

高等院校信息科学系列教材

# C++语言 基本程序设计

裘宗燕 编著



科学出版社

TP312C  
34

高等院校信息科学系列教材

# C++语言基本程序设计

袁宗燕 编著

科学出版社

北京

## 内 容 简 介

本书从基本原理出发,以标准 C++语言为工具,介绍了基本程序设计的主要问题。本书在多个层面上详细讨论了对程序问题的分析和分解,还讨论了程序的调试与排除错误等。书中给出了许多程序设计实例,讨论了许多实例的思考、开发和改进过程,并在此基础上提出了许多希望读者进一步思考的问题。

本书共 10 章,内容包括:引论,数据对象与计算,变量和控制结构,基础程序设计,处理数据的集合,类型与指针,C++程序结构,类型程序设计,模板和标准模板库,进一步学习的建议。几个附录汇集了 C++语言及其标准库的许多信息。每章末都附有相当数量的习题。

本书可作为高等院校面向信息科学和计算机等专业的基础课教材,也可供各层次计算机专业人员和业余爱好者阅读参考。

---

### 图书在版编目(CIP)数据

C++语言基本程序设计/裘宗燕编著. —北京:科学出版社, 2003

ISBN 7-03-011125-7

I. C… II. 裘… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 003165 号

---

责任编辑:鞠丽娜/责任校对:都 岚

责任印制:吕春珉/封面设计:槐寿明

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

\*

2003年3月第 一 版 开本: B5 (720×1000)

2003年3月第一次印刷 印张: 27 1/2

印数: 1—5 000 字数: 550 000

定价: 35.00 元

(如有印装质量问题,我社负责调换〈路通〉)

# 序 言

1998年教育部进行高校专业调整时,设立了“信息与计算科学”专业,该专业的设立,受到很多高等院校的热烈响应。据不完全统计,几年来已有约280所院校招收了该专业的本科生,其中大部分院校计划开设信息科学方面的系列课程。

为了配合高等院校在学科专业设置上的改革与深化,来自几十所高等院校有关专业的部分领导和教师,于1999年、2000年召开了第一、二届“信息科学专业发展与学术研讨会”,与会者热烈讨论并探讨了许多关于信息学科的学科发展和建设的基本问题。会议一致认为教材建设是目前最为紧迫的任务,因此成立了教材编写委员会来组织该系列教材的编写。

2001年教材编写协调组召集了有多位经验丰富的教师和出版社参加的教材建设会议。会议明确了教材建设是一项长期的工作,并决定首先编写和出版本套教材来满足近期急需,为了保证教材的质量,会议对每本教材的要求、内容和大纲进行了具体研讨,并请具有多年教学经验的重点院校教授担任各教材的负责人。

为了贴近教学的实际,每本教材都配有习题或思考题,同时对内容也作了结构化安排,以便教师能根据实际情况部分选讲。本套教学用书不仅适用于教学,也可供相关读者参考。

在教材编写和出版过程中,作者对内容的取舍、章节的安排、结构的设计以及表达方式等方面多方听取意见,并进行了反复修改。在感谢作者们辛勤劳作的同时,编委会还特别感谢科学出版社的鞠丽娜编辑,她不辞辛劳,在统筹印刷出版、督促进度、征求意见、组织审校等方面做了大量的工作。这套教材能在保证质量的前提下及时与读者见面,是和她的努力分不开的。

从长远的教学角度考虑,为了适应不同类型院校、不同要求的课程需要,教材编写委员会将不断组织教材的修订、编写(译),从而使信息科学教学用书做到逐步充实、完善、提高和多样化。在此衷心希望采用该系列用书的教师、学生和读者对书中存在的问题及时提出修改意见和建议。

高等院校信息科学系列教材编委会

2002年3月

# 前 言

本书是面向信息科学技术和计算机科学技术领域本科教育的基础课程教材，也可供各层次计算机专业人员和业余爱好者阅读参考。本书以标准 C++ 语言为工具，讨论了程序设计方方面面的问题，介绍了 C++ 的重要功能和机理。本书在许多层面上详细讨论了对程序问题的分析和分解：函数抽象、语言结构和标准库功能的选择使用、类型定义、基于模板和标准模板库的程序设计，同时还讨论了程序调试与排除错误等问题。本书希望告诉读者如何基于 C++ 写出正确、清晰、简洁、高效、可读、易修改的程序。书中给出了许多程序设计实例，讨论了许多实例的思考、开发和改进过程，并在此基础上提出了许多希望读者进一步思考的问题。每章末都附有数量较多的习题。

程序设计在计算机科学教育中的重要性是无庸置疑的。人们也一直在思考第一门课程 (CS1) 应该教什么，在本领域蓬勃发展的情况下 CS1 应怎样改造。国外从 20 世纪 90 年代开始研究用 C++ 语言改造 CS1，并推出了一批教材，虽然它们各有特点，但也有一些问题，因此我才动手撰写了这本基于 C++ 的教材，书中反映了对本课程的以下一些基本考虑：

1) 本书假定读者 (学生) 没有程序设计经验，或者只有很少经验，因此书中讨论了学习中可能遇到的各种问题，对概念和观点都尽量给出清楚的解释，其目的就是希望本书适合作为基于 C++ 的第一门程序设计课程的教材。

2) 本书以程序设计为基本线索，讨论如何认识程序、写程序，如何用 C++ 写程序，因此书中对从问题出发分析和分解问题、设计求解过程、找出主要步骤、确定函数抽象、确定循环、选择语言结构和标准库功能、定义类型、最后写出程序的全过程有许多深入讨论，此外对不少实例给出了在不同考虑下可能形成的多种解法，希望能更好地帮助读者理解程序设计的真谛。

3) 强调好的程序设计风格，强调通过功能分解建立程序清晰结构的重要性。本书特别强调思考程序设计的正确方式，强调基于标准 C++ 语言及其标准库的良好程序设计途径。书中很早就介绍了函数概念，从使用库函数到简单函数定义，再到函数的完整定义等。为了让读者更容易深入到程序设计的丰富领域，不致陷入低级程序机制的细节之中，本书从第一章就开始使用标准库类型。书中特别强调程序的结构性、可读性和易修改性，提供的程序实例也努力遵循这些原则，努力使它们简洁、清晰、易读。书中还根据遇到的问题分析了一些不良程序设计习惯及其危害。

4) 强调良好的 C++ 基本程序设计。由于历史原因，C++ 语言是一个较复杂的混合型语言，其中不但包括许多新的高级程序机制，也包含许多从 C 语言继承来

的低级机制。从 C++ 的低级机制入门，很容易受到烦琐细节的纠缠，反而容易忽略对程序设计更重要方面的探究和理解。有些基础书籍从面向对象的观点出发，开始就讨论了一大堆术语，初学者常常摸不着头脑，难以理解和深入。本书虽然仍从基本程序设计入手，但尽可能回避 C++ 的低级成分，利用流输入输出，标准库的各种强大类型，使书中讨论能较平稳地进入实质性的领域。虽然本书中讨论了许多高级功能（包括各种标准库类型、类型程序设计、模板和标准模板库等），但却设法回避了许多难点。这样，在学完本书之后，读者已经可以写出很多有实际意义的程序了。对程序设计的这些基本认识和经验，将为读者进一步去学习 C++ 的低级机制或面向对象等高级技术打下基础。

5) 尽可能对 C++ 语言中所涉及到的结构和机制做较细致的介绍和解释，因为其中不少问题反应了相关领域的普遍性问题。这里不仅介绍情况，还尽力解释有关的背景和理由，以期帮助读者理解问题的实质，无论是关于程序设计过程，还是关于 C++ 语言本身。这样做的目的是希望学习者能在阅读中理解问题的实质，而不是自己设法在模糊中去感悟。这对于帮助初学者尽快进入程序领域是非常重要的，也可能使教师有更大的自由度，更灵活地选择例子和教学方式。

6) 由于程序设计的性质，书中实例没有采用常见的“提出问题，给出解答，加一点解释”的简单三步形式，而是着重于帮助人们认识理解从问题到程序的思考过程，希望能使读者更好地理解程序设计的实际过程。

除了上述考虑之外，本书撰写中的一个主要想法就是希望反映程序设计的两个重要特征：科学性和工程性。科学性指程序构造过程本身的科学依据。有关程序设计过程的研究发展形成了程序和程序设计的理论，初级教科书也必须反映程序理论的精神实质，使初学者及早看到程序和程序语言的一些本质性问题。这一点，无论从提高人们对计算机科学技术认识水平的角度看，还是从进一步深入学习角度看，都是十分必要的。

目前不少学生在本课程之前已经有了一些计算机经验，有些人对计算机很熟，有的已写过不少程序。但还要看到问题的另一面，对程序设计已建立起较好认识的学生仍是极少数，即使是那些计算机已经很熟的学生，也有许多观点需要建立，或有些错误认识需要纠正，这也是本课程应特别强调科学性的原因。

现在有一些学生对计算机很熟，动手很快，编程序时不先做细致的分析设计就很快写出一堆代码，随后就去运行、追踪、打补丁改错。这些学生能写出解决简单问题的程序，但质量往往很差，一个简单问题也写出一大篇程序，自己也说不清其中的细节。当问题变得复杂时，他们就遇到了困难，出了错误也弄不清错在哪儿，应怎样修改补救。这种程序就像建在沙滩上的大楼，无论如何修补也为时已晚，只能推倒重来。

上述情况说明，强调程序设计过程的科学性是非常必要的，基础课程中更是这样。从最简单的程序开始，就必须注意程序设计的正确途径。只有学会做好小

程序，今后才可能做出大程序，只满足于做出可运行的小程序是远远不够的。本书从开始就强调这些问题，强调在各种层次上的分解和信息隔离，后面章节也有进一步的论述。书中还通过程序实例介绍了一些更深入的问题，如通过运行时间介绍计算的基本性质（复杂性）；通过对循环的分析介绍“循环不变关系”的概念和意义，通过类型的设计介绍“数据不变式”的概念等，希望这些实例能成为读者们思考程序问题的线索。

程序设计工作的另一重要特征是工程性。在工程计划设计中不仅需要有问题分析和理解能力，设法寻找可能的解决方案，对各种方案做出评价和选择，还应对所做选择有清楚的认识（优点和缺点，是否有所偏向或排斥等等），进而确定具体实施方案。在程序设计过程中，这些问题都有直接反映。如果我们不是把学习目标定位在解几道典型程序题目上，而是放在提高能力、真正理解程序设计过程方面，那么就特别需要努力提高处理这些问题的能力。

为帮助读者认识程序设计的实质，本书的叙述和实例都力图强调一种观点：由于对问题分析的不同考虑，以及工作过程中的不同决定或选择，对同一问题，人们常会得到许多合理的特征迥异的程序。它们常常各有长短，可能有侧重点差异，也可能反映了对问题的认识差异。程序工作中需要理清所面对的问题，寻找可能的求解途径，把复杂问题分解为相对简单的步骤，合理选择适用的程序语言结构等。这里的每一步都可能产生分支，而我们在每次选择时都应该设法看清所采纳选择的影响，无论是收获还是损失。

工程中通常没有完美的选择，更多的是平衡和折衷。这些对于程序设计也是本质性的。本书给出了对许多实例的详细分析过程，有时对一个问题给出了多个解答，指出了其他可能性：还可能如何看问题，还可能如何做等。对一些实例，在做出一个解之后又进一步分析，提出其中的缺陷并进一步改进。书中常针对一些实例，给读者提出问题，希望读者发挥自己的主观能动性。各章节后面的练习也试图反映这些看法。

本书希望帮助读者建立一种认识：程序设计中并没有惟一的标准答案，程序设计过程只是在努力追求比较好的“正确”答案。各种书籍（包括本书）中给出的答案不过是作者对问题的分析和理解的反映，还可能存在很多其他选择。进一步说，我们还希望读者在看别人的程序时，应注意思考程序中隐含了程序作者的哪些考虑和选择，其中哪些是合理的有价值的（或不合理的无价值的），在哪些地方还能有什么选择，沿着其他选择走下去可以得到什么（或失去什么）等等。如果读者能养成这种思考习惯并坚持这样做，必将会从中受益无穷。当然，这绝不是说书中的例子不重要，恰恰相反。正因为程序设计过程中有许多可能选择，书中实例更应当反映某种合理的好选择，以供读者参考。在初等入门书中，一定要说明采用特定选择的理由。如果可能，还应当指出采取这些选择带来的问题（缺点、造成的限制等）。

程序设计是“从问题到程序”的一个工作过程。这个工作既要求遵循严格的科学方法，又要求具有谨慎灵活的工程能力。要很好地完成程序设计工作，编程的人既需要充分发挥自己的聪明才智，又需要有细致认真、一丝不苟的工作态度。即使是将来不从事计算机程序方面的工作，由这个课程得到的锻炼也可能非常重要，尤其对于理科学生，这个课程可能弥补他们工程方面训练的不足。

前些年 CS1 常用 PASCAL 等语言讲授，近年许多课程中用 C 语言。从作为 CS1 教学用语言的角度看，没有一种程序语言具有无可比拟的天然优势，选择任何语言都需要考虑其有利方面，也需要努力克服这种选择带来的缺陷。

选择 C++ 作为教学语言的主要理由包括：C++ 是目前实际程序设计实践工作中使用最广泛的语言之一，它包含了程序设计中需要理解和使用的传统程序机理和机制，还包含了许多能反映近十几年程序设计理论和实践发展的新机制和新认识。正确地理解掌握这些机制及其原理，能帮助学习者理解程序与程序设计的主要问题，完成各种作为练习的程序，并得到有关方面的知识积累和能力锻炼。目前许多实际系统的全部或部分是用 C++ 编写的，以 C++ 作为教学工具语言，不但能学习程序设计的基础知识，同时也能掌握一种实用的程序设计工具。另一方面，在学习了 C++ 程序设计之后，学习后续课程都比较方便。第三，C++ 是一种很丰富的语言，这使我们有可能选择一种合理的入门途径，也能为将来进一步学习提供广阔的空间。本书第 10 章介绍了一些进一步学习的可能性。请读者谨记，用任何“好语言”都不会妨碍人们写出“坏程序”。

本书包括 10 章和若干附录，章节内容如下：

第 1 章，引论。首先介绍程序与程序语言的概念，C++ 语言的简单发展及其特点。用几个小例子介绍了 C++ 程序的一些基本问题、程序加工和执行，最后介绍程序设计与开发过程中的问题，如调试、追踪、排除错误等。

第 2 章，数据对象与计算。讨论程序语言的许多最基本概念，包括字符集、标识符和关键字，数据与类型，数据表示，运算符、表达式与计算过程，数学函数库的使用，基本输入输出等。

第 3 章，变量和控制结构。讨论基本程序设计的一些问题，包括基本语句与复合结构，变量的概念和使用，简单函数的定义，程序中逻辑条件的描述与使用，最后介绍了几种最常用的控制结构。

第 4 章，基础程序设计。首先通过许多程序实例讨论循环程序设计的基本方法，随后介绍了函数的递归定义和为程序计时，其他控制结构及其使用，最后用较大的篇幅讨论了与输入输出有关的许多问题：输出格式，文件的使用，输入控制，写交互式程序时需要考虑的问题，还讨论了程序的调试等。

第 5 章，处理数据的集合。首先通过实例，介绍了标准库的向量类型，使用向量的程序设计技术以及 string 类型的各种有用操作。后一部分介绍 C++ 低级的数组机制，字符数组和 C 风格字符串，多维数组的处理等。



第 6 章，类型与指针。综述了 C++ 的内部类型，而后介绍了 C++ 的用户类型定义机制 `enum` 和 `struct`，并通过一个较大的程序实例，较全面地讨论了程序开发和改进的过程。本章后一部分介绍指针的概念和指针变量的使用，介绍了 C++ 语言中指针与数组的关系，命令行参数的处理，处理多维数组的通用函数，指向函数的指针等。

第 7 章，C++ 程序结构。这里讨论函数抽象，作用域和生存期，预处理的概念和预处理命令，程序的分块开发，名字空间概念和使用，程序中的错误处理和异常，程序的初始化。最后介绍了动态存储管理机制。

第 8 章，类型程序设计。通过一个定义类 (`class`) 的例子，介绍了 C++ 中最重要的类型定义机制，而后全面讨论了各种类型定义机制的情况，通过实例讨论定义类型过程中的思考，最后讨论了几个类型定义实例，最重要的是一个可以动态增长的向量类。

第 9 章，模板和标准模板库。通过实例阐释了类模板和函数模板的定义过程，随后的主要部分介绍标准模板库的重要概念和机制：容器和迭代器，标准容器 `map` 类模板，序列的概念，各种标准算法，函数对象，最后给出了所有标准库算法的总表。

第 10 章，进一步学习的建议。

最后有几个附录，总结了 C++ 语言、标准库的主要功能，每种功能的简介中都包含了与前几章讨论有关机制的章节的索引，供读者交叉参考。

本书以标准 C++ 语言为基础，内容只牵涉到 C++ 语言本身，所有例子都按 C++ 标准书写，习题中也不牵涉任何具体环境，因此，本书的学习并不依赖于具体的 C++ 语言系统，不要求复杂的环境支持，可用任何符合 ISO/ANSI C++ 标准的 C++ 系统作为编程实践环境。附录中介绍了若干可用 C++ 系统的情况，也讨论了它们与 C++ 标准的差异，供读者参考。

虽然本书是我多年工作和思考的产物，但书中难免有错误和不足，这些都只能由我自己负责，希望读者和同行们提出宝贵意见。

袁宗燕

北京大学数学学院信息科学系

2002 年 5 月

# 目 录

<b>第 1 章 引论</b> .....	1
1.1 计算机和程序、程序设计语言 .....	1
1.2 C++语言简介 .....	5
1.3 简单的 C++程序 .....	6
1.3.1 注释 .....	7
1.3.2 C++程序与格式 .....	8
1.3.3 语法和语义 .....	9
1.3.4 空白字符和程序格式 .....	11
1.3.5 程序的加工和运行 .....	11
1.3.6 基本操作与程序 .....	12
1.3.7 输入和输出 .....	14
1.4 程序开发 .....	17
1.5 学习程序设计 .....	22
1.6 有关 C++语言和程序的一些说明 .....	24
习题 .....	27
<b>第 2 章 数据对象与计算</b> .....	28
2.1 基本字符、标识符和关键字 .....	28
2.2 基本数据类型 .....	29
2.2.1 整数类型 .....	30
2.2.2 实数类型 .....	34
2.2.3 字符类型和字符串 .....	36
2.3 数据的外部表示、内部表示与输入输出转换 .....	38
2.4 表达式的计算过程 .....	40
2.4.1 表达式求值的顺序问题 .....	40
2.4.2 表达式计算与类型 .....	42
2.5 数学函数库及其使用 .....	45
2.5.1 函数 .....	45
2.5.2 标准库的数学函数 .....	46
2.5.3 函数调用中的类型转换 .....	48
2.6 类型问题 (阅读材料) .....	49
习题 .....	52

<b>第3章 变量和控制结构</b> .....	<b>53</b>
3.1 语句和复合结构.....	53
3.2 变量的概念、定义和使用.....	54
3.2.1 变量的定义.....	55
3.2.2 变量的使用：取值与赋值.....	56
3.2.3 与变量有关的几个问题.....	58
3.3 函数与函数定义（初步）.....	61
3.3.1 函数定义.....	62
3.3.2 函数和程序.....	65
3.3.3 函数与类型.....	66
3.4 关系表达式、逻辑表达式、条件表达式和 bool 类型.....	67
3.4.1 关系表达式和条件表达式.....	67
3.4.2 复杂条件的清晰描述：逻辑表达式.....	70
3.5 语句与控制结构.....	72
3.5.1 条件语句（if 语句）.....	73
3.5.2 循环语句（1）：while 语句.....	77
3.5.3 循环语句（2）：for 语句.....	80
3.6 程序中的常量.....	81
3.7 一些常用程序机制.....	82
3.7.1 增量和减量运算符（++、--）.....	83
3.7.2 逗号运算符.....	84
3.7.3 实现二元运算符操作的赋值运算符.....	84
3.7.4 空语句.....	85
3.7.5 程序中的另一种注释形式.....	85
3.8 按位运算和按位运算符（选读）.....	86
习题.....	89
<b>第4章 基础程序设计</b> .....	<b>91</b>
4.1 基本的循环程序设计.....	92
4.1.1 发现循环.....	92
4.1.2 典型循环.....	95
4.1.3 一个程序实例.....	100
4.1.4 判断函数（谓词）.....	102
4.2 递归的函数定义.....	104
4.2.1 递归函数定义.....	105
4.2.2 循环和递归定义导致的计算过程.....	106

4.2.3	最大公约数.....	107
4.3	计算和时间.....	111
4.4	其他控制结构和控制语句.....	116
4.4.1	do-while 循环结构.....	116
4.4.2	流程控制语句.....	117
4.4.3	开关语句.....	120
4.5	输入和输出.....	121
4.5.1	输出格式控制.....	121
4.5.2	输入和控制.....	124
4.5.3	文件的概念和输入输出.....	126
4.5.4	字符的输入和输出.....	132
4.5.5	字符类别.....	136
4.6	交互式程序的问题.....	138
4.6.1	由输入控制的循环.....	138
4.6.2	处理输入错误.....	140
4.7	程序的测试和排错.....	143
4.7.1	测试.....	144
4.7.2	排错.....	147
	习题.....	149
<b>第 5 章</b>	<b>处理数据的集合</b> .....	<b>153</b>
5.1	向量.....	154
5.1.1	定义向量.....	154
5.1.2	使用向量.....	156
5.1.3	有关向量的总结.....	161
5.2	处理向量的函数.....	162
5.3	向量与检索.....	167
5.3.1	分析和认识.....	168
5.4	string 的操作.....	170
5.4.1	拼接、附加和比较.....	170
5.4.2	字符和子串.....	171
5.4.3	输出与输入.....	172
5.5	数组.....	173
5.5.1	定义数组变量.....	173
5.5.2	数组操作.....	174
5.5.3	数组的存储实现.....	176

5.5.4	数组作为函数的参数 .....	177
5.5.5	有关数组参数的说明 .....	179
5.6	字符数组与 C 风格的字符串 .....	181
5.6.1	字符数组 .....	181
5.6.2	C 风格字符串 .....	181
5.6.3	处理 C 风格字符串的程序和函数 .....	184
5.7	二维和 multidimensional 数组 .....	185
5.7.1	多维数组的定义和初始化 .....	186
5.7.2	多维数组的使用 .....	187
5.7.3	以多维数组为参数的函数 .....	188
5.8	一个程序实例 .....	189
	习题 .....	195
<b>第 6 章</b>	<b>类型和指针 .....</b>	<b>197</b>
6.1	类型问题 .....	197
6.2	内部类型 .....	200
6.2.1	实数类型 .....	200
6.2.2	具有整数值的类型 .....	201
6.2.3	选择基本类型的建议 .....	203
6.3	枚举 .....	203
6.4	结构和类 .....	206
6.4.1	一个问题: 学生成绩评定 .....	206
6.4.2	struct 的概念和定义 .....	207
6.4.3	struct 变量的使用 .....	209
6.4.4	结构与函数 .....	209
6.5	程序实例: 学生成绩总评 .....	211
6.5.1	读入循环和辅助函数 .....	212
6.5.2	输出和主函数 .....	214
6.5.3	字符式直方图 .....	215
6.5.4	字符式直方图程序的改进 .....	217
6.5.5	程序的重整 .....	218
6.6	指针 .....	220
6.6.1	指针变量的定义和使用 .....	221
6.6.2	指针与函数 .....	225
6.6.3	指针与数组 .....	227
6.6.4	字符指针 .....	232

6.6.5	命令行参数的处理 .....	235
6.6.6	多维数组作为函数参数 .....	238
6.7	类型同义词 (typedef) .....	240
6.8	指向函数的指针 .....	240
6.8.1	函数指针的定义和使用 .....	241
6.8.2	函数指针作为函数的参数 .....	242
6.8.3	数值积分函数 .....	244
习题	.....	246
<b>第7章</b>	<b>C++程序结构</b> .....	<b>249</b>
7.1	程序的基层组织——函数抽象 .....	249
7.1.1	函数 .....	250
7.1.2	程序功能的函数分解 .....	252
7.1.3	有关函数的两种观点 .....	254
7.1.4	函数原型 .....	258
7.1.5	带默认值的参数 .....	259
7.1.6	函数重载 .....	260
7.2	基本程序结构, 作用域与生存期 .....	262
7.2.1	外部定义的变量 .....	262
7.2.2	作用域和生存期 .....	264
7.2.3	变量定义的嵌套 .....	266
7.2.4	静态局部变量 .....	267
7.2.5	变量的初始化 .....	268
7.2.6	常变量 .....	269
7.2.7	引用变量 .....	270
7.3	程序加工和预处理 .....	270
7.3.1	文件包含 .....	271
7.3.2	宏定义和宏替换 .....	271
7.3.3	条件命令 .....	272
7.4	分别编译、分块开发和源程序组织 .....	273
7.4.1	实例研究: 学生成绩评定 .....	274
7.4.2	源程序分块技术 .....	278
7.4.3	防止重复包含 .....	281
7.5	全局作用域与名字空间 .....	281
7.5.1	标准库和名字空间的使用 .....	282
7.5.2	使用声明 .....	283

7.5.3	名字空间的定义 .....	284
7.6	错误处理 .....	286
7.6.1	朴素处理方式及其问题 .....	287
7.6.2	处理复杂的输入格式 .....	292
7.6.3	异常和异常处理 .....	295
7.7	程序的功能参数和初始化 .....	298
7.7.1	交互式初始化方式 .....	300
7.7.2	通过命令行参数方式 .....	300
7.7.3	采用初始化文件方式 .....	302
7.8	程序开发 .....	303
7.9	动态存储管理 .....	304
	习题 .....	307
<b>第 8 章</b>	<b>类型程序设计 .....</b>	<b>309</b>
8.1	一个有理数类型 .....	309
8.1.1	访问控制 .....	310
8.1.2	初始化与构造函数 .....	311
8.1.3	复制和赋值 .....	313
8.1.4	各种运算 .....	315
8.1.5	输入和输出 .....	318
8.1.6	补充和整理 .....	320
8.1.7	类定义的程序组织 .....	322
8.2	定义类型 .....	325
8.2.1	枚举 (enum) .....	325
8.2.2	结构 (struct) .....	326
8.2.3	类 (class) .....	327
8.3	一个随机数类 .....	334
8.4	一个向量类 .....	337
8.4.1	界面 .....	337
8.4.2	数据表示和构造函数 .....	338
8.4.3	完成的类 .....	342
8.4.4	修改设计: 能改变大小的向量 .....	343
	习题 .....	349
<b>第 9 章</b>	<b>模板和标准模板库 .....</b>	<b>351</b>
9.1	类模板 .....	351
9.1.1	类模板的定义 .....	352

9.1.2	构造函数和赋值 .....	353
9.1.3	类模板的使用和实例化 .....	354
9.1.4	多参数类模板和非类型参数 .....	355
9.2	函数模板 .....	357
9.2.1	一个简单例子 .....	357
9.2.2	类模板 FVec 的输出运算符 .....	358
9.3	容器和迭代器 .....	359
9.3.1	标准库 map 类型 .....	359
9.3.2	容器和迭代器 .....	361
9.4	标准算法 .....	365
9.4.1	装填和生成 (fill 和 generate) .....	365
9.4.2	复制和转换 (copy 和 transform) .....	367
9.4.3	排序 (sort) .....	370
9.4.4	操作每个元素 (for_each) .....	373
9.5	函数对象 .....	373
9.5.1	函数对象的例子 .....	374
9.5.2	实现算术运算的函数对象类 .....	376
9.5.3	谓词和实现逻辑运算的函数对象类 .....	377
9.5.4	约束器 .....	377
9.6	另一些常用算法 .....	378
9.6.1	查找和计数 (find 和 count) .....	378
9.6.2	替换 (replace) .....	380
9.6.3	删除和去除重复 (remove 和 unique) .....	381
9.6.4	相等和不匹配 (equal 和 mismatch) .....	383
9.6.5	各种值交换 (swap) .....	384
9.6.6	标准算法汇总 .....	385
9.6.7	自定义算法 .....	388
9.7	标准容器 map 的比较准则 .....	389
	习题 .....	390
<b>第 10 章</b>	<b>对进一步学习的建议 .....</b>	<b>392</b>
<b>附录一</b>	<b>C++语言基本元素 .....</b>	<b>396</b>
<b>附录二</b>	<b>C++语言结构 .....</b>	<b>401</b>
<b>附录三</b>	<b>C++标准库 .....</b>	<b>407</b>
<b>附录四</b>	<b>C++语言系统及其使用 .....</b>	<b>416</b>



# 第 1 章 引论

## 1.1 计算机和程序、程序设计语言

计算机的全称为“通用电子数字计算机”，是一种通用的信息处理设备，采用电子电路作为其实现的基础，采用数字化的信息表示方式。经过几十年的发展，计算机已经广泛地应用于社会生产生活的各个领域，这些应用已经给社会各个方面带来了极其深刻的影响，推动了整个社会信息化的大潮。计算机科学技术的发展对社会进步的推动作用，已经是每个人都已很清楚地感受到、认识到的现实了。放眼未来，计算机科学技术在社会生活各领域中的进一步渗透的潜力似乎是无穷的。

从本质上看，计算机与以往人们开发使用的各种信息设备截然不同。其他可以看作信息设备的器具都是专用的，只能服务于某些特定用途。例如，铅笔也可以看作是一种信息设备，其功能就是能将信息记录到纸面上；电话也是一种信息设备，其功能就是能通过电话线传播声音信号。计算机则不同，就其本质而言，它并不是为用于任何特定信息服务而开发的设备，而是为一切信息服务而设计实现的。计算机的工作原理保证它可能被用到所有的各种各样的信息服务中，无论是计算、处理、存储、传输、控制等。

### 1. 通用性与程序

计算机无穷潜力的基础在于它可以按照程序所指定的方式自动工作，这就使我们可以通过编程，使计算机能服务于各种特定用途。编程或程序设计（programming）的概念并不是计算机发明之后才出现的。按照英文原义，编程就是做计划，细致地计划好一系列工作步骤，以便完成某件比较复杂的工作。可编程设备发展的一个重要里程碑是可编程的织机，例如织袜机、织毛衣机一类设备。19 世纪初英国人发明了这种设备，通过给它提供不同的表示各种彩色线走上走下的打孔纸带，自动织机就能织出各种花纹图案了。

计算机硬件的基本功能是可以执行一组指令，其 CPU（中央处理器）遇到了某条指令，就去执行与之对应的一个动作。CPU 的特点是能够自动地去取指令并执行指令，这样，只要我们为它提供了编排好的一系列指令，一个程序，CPU 就能按照这个程序自动工作了。在这里还有一个情况值得指出：CPU 不但能够一条一条地顺序执行指令，还能根据被处理数据的情况，按照某些特定指令的要求，转到程序里指定的地方继续执行。自动取指令自动执行，并能根据数据情况决定