



普通高等教育“十一五”国家级规划教材

2009年度普通高等教育精品教材
第2版为

丛书主编 谭浩强

高等院校计算机应用技术规划教材

应用型教材系列



C语言 程序设计 (第3版)

谭浩强 著

根据“中国高等院校计算机基础教育课程体系”组织编写

清华大学出版社



普通高等教育“十一五”国家级规划教材

丛书主编 谭浩强

高等院校计算机应用技术规划教材

应用型教材系列

C语言程序设计

(第3版)

谭浩强 著

清华大学出版社
北京

内 容 简 介

谭浩强教授所著的《C 程序设计》一书受到国内专家和读者的一致好评,被公认为学习 C 语言程序设计的优秀教材,被全国大多数高校选用,是学习 C 语言的主流用书。已累计发行了 1250 多万册,创同类书的全国发行最高纪录。

针对我国应用型大学的实际情况,谭浩强教授于 21 世纪初在《C 程序设计》一书的基础上编写出版了《C 语言程序设计》,降低起点,精简内容,通俗易懂,突出重点,紧扣基本要求,使读者容易学习。该书出版后受到广泛欢迎,十多年已出了两版,重印 50 多次。本书是在《C 语言程序设计(第 2 版)》的基础上修订再版的。

在这次修订中,保持了原书概念清晰、通俗易懂的特点,同时根据 C99 新标准介绍程序设计,体现教材的先进性和规范性,并且更加容易学习与入门。本书定位准确,体系科学,内容适当、取舍合理、例题丰富,分析透彻。作者采用的“提出问题—解决问题—归纳分析”的三部曲,符合初学者的认知规律,取得很好的效果。

凡具有计算机初步知识的读者都能读懂本书。本书可作为应用型大学各专业知识学习 C 语言程序设计的教材,基础较好的高职高专也可选用,本书也是不可多得的用于自学的好教材。

本书还配有辅助教材《C 语言程序设计(第 3 版)学习辅导》。

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/谭浩强著. —3 版. —北京:清华大学出版社,2014

高等院校计算机应用技术规划教材·应用型教材系列

ISBN 978-7-302-36964-6

I. ①C… II. ①谭… 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(2014)第 135565 号

责任编辑:谢 琛

封面设计:常雪影

责任校对:梁 毅

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:清华大学印刷厂

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:22.5

字 数:557 千字

版 次:2000 年 1 月第 1 版 2014 年 9 月第 3 版

印 次:2014 年 9 月第 1 次印刷

印 数:1~5000

定 价:44.80 元

编辑委员会

《高等院校计算机应用技术规划教材》

主 任 谭浩强

副 主 任 焦金生 陈 明 丁桂芝

委 员 王智广 孔令德 刘 星 刘荫铭
安志远 安淑芝 李文英 李叶紫
李 琳 李雁翎 宋 红 陈 强
邵丽萍 尚晓航 张 玲 侯冬梅
郝 玲 赵丰年 秦建中 袁 玫
訾秀玲 薛淑斌 谢树煜 谢 琛



序

《高等院校计算机应用技术规划教材》

进入 21 世纪,计算机成为人类常用的现代工具,每一个有文化的人都应当了解计算机,学会使用计算机来处理各种的事务。

学习计算机知识有两种不同的方法:一种是侧重理论知识的学习,从原理入手,注重理论和概念;另一种是侧重于应用的学习,从实际入手,注重掌握其应用的方法和技能。不同的人应根据其具体情况选择不同的学习方法。对大多数人来说,计算机是作为一种工具来使用的,应当以应用为目的、以应用为出发点。对于应用性人才来说,显然应当采用后一种学习方法,根据当前和今后的需要,选择学习的内容,围绕应用进行学习。

学习计算机应用知识,并不排斥学习必要的基础理论知识,要处理好这二者的关系。在学习过程中,有两种不同的学习模式:一种是金字塔模型,亦称为建筑模型,强调基础宽厚,先系统学习理论知识,打好基础以后再联系实际应用;另一种是生物模型,植物中的树并不是先长好树根再长树干,长好树干才长树冠,而是树根、树干和树冠同步生长的。对计算机应用性人才教育来说,应该采用生物模型,随着应用的发展,不断学习和扩展有关的理论知识,而不是孤立地、无目的地学习理论知识。

传统的理论课程采用以下的三部曲:提出概念—解释概念—举例说明,这适合前面第一种侧重知识的学习方法。对于侧重应用的学习者,我们提倡新的三部曲:提出问题—解决问题—归纳分析。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。这种教学方法更适合于应用型人才。

检查学习好坏的标准,不是“知道不知道”,而是“会用不会用”,学习的主要目的在于应用。因此希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而可以指定学生通过上机掌握这些内容。这样做可以培养学生的自学能力,启发学生的求知欲望。

全国高等院校计算机基础教育研究会历来倡导计算机基础教育必须坚持

面向应用的正确方向,要求构建以应用为中心的课程体系,大力推广新的教学三部曲,这是十分重要的指导思想,这些思想在《中国高等院校计算机基础课程》中作了充分的说明。本丛书完全符合并积极贯彻全国高等院校计算机基础教育研究会的指导思想,按照《中国高等院校计算机基础教育课程体系》组织编写。

这套《高等院校计算机应用技术规划教材》是根据广大应用型本科和高职高专院校的迫切需要而精心组织的,其中包括4个系列:

(1) 基础教材系列。该系列主要涵盖了计算机公共基础课程的教材。

(2) 应用型教材系列。适合作为培养应用性人才的本科院校和基础较好、要求较高的高职高专学校的主干教材。

(3) 实用技术教材系列。针对应用型院校和高职高专院校所需掌握的技能技术编写的教材。

(4) 实训教材系列。应用型本科院校和高职高专院校都可以选用这类实训教材。其特点是侧重实践环节,通过实践(而不是通过理论讲授)去获取知识,掌握应用。这是教学改革的一个重要方面。

本套教材是从1999年开始出版的,根据教学的需要和读者的意见,几年来多次修改完善,选题不断扩展,内容日益丰富,先后出版了60多种教材和参考书,范围包括计算机专业和非计算机专业的教材和参考书;必修课教材、选修课教材和自学参考的教材。不同专业可以从中选择所需要的部分。

为了保证教材的质量,我们遴选了有丰富教学经验的高校优秀教师分别作为本丛书各教材的作者,这些老师长期从事计算机的教学工作,对应用型的教学特点有较多的研究和实践经验。由于指导思想明确、作者水平较高,教材针对性强,质量较高,本丛书问世9年来,愈来愈得到各校师生的欢迎和好评,至今已发行了240多万册,是国内应用型高校的主流教材之一。2006年被教育部评为普通高等教育“十一五”国家级规划教材,向全国推荐。

由于我国的计算机应用技术教育正在蓬勃发展,许多问题有待深入讨论,新的经验也会层出不穷,我们会根据需要不断丰富本丛书的内容,扩充丛书的选题,以满足各校教学的需要。

本丛书肯定会有不足之处,请专家和读者不吝指正。

全国高等院校计算机基础教育研究会会长
《高等院校计算机应用技术规划教材》主编

谭浩强

2008年5月1日于北京清华园

前言

程序设计是计算机工作者的一项基本功。C语言是在国内外广泛学习和使用的一种计算机语言,受到广大计算机爱好者的喜爱。我国几乎所有的理工科大学都开设了C语言程序设计课程。

为了使C语言程序设计课程教学取得更好的效果,需要弄清楚几个问题:

1. 为什么要学习程序设计

计算机的本质是“程序的机器”,程序和指令的思想是计算机系统中最基本的概念。只有懂得程序设计,懂得计算机是怎样工作的,才能较深入地理解和应用计算机,才能较好地懂得怎样使计算机按照人们的意图进行工作。学习程序设计,能学到计算机处理问题的方法,培养计算思维,培养分析问题和解决问题的能力。

虽然非计算机专业的大学生,大多数将来并不一定会成为专业的软件开发人员,不一定有许多机会用C语言编写实用程序,但是在大学期间仍然需要学习程序设计。通过学习程序设计,使大学生具有程序设计的初步知识,具有编写程序的初步能力,知道软件开发的过程和特点,有利于各专业人士在今后工作中与程序开发人员沟通合作,进行应用软件的开发工作。因此,大学生(尤其是理工类学生)学习程序设计是很有好处的。

具体地说,学习程序设计的目的有两个方面:(1)学习用计算机解决问题的思路和方法;(2)学习怎样用计算机工具去实现算法,即实现用计算机解题。

2. C语言是基础而实用的语言

进行程序设计,必须用计算机语言作为工具,否则只是纸上谈兵。可供选择的语言很多,各有特点。C语言是基础而实用的语言。并不是每一种语言都具有此特点的,有的语言实用,但不能作为基础语言(如FORTRAN),有的语言可以作为基础,但实际应用不多(如Pascal)。C语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好。既具有高级语言的优点,又具有低级语言的许多特点,既适合于编写系统软件,又能方便地用来编写应用软件。是多年来在国内外使用最广泛的语言。国内外许多专家

认为,C 语言是最基本的通用语言,有了 C 语言的基础,以后过渡到任何一种语言(如 C++ ,C# ,Java)都不会困难。C 语言被认为是计算机专业人员的基本功。

3. 正确处理好算法和语法的关系

C 程序设计课程既有基础的性质(了解计算机处理问题的方式,学习算法),又有应用和工具的性质(掌握语言工具,具有编程的初步能力,能具体应用)。因此,既要注意讲清概念,使学生建立正确的概念,又要培养学生实际处理问题的能力。

程序设计有 4 个要素:(1)算法是程序的灵魂;(2)数据结构是加工的对象;(3)语言是编程工具(算法要通过语言来实现);(4)要采用合适的程序设计方法。程序设计教学是否成功取决于能否将以上 4 个要素(尤其是算法与语言二者)紧密结合。

本书自始至终把这 4 个要素自然有机地结合起来,全面兼顾。算法和语法二者都很重要,都要重视。本书不是孤立地介绍语法,也不是全面系统地介绍算法,更不是根据语言规则的分类和顺序作为教学和教材的章节和顺序,而是从应用的角度出发,以编程为目的和主线,由浅入深地介绍怎样用 C 语言处理问题,把算法和语法紧密结合,同步展开,步步深入。作者精心安排顺序,算法的选择由易而难,细心选择例子,使该读者容易学习。在此基础上,构造了新的教学和教材体系。

具体的做法是:在每一章中,首先举几个简单的例子,引入新的问题;接着介绍怎样利用 C 语言去解决简单的问题;然后再循序渐进地介绍较深入的算法和程序。使学生在富有创意、引人入胜的编程过程中,学会了算法,掌握了语法,领悟了程序设计的思想和方法。把枯燥无味的语法规则变成生动活泼的编程应用。多年的实践表明,这种做法是成功的。

既然要学习编写程序,就离不开计算机语言。如果不掌握语法规则的应用,连最简单的程序也编写不出来,或者错误百出。问题不是要不要掌握语言工具,而是通过什么方法学习和掌握语言工具的使用方法,是枯燥地死记语法规则,还是通过实例掌握语言的使用?初学时可以学习最基本的语法现象而淡化语法的细节,有些细节不必一一讲授,可以让学生自查自学。

建议教师在讲授时,以程序为中心展开,着重讲清解题思路以及怎样用程序去实现它,不要孤立介绍语法规定,教材中叙述的语法规定,可以在介绍编写程序的过程中加以说明,或在作简单介绍后请学生自己阅读,并通过上机实践掌握它。

4. 在学习和应用计算机过程中培养计算思维

计算机基础教育应该面向计算机应用,着眼于提高大学生的计算机应用能力,培养各个领域中的计算机应用人才,同时在教学过程中要注意培养学生

的科学思维和信息素养。计算思维是科学思维的一部分。有专家认为：计算思维是运用计算机科学的基础概念进行问题求解、系统设计和理解人类行为的思维活动。计算思维的本质是抽象和自动化。程序设计中的算法思维就是典型的计算思维。学习程序设计的过程就是培养计算思维的过程。

作者认为：计算思维是在学习和应用计算机的过程中自然而然地培养的，计算思维并不神秘，它是自然而然的结果，而不是生硬进行的过程；是本身存在的内在关联，而不是外界强加的概念。我们要善于在教学过程中有意识地引导学生善于思考，掌握规律，举一反三，养成科学思维（包括计算思维）。

培养计算思维不是最终目的。培养计算思维是为了更好地应用计算机，培养更多更好的计算机应用人才。

5. 要从实际出发，准确定位

在学习 C 语言程序设计的人群中，有不同的情况和要求。有的是计算机专业的，有的是非计算机专业的；有的是本科生，有的是高职生；有的来自研究型大学，有的来自应用技术型大学；有的以后成为计算机专业人员，有的是计算机应用人员；有的要求精通，有的要求粗通，有的则只求一般了解。必须准确定位，从具体实际出发，对不同对象提出不同的要求。不可能要求一个没有计算机基础的人，通过三四十学时的课程就能精通程序设计，掌握复杂的算法，编写出复杂的程序。要求必须恰当，内容要有所取舍。在有限的学时内，只能学习基本的内容，为今后进一步的学习和使用打下必要的基础。如果有少数人将来需要熟练地编程，甚至成为专业的软件开发人员，则应当在学习本课程的基础上继续进行更深入的学习，例如至少应当完成一个综合作业，开发一个具有一定规模的应用软件的实践项目。

本书的对象是应用型大学本科学生，程度较高的高职（大专）学生也可以用本书作为教材。

为了推动 C 程序设计教学，作者于 1991 年在全国率先编著了《C 程序设计》一书，由清华大学出版社出版，向全国发行。各校师生普遍认为该书内容全面而系统、概念清晰而易懂，例题丰富，讲解详尽、包含了许多其他教材中没有的内容，尤其是针对编程实践中容易出现的问题作了分析，被认为是学习 C 语言程序设计的理想教材，至今已出版了四版，重印 180 次。21 世纪初，作者根据不同学校（尤其是应用型大学）的需要，在《C 程序设计》一书的基础上，编写出版了《C 语言程序设计》一书，适当降低要求，精简内容，降低起点，更加易于学习，很受广大师生欢迎。被教育部定为“普通高等教育十一五国家级规划教材”，向全国推荐，至今已出版了两版，重印 50 余次。以上这两本程度不同的教材至今已累计发行超过 1250 万册，成为大学 C 程序设计的主流用书，有力地推动了我国高校的 C 程序设计教学。

本书是在《C 语言程序设计（第 2 版）》的基础上进行修订的。修订中遵循以下原则：

(1) **保持原有特点和风格。**由于作者所著的《C 程序设计》和《C 语言程序设计》内容全面,叙述清晰,通俗易懂,在教学实践中取得了很好的效果,因此在修订时尽量保持原有的优点,基本上保留原书的体系,注意概念准确,讲解透彻,使读者对 C 语言和程序设计有全面、完整的理解。

(2) **按照 C99 标准进行介绍,**以适应 C 语言的发展,使编写程序更加规范。目前国内许多介绍 C 语言的教材是按照 C89 标准介绍的,而国外的同类教材已改为以 C99 标准为蓝本。本书按照 C99 标准的规定介绍程序设计,例如:

① 数据类型中,C99 增加了双长整型(long long int)、复数浮点型(float_complex, double_complex, long long_complex)、布尔型(bool)等。

② 根据 C99 的建议,main 函数的类型一律指定为 int 型,并在函数的末尾加返回语句“return 0;”。

③ C99 增加了注释行的新形式——以双斜线“//”开始的内容作为注释行,这本来是 C++ 的注释行形式,现在 C99 把它扩充进来了,使编程更加方便。同时保留了原来的“/* …… */”形式,以使原来按 C89 标准编写的程序可以不加修改仍可使用。本书采用 C99 的注释新形式,读者使用更方便,而且符合发展需要。因此,本书的程序基本上采用下面的形式:

```
# include < stdio.h>           // 以“//”作为注释行的开始
int main()                     // 指定 main 函数为 int 类型
{
    :
    return 0;                  // 如函数正常执行,返回整数 0
}
```

由于 C99 是在 C89 的基础上增加或扩充一些功能而成的,因此 C89 和 C99 基本上是兼容的。过去用 C89 编写的程序在 C99 环境下仍然可以运行。C99 所增加的许多新的功能和规则,是在编制比较复杂的程序时为方便使用和提高效率而用的,在初学时可以不涉及,因此本书对目前暂时用不到的内容不作介绍,以免读者分心,增加学习难度。在将来进行深入编程时再逐步了解和学习的。

(3) **加强算法,强化解题思路。**在介绍例题时,先进行问题分析,探讨解题思路,构造算法,然后才是根据算法编写程序,而不是先列出程序再解释程序。在各章中由浅入深地结合例题介绍各种典型的算法。对穷举、递推、迭代、递归、排序(包括比较交换法、选择法、起泡法)、矩阵运算、字符处理应用等算法作了详细的介绍,对难度较大的链表操作的算法作了清晰的思路说明。引导读者在拿到题目后,先考虑算法再编程,而不是坐下来就写程序,以培养好的习惯,培养算法思维。

(4) **更加通俗易懂,容易学习。**作者充分考虑到广大初学者的情况,精心

设计体系,适当降低门槛,便于读者入门。尽量少用深奥难懂的专业术语,用通俗易懂的方法和语言阐述清楚复杂的概念,使复杂的问题简单化。没有学过计算机原理和高等数学的读者完全可以掌握本书的内容。

本书采用作者提出的“提出问题—解决问题—归纳分析”的新的教学三部曲,先具体后抽象,先实际后理论,先个别后一般。而不是先抽象后具体,先理论后实际,先一般后个别。实践证明这样做符合读者的认知规律,读者很容易理解。

在介绍每个例题时,都采取以下的步骤:给出问题→解题思路→编写程序→运行结果→程序分析→程序改进。对有关内容用说明、注意、提醒、思考等标注,并用楷体字以引起注意。这种方法符合读者认知规律,使读者思路更加清晰,更容易接受和理解。通过运行程序,读者看到结果,便于验证算法的正确性,学习过程不觉得抽象,而会觉得算法具体有趣,看得见,摸得着。

在每一章的最后有“本章小结”,归纳本章的要点,提纲挈领,画龙点睛,以加深印象,增强条理性。

本书便于自学。具有高中以上文化水平的人,即使没有教师讲解,也能基本掌握本书的内容。这样,就有可能做到:教师少讲,提倡自学,上机实践。

(5) 把教学内容分为基本要求和提高要求两个部分。这是考虑到不同对象的要求的差别。在各章中把一部分较深入的内容作为“提高部分”,单独列出,放在各章的最后,供学生选学和参考。如果学时不够,可以只学基本部分。通过学习基本部分,能初步掌握 C 程序设计的基本内容,写出相对简单的程序,如果希望学习得更深入一些,掌握更多的编程思路和技巧,可以选学各章的“提高部分”。也可以教师讲授基本部分,学生自学“提高部分”,以培养学生的自学能力。

应当怎样学习 C 语言程序设计呢? 作者给出以下建议:

(1) 在开始学习时不要死记语法细节。请记住:重要的是学会编程序,而不是背语法。一开始就要学会看懂程序,编写简单的程序,然后逐步深入。有一些语法细节是需要通过较长期的实践才能熟练地掌握的。初学时,切忌过早地滥用 C 语言的某些容易引起错误的细节(如不适当地使用++和--的副作用)。

(2) 不能设想今后一辈子只使用在学校里学过的某一种语言。但是无论用哪一种语言进行程序设计,其基本思路和方法都是一样的。从这个意义上说,在大学里学哪一种语言,并不是很重要的原则问题。学会了一种计算机语言,可以很快地学会另一种语言。因此,在学习时一定要学活用活,举一反三,掌握规律,在以后需要时能很快地掌握其他语言进行工作。

(3) 在学校学习阶段,主要是学习程序设计的方法,进行程序设计的基本训练,打下将来进一步学习的基础。学习程序设计课程时,应该把精力放在最基本、最常用的内容上,学好基本功。如果对学生有较高的程序设计要求,应当在学习本课程后,安排一次集中的课程设计环节,完成有一定规模的程序

设计。

(4) 程序设计是一门实践性很强的课程,既要掌握概念,又要动手编程,还要上机调试运行,希望读者一定要重视实践环节,包括编程和上机。既会编写程序,又会调试程序。衡量这门课学习的好坏,不是“知不知道”,而是“会不会干”。考核的方法不应当采用标准题(是非题或选择题),而应当把重点放在编制程序和调试程序上。

(5) 使用哪一种编译系统并不是原则问题。程序编好以后,用哪一种编译系统进行编译都可以。读者不应该只会用某一种编译环境,应当了解、接触和使用不同的编译环境。不同的编译系统,其功能和使用方法有些不同,编译时给出的信息也不完全相同,要注意参阅使用说明书,特别要在使用中积累经验,举一反三。

为了帮助读者学习本书,作者还编写了本书的配套参考书《C 语言程序设计(第 3 版)学习辅导》,提供本书中各章习题的参考答案以及上机实习指导。

薛淑斌高级工程师和金莹副教授参加了本书的修订。许多高校老师多年来使用本教材,提出了许多宝贵的意见和建议,使本书得以日臻完善。在此谨向一切帮助和支持过作者的人士表示由衷的感谢。

本书肯定会有不少缺点和不足,热切期望得到专家和读者的批评指正。

谭浩强

2014 年 3 月 1 日于清华园

目录

▶ 第1章 程序设计与 C 语言	1
1.1 计算机与程序、程序设计语言	1
1.2 C 语言的出现和发展过程	2
1.3 简单的 C 语言程序	2
1.4 运行 C 程序的步骤与方法	7
1.4.1 运行 C 程序的步骤	7
1.4.2 上机运行 C 程序的方法	8
本章小结	15
习题	16
▶ 第2章 数据的存储与运算	17
2.1 数据在计算机中是怎样存储的	17
2.1.1 数据在计算机中是以二进制形式存储的	17
2.1.2 位、字节和地址	18
2.1.3 不同类型数据的存储方式	19
2.2 整型数据的运算与分析	21
2.2.1 整型数据运算程序举例和分析	21
2.2.2 整型常量与整型变量	23
2.3 实型数据的运算与分析	26
2.3.1 实型数据的运算举例	26
2.3.2 实型常量的表示形式	28
2.3.3 实型变量	29
2.4 字符型数据的运算	31

2.4.1	字符数据运算的简单例子	31
2.4.2	字符常量和字符变量	32
2.4.3	字符串常量	36
2.5	符号常量	37
2.5.1	为什么要用符号常量	38
2.5.2	符号常量的性质和使用方法	40
2.6	算术运算符和算术表达式	41
2.6.1	算术运算符	41
2.6.2	算术表达式	43
2.7	C 运算符和 C 表达式	45
2.7.1	C 运算符	45
2.7.2	C 表达式	46
2.8	提高部分	46
2.8.1	求补码的方法	46
2.8.2	整型常量的表示形式	47
2.8.3	整型变量的类型	47
2.8.4	整型常量的类型	48
2.8.5	C 语言允许使用的数据类型	49
2.8.6	运算符的优先级与结合性	50
	本章小结	50
	习题	51

第 3 章 最简单的 C 程序设计——顺序程序设计 53

3.1	算法是程序的灵魂	53
3.1.1	什么是算法	53
3.1.2	怎样表示算法	55
3.2	程序的三种基本结构	58
3.3	C 语句综述	59
3.4	赋值表达式和赋值语句	62
3.4.1	赋值表达式	62
3.4.2	赋值过程中的类型转换	64
3.4.3	赋值语句	65
3.4.4	变量赋初值	66
3.5	数据输入输出的概念	67
3.6	字符数据的输入输出	68
3.6.1	用 putchar 函数输出一个字符	69
3.6.2	用 getchar 函数输入一个字符	70

3.7 简单的格式输入与输出	72
3.7.1 用简单的 printf 函数输出数据	72
3.7.2 用简单的 scanf 函数输入数据	76
3.8 顺序结构程序设计举例	79
3.9 提高部分	83
3.9.1 关于无符号数据与有符号数据之间的赋值	83
3.9.2 较复杂的输入输出格式控制	85
3.9.3 简单易用的C++的输入输出	92
本章小结	93
习题	94

▶ 第4章 选择结构程序设计

96

4.1 条件判断	96
4.1.1 条件判断的含义	96
4.1.2 关系运算符和关系表达式	97
4.1.3 逻辑运算符和逻辑表达式	98
4.2 用 if 语句实现选择结构	102
4.2.1 用 if 语句实现选择结构举例	102
4.2.2 if 语句的一般形式	104
4.2.3 if 语句使用的说明	104
4.2.4 使用嵌套的 if 语句实现多层条判断	106
4.3 利用 switch 语句实现多分支选择结构	110
4.4 程序综合举例	112
4.5 提高部分	116
4.5.1 用条件表达式实现简单的选择结构	116
4.5.2 在程序中使用条件表达式	118
本章小结	119
习题	120

▶ 第5章 循环结构程序设计

123

5.1 程序中需要用循环结构	123
5.2 用 while 语句和 do-while 语句实现循环	124
5.2.1 用 while 语句实现循环	124
5.2.2 用 do-while 语句实现循环	126
5.3 用 for 语句实现循环	128
5.3.1 for 语句的一般形式和执行过程	128

5.3.2	for 循环程序举例	129
5.4	循环的嵌套	133
5.5	提前结束循环	133
5.5.1	用 break 语句提前退出循环	133
5.5.2	用 continue 语句提前结束本次循环	135
5.6	几种循环的比较	137
5.7	循环程序综合举例	137
5.8	提高部分	142
5.8.1	while 和 do-while 循环的比较	142
5.8.2	for 语句的各种形式	143
	本章小结	147
	习题	147
第 6 章 利用数组处理批量数据		149
6.1	为什么要用数组	149
6.2	怎样定义和引用一维数组	149
6.2.1	怎样定义一维数组	150
6.2.2	怎样引用一维数组的元素	150
6.2.3	一维数组的初始化	152
6.2.4	一维数组程序举例	153
6.3	怎样定义和引用二维数组	156
6.3.1	怎样定义二维数组	156
6.3.2	怎样引用二维数组的元素	157
6.3.3	二维数组的初始化	157
6.3.4	二维数组程序举例	159
6.4	字符数组	162
6.4.1	怎样定义字符数组及其初始化	162
6.4.2	怎样引用字符数组	163
6.4.3	字符串和字符串结束标志	164
6.4.4	怎样进行字符数组的输入输出	166
6.4.5	字符串处理函数	168
6.4.6	字符数组应用举例	169
6.5	提高部分	173
6.5.1	为什么在定义二维数组时采用两对双括号的形式	173
6.5.2	对 C 的字符串函数的详细说明	174
	本章小结	178

习题	179
第 7 章 用函数实现模块化程序设计	181
7.1 函数是什么	181
7.2 函数的定义和调用	184
7.2.1 为什么要定义函数	184
7.2.2 怎样定义函数	184
7.2.3 怎样调用函数	185
7.2.4 对被调用函数的声明和函数原型	188
7.3 函数的嵌套调用和递归调用	191
7.3.1 函数的嵌套调用	191
7.3.2 函数的递归调用	193
7.4 数组作为函数参数	199
7.4.1 用数组元素作函数实参	199
7.4.2 用数组名作函数参数	200
7.5 变量的作用域和生存期	207
7.5.1 变量的作用域——局部变量和全局变量	207
7.5.2 变量的存储方式和生存期	210
7.5.3 关于作用域和生存期的小结	214
7.6 内部函数和外部函数	215
7.6.1 什么是内部函数	215
7.6.2 什么是外部函数	215
7.7 提高部分	218
7.7.1 实参求值的顺序	218
7.7.2 递归的典型例子——Hanoi(汉诺)塔问题	218
本章小结	221
习题	223
第 8 章 善于使用指针	225
8.1 什么是指针	225
8.2 指针变量	226
8.2.1 使用指针变量访问变量的例子	226
8.2.2 怎样定义指针变量	228
8.2.3 怎样引用指针变量	229
8.2.4 指针变量作为函数参数	231
8.3 通过指针引用数组	236