章

程序设计引论

本章首先介绍计算科学与求解问题的基本特点、程序与程序设计的基本概念、程序设计语言分类、程序的执行方式、C 语言的历史与特点等内容，最后简要介绍 C 语言程序开发工具。

1.1 计算科学与问题求解

维基百科 (Wikipedia) 关于计算机科学 (Computer Science, CS) 的定义是：系统性研究信息与计算的理论基础以及它们在计算机系统中如何实现与应用的实用技术的学科。它通常被形容为对那些创造、描述以及转换信息的算法处理的系统研究。计算机科学包含很多分支领域；有些强调特定结果的计算，如计算机图形学；而有些是探讨计算问题的性质，如计算复杂性理论；还有一些领域专注于怎样实现计算。例如，程序设计是应用特定的编程语言解决特定的计算问题，人机交互则是专注于怎样使计算机和计算变得有用、好用，以及随时随地为人所用。

美国计算机学会 (ACM) 指出“计算机科学是计算机和算法过程的研究，包括它们的原理、它们的硬件和软件设计、它们的应用以及它们对社会的影响”。

从第一台电子计算机诞生到现在的几十年来，计算机科学得到了蓬勃发展和广泛普及，计算机科学正深刻影响着人类的生产方式、认知方式和社会生活方式。尤其是近十年来，随着网络和 WWW 技术的发展，更是极大地丰富了计算机科学的内涵，以至于 ACM/IEEE-CS 的专家们认为，目前已经无法用计算机学科来称谓它，而改称为计算学科 (Computing Discipline)。总体而言，计算机科学的发展和应用水平已是衡量一个国家综合竞争力的重要标志之一，它已成为现代科学体系的主要基石之一。

问题求解是指人们在生产、生活中面对新的问题时，由于缺少现成的有效对策所引起的一种积极寻求问题答案的活动过程。

问题求解是计算科学的根本目的，计算机科学多半也是在问题求解的实践中发展起来的。既可用计算机来求解如数据处理、数值分析等问题，也可用计算机来求解如生物学（如基因测序）、物理学（如研究准原子核的结构）和心理学（如对求解问题的意图和连续性行为的分析）所提出的问题。问题求解是一个非常复杂的思维活动过程，它不仅包括整个认识活动，而且也渗透了许多非智力因素的作用，但思维活动是解决问题的核心成分。

利用计算机进行问题求解前，人们必须给出问题的适当描述，通过抽象将问题模型化并用适当的符号表示出来，然后计算机通过对这些符号实施规定的“计算”完成问题求解，如图 1-1 所示。

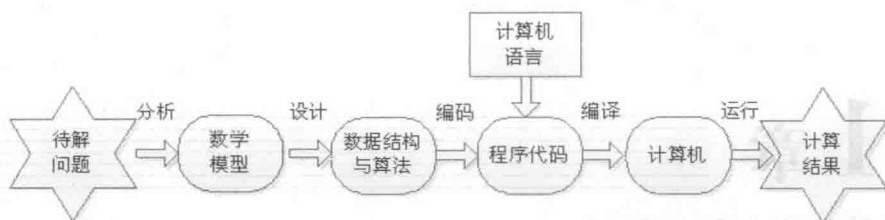


图 1-1 借助于计算机的问题求解过程

图灵奖获得者 Nicklaus Wirth 指出：“在较高的认识层次上，硬件和软件是一样的”。因此它们最终都可以归结为一定形式的数据表示物质世界的某一系统，并使用算法通过对这些数据的变换来获得相应的处理结果。然而，并不是所有的问题都是“可计算”的。所以计算机学科不仅要研究什么可以被有效地自动计算，而且还要研究如何进行有效的计算，因此，可以认为，计算机学科的根本问题是：什么能、且如何被有效地自动计算。

综上所述，尽管关于计算机科学的定义有多种，但它们实质上都强调了算法的研究。

算法是一组明确的、有效可计算操作的有序集合，它能在有限的时间内结束，并产生计算结果。如果我们可以指定一个算法来解决问题，那么我们就可以对该问题进行自动化的求解。计算机算法最终需要通过程序设计来实现，程序设计使计算机科学成为实验学科而非纯理论学科。

一段时期以来，计算机和信息技术被社会看成是一种高科技工具，计算机科学也被构造成一门专业性很强的工具学科，这种社会认知很容易导致负面的狭义工具论。这种狭义的认识是计算机科技向各行各业渗透的最大障碍，对计算机和信息科技的全面普及极其有害。2006 年，美国 Carnegie Mellon 大学计算机科学系前系主任周以真（Jeannette M. Wing）教授在（*Communications of the ACM*）杂志发表了一篇名为《计算思维》（*Computational Thinking*）的文章，在国内外计算机教育界产生了广泛的影响。计算思维是指运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。今天，计算思维应成为一种普适思维，是每个人应具有的基本技能。计算思维强调一切皆可计算，从物理世界模拟到人类社会的模拟，从人类社会模拟再到智能活动，都可以认为是计算的某种形式。将计算思维贯穿于理论教学和实践应用，将有助于促进知识向能力的转化。因此，程序设计的学习也是传播计算思维的一种重要途径。以“程序设计”为载体培养读者计算思维能力正是编者撰写本书的主要目的之一。

1.2 程序与程序设计

1.2.1 程序

“程序”并非计算机专利，其实做任何事情都要讲究程序。《舌尖上的中国》介绍的每道美食都有其特有的制作程序，第二季第 2 集《心传》中讲述的陕西吴堡县张家山镇的空心挂面加工过程给观众留下了深刻印象：和面、搓条、盘条、上筷子、阴条、分筷子、再阴条、出筷子、装封。为获得独特的美味，这些程序缺一不可。

本书中的程序特指计算机程序。计算机程序（Computer Program）是指一组指示计算机或其他具有信息处理能力的装置进行每一步动作的指令。程序通常用某种程序设计语言编写，运行于

某种目标体系结构上。简单来讲, 计算机程序是计算任务的处理对象和处理规则的描述。打个比方, 一个程序就像一个用汉语(语言)写下的红烧肉菜谱(程序), 用于指导懂汉语和烹饪手法的人来做这道菜。计算机程序需要由程序设计语言来编写, 才能直接或间接地被计算机理解并执行。

计算机(硬件)本身并不是一个智能设备, 它需要运行特定的程序才能完成特定的任务。而程序及与之相关的数据和文档则称为软件。软件与硬件相比具有不同的特点, 它既看不见又摸不着。它不仅存在于大家熟悉的计算机、平板电脑及智能手机中, 还广泛被应用于很多普通家用电器或电子设备中, 如 MP3 播放器、微波炉、智能手机、数码相机和汽车等。

为了运行一个程序, 计算机要加载程序代码, 可能还要加载数据, 从而初始化成一个开始状态, 然后调用某种启动机制开始执行程序, 最终将输入的数据经处理后得到运行结果。

例如, 应用智能手机中的全景照相程序可以将连续拍摄的多幅照片自动拼接成全景照片, 如图 1-2 所示, 这里的数据是照片, 照片处理的步骤就是算法。

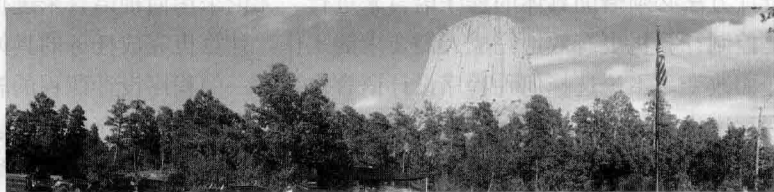


图 1-2 利用全景拍摄程序拍摄的全景照(拍摄于 Devils Tower, USA)

再如, 电波钟表将传统钟表技术与现代时频技术、微电子技术、计算机技术等多项技术相结合, 通过接收国家授时中心以无线电长波传送的标准时间信号, 经过内置微处理器中的程序解码处理后, 自动校准计时器走时, 使电波钟表显示的时间与国家的标准时间自动保持精确同步。如图 1-3 所示(图片来源: <http://www.amazon.com>)。



图 1-3 电波钟表



图 1-4 车载行车电脑

而利用微信、QQ、易信等软件让人与人之间的通信变得方便快捷。利用支付宝软件不仅可以实现电子支付, 还可实现自助理财。电子商务平台及大规模在线教育课程(MOOC)也需要大量软件的支持才能推广与应用。

可见, 无处不在的程序正改变着人们的生活与社会的生产方式。

1.2.2 程序设计

程序设计, 简单地讲就是设计“程序”的过程, 程序设计是创造性劳动, 目前尚不能完全实

现软件自动化^①。专门进行程序设计的人员称为程序员（Programmer），许多计算机专业的毕业生第一份工作就是做程序员。当然程序设计并非程序员的专利，各种软件开发平台操作日趋简单，功能日趋强大，甚至出现了一些积木式的程序开发平台，让普通的民众也可以 DIY 自己的应用程序。一些原本不是程序员的普通人甚至因为开发出某种受欢迎的应用程序（如手机游戏）而一举成名。然而，对于一些结构复杂、规模庞大的软件系统需要专门的设计人员与经验丰富的程序员来共同开发。这正如许多普通人可以做出美味的家常菜，但想成为大厨，则需经过专业的理论学习与实践训练。

如前所述，程序需使用程序设计语言进行编写，从计算机诞生至今，产生了成千上万种程序设计语言和各式各样的程序设计方法，从机器语言到汇编语言，从汇编语言到 FORTRAN、BASIC、COBOL，再到 C、C++、Java、C#、Python；从结构化程序设计方法到面向对象的程序设计方法；从串行程序设计到并行（并发）、多核程序设计等。

学习程序设计方法必须借助具体的程序语言来进行，无论采用何种语言来进行程序设计，其最终的目的都是控制计算机更有效地完成人们要求的工作。计算机完成任务的具体算法不因程序设计语言的变化而改变。不论选择何种程序设计语言作为第一门程序设计课程的教学语言，学习者首先都要掌握程序设计的基本方法和规则，程序设计语言只是实现程序设计的表现手段。只有真正掌握了程序设计的基本方法，培养了计算思维能力，才能够做到触类旁通，举一反三。

1.3 程序设计语言

1.3.1 程序设计语言分类

1. 机器语言

计算机能够直接识别的程序设计语言为机器语言，一条机器指令是由一个或多个字节组成的二进制编码。指令中一般包括操作码和地址码，操作码用于指示该指令的性质，如加法、减法等；地址码用于指示该指令操作的对象，如减法指令需要告诉 CPU 被减数与减数在内存的位置，这样 CPU 才能从指定的内存中读取数据并进行相应的计算。

每一种 CPU 都具有自己特定的机器语言，例如，在 8086/8088CPU 中机器指令：

```
10111000 00000000 0001000
```

表示将十六进制数 1000H 存入 CPU 内部的名为 AX 的寄存器。

```
00000001 11011000
```

表示将寄存器 AX 与寄存器 BX 中的内容相加，结果存回 AX。

用机器指令编写的程序可以直接被机器执行，但用机器语言编写程序很烦琐，程序冗长，而且还特别容易出错。此外，由于它与机器硬件密切相关，所以用机器语言编写的程序不能在异种机型间移植。

2. 汇编语言

“汇编语言”用指令助记符来代替机器指令中的操作码与操作数。例如，前面两条机器指令用

^① 软件自动化是尽可能借助计算机系统（特别是自动程序设计系统）进行软件开发的过程。按狭义的理解，软件自动化是从形式的软件功能规格说明到可执行的程序代码这一过程的自动化。