中国高等学校计算机科学与技术专业（应用型）规划教材

丛书主编 陈明

程序设计基础

谢书良 编著

清华大学出版社

中国高等学校计算机科学与技术专业（应用型）规划教材

丛书主编 陈明

程序设计基础

清华大学出版社

北京

内 容 简 介

本书是为从来没有接触过程序设计的读者编写的“零起点”入门教材。全书共分 8 章，第 1 章主要介绍程序设计的概念和程序运行的环境，第 2 章介绍了基本的数据类型、运算符与表达式，第 3 章介绍面向过程程序的顺序、分支选择和循环三种控制结构，第 4 章至第 7 章分别介绍了数组、指针的概念，结构体和其他数据类型，函数及其调用，内容涵盖了 C++ 面向过程程序设计内容，与 C 语言教材完全兼容。第 8 章是体现本书特色的一章，分别介绍了“小型通讯录查询系统”和“学生成绩管理系统”的设计过程并附有完整代码，作为最后的“课程实践”还提供了两个控制台工程样例。

本书还为授课教师提供精心设计的配套电子课件、全部例题源代码、自测练习题答案和部分题目的源代码，可在清华大学出版社网站上下载。

本书可作为高等院校涉及程序设计的相关专业程序设计课程的教材，也可作为工程技术人员的参考用书和有志于程序设计的社会青年的自学用书。

图书在版编目（CIP）数据

程序设计基础/谢书良编著．—北京：清华大学出版社，2010.5
（中国高等学校计算机科学与技术专业（应用型）规划教材）
ISBN 978-7-302-21791-6

Ⅰ．①程…　Ⅱ．①谢…　Ⅲ．①程序设计—高等学校—教材　Ⅳ．①TP311.1

中国版本图书馆 CIP 数据核字（2010）第 002510 号

责任编辑：谢　琛　顾　冰
责任校对：时翠兰
责任印制：何　芊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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：北京市世界知识印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185×260　　**印　　张**：20.25　　**字　　数**：455 千字
版　　次：2010 年 5 月第 1 版　　**印　　次**：2010 年 5 月第 1 次印刷
印　　数：1～4000
定　　价：29.50 元

产品编号：035223-01

编委会

序　言

应用是推动学科技术发展的原动力，计算机科学是实用科学，计算机科学技术广泛而深入地应用推动了计算机学科的飞速发展。应用型创新人才是科技人才的一种类型，应用型创新人才的重要特征是具有强大的系统开发能力和解决实际问题的能力。培养应用型人才的教学理念是教学过程中以培养学生的综合技术应用能力为主线，理论教学以够用为度，所选择的教学方法与手段要有利于培养学生的系统开发能力和解决实际问题的能力。

随着我国经济建设的发展，对计算机软件、计算机网络、信息系统、信息服务和计算机应用技术等专业技术方向的人才的需求日益增加，主要包括：软件设计师、软件评测师、网络工程师、信息系统监理师、信息系统管理工程师、数据库系统工程师、多媒体应用设计师、电子商务设计师、嵌入式系统设计师和计算机辅助设计师等。如何构建应用型人才培养的教学体系以及系统框架，是从事计算机教育工作者的责任。为此，中国计算机学会计算机教育专业委员会和清华大学出版社共同组织启动了《中国高等学校计算机科学与技术专业（应用型）学科教程》的项目研究。参加本项目的研究人员全部来自国内高校教学一线具有丰富实践经验的专家和骨干教师。项目组对计算机科学与技术专业应用型学科的培养目标、内容、方法和意义，以及教学大纲和课程体系等进行了较深入、系统的研究，并编写了《中国高等学校计算机科学与技术专业（应用型）学科教程》（简称《学科教程》）。《学科教程》在编写上注意区分应用性人才与其他人才在培养上的不同，注重体现应用型学科的特征。在课程设计中，《学科教程》在依托学科设计的同时，更注意面向行业产业的实际需求。为了更好地体现《学科教程》的思想与内容，我们组织编写了《中国高等学校计算机科学与技术专业（应用型）规划教材》，旨在能为计算机专业应用型教学的课程设置、课程内容以及教学实践起到一个示范作用。本系列教材的主要特点如下：

1. 完全按照《学科教程》的体系组织编写本系列教材，特别是注意在教材设置、教材定位和教材内容的衔接上与《学科教程》保持一致。

2. 每门课程的教材内容都按照《学科教程》中设置的大纲精心编写，尽量体现应用型教材的特点。

3. 由各学校精品课程建设的骨干教师组成作者队伍，以课程研究为基础，将教学的研究成果引入教材中。

4. 在教材建设上，重点突出对计算机应用能力和应用技术的培养，注重教材的实践性。

5. 注重系列教材的立体配套，包括教参、教辅以及配套的教学资源、电子课件等。

高等院校应培养能为社会服务的应用型人才，以满足社会发展的需要。在培养模式、教学大纲、课程体系结构和教材都应适应培养应用型人才的目标。教材体现了培养目标和育人模式，是学科建设的结晶，也是教师水平的标志。本系列教材的作者均是多年从事计算机科学与技术专业教学的教师，在本领域的科学研究与教学中积累了丰富的经验，他们将教学研究和科学研究的成果融入教材中，增强了教材的先进性、实用性和实践性。

目前，我们对于应用型人才培养的模式还处于探索阶段，在教材组织与编写上还会有这样或那样的缺陷，我们将不断完善。同时，我们也希望广大应用型院校的教师给我们提出更好的建议。

《中国高等学校计算机科学与技术专业（应用型）规划教材》主编

陈明

2008 年 7 月

前　言

当前，IT 技术的发展正在突飞猛进，日新月异。在计算机应用日益普及的形势下，软件的概念和程序设计的基础知识已逐渐成为人们渴求的新目标。有人把数学誉为“训练思维的体操”，其实，程序设计基础课程和数学课程一样，对培养人们的逻辑思维能力，有着异曲同工的作用。

目前市面上高级语言程序设计类的教材琳琅满目，但是真正适合初学者使用的入门教材并不多见，很多教材没有充分考虑初学者的实际情况，使用效果不太理想。作者经过在不同条件、不同环境下长期的教学实践，编写了本书，这是一本适合初学者使用的、体现零起点的入门教材，本书有如下特点：

1. 教材的广度虽然是按传统的知识范围来确定的，但进行了删繁就简处理，以体现入门教材起点低、逐步提升、循序渐进的精神。

2. 对于数据的输入和输出的方式，输入采用的是输入流 cin，无格式输出采用输出流 cout，格式输出采用输出函数 printf。格式输入函数 scanf 只在学了有关“地址”的内容后，结合指针内容进行介绍，这样既显得自然，又有效地降低了教学难度。

3. 将“指针”一章紧接着“数组”一章介绍，将“函数及其调用”一章后移，以适应对学生逻辑思维能力循序培养，逐步上升的进程。部分 * 号注明指针内容，供选用。

4. 根据“学以致用”的原则，增加了“综合应用”一章，并在其他各章选用的例题中，只采用了部分对理解所学知识有用的纯数学类型题，增加了大量的有实用价值的题，以提高兴趣，激发上进，使理论与实践结合得更为紧密。

5. “多思考，勤上机”是学好程序设计课程的关键，本教材对每次上机的目的、内容等项目均有明确的要求，所采用的是 32 位上机环境，考虑到初学者入门的需要，只介绍 Visual C++ 6.0 的简单使用。

6. 为了对教与学提供方便，本教材备有演示文稿提供给教师教学和学生复习选用。

7. 每章之后都设计了一套有多种题型、一定题量的自测练习题，供课堂练习使用。全部题目的参考答案已附于书后，也可以通过清华大学出版社网站下载。

本书涵盖 C++ 面向过程程序设计内容，实际上是用 C++ 的语法讲 C 的内容，与 C 语言教材完全兼容，是按任务导引教学的方法进行编写。与本书配套的《程序设计应用》则涵盖了 C++ 面向对象程序设计和 VC++ 可视化程序设计内容。这样设计，一是便于教学安排，用一个学期 4 至 6 课时/周讲完本书，用另一个学期 4 课时/周讲完《程序设计应用》；二是为

了最后一章综合应用的设计工程的需要。学完程序设计应用课程之后，工程项目的界面应当符合实际需要，且必须与数据库挂接。这样就为今后学习其他程序设计专业课程，如C#等打下了较好的基础。多年的教学实践证明，一个学期很难学完C++的全部内容。强制执行的结果，只会大量削减面向对象的内容，这样做是很不妥当的。

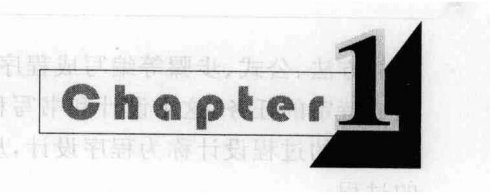
供后续学期使用的《程序设计应用》的内容主要包括类、类的数据共享与类的对象、继承和派生、多态性与虚函数、模板和异常处理、可视化编程基础、资源的应用、综合应用。

用C++语言讲述C语言的内容，重视基础、强化应用是顺应程序设计语言发展历史潮流的一个新的尝试。本书虽已经过多年的试用，进行过反复修改，仍会存在不足之处，请使用者不吝赐教，使其不断完善。

谢书良

2010年2月

目 录



第1章 程序设计概述

教学要求

- 了解程序及程序设计的概念。
- 了解面向过程程序设计的基本特点。
- 理解算法的含义。
- 掌握程序的基本结构和程序运行的流程。
- 熟悉 Visual C++ 6.0 开发环境的简易使用。

1.1 基本概念

1. 程序

本书从如何计算两个数的平均值这样一个最简单的问题讲起。

- 如果这两个数是 3 和 5，几乎可以不加思索地说出它们的平均值是 4。
- 如果这两个数是 23763965432.2187563 和 8456234445446456.43701，它们的平均值只能由计算机去完成。

不管怎么算，人和计算机的计算步骤都是：

① 确定要计算的是哪两个数；

② 先求出两个数之和；

③ 再将此和除以 2；

④ 最后报告计算结果。

其实计算机自身并不会计算，必须由人来教会它。那么人们应该做什么呢？就一般的问题来说，人们要做的事应该是：针对要完成的任务，编排出正确的方法和步骤，并且用计算机能够接受的形式，把方法和步骤告诉计算机，指挥计算机完成任务。

解决问题的方法和步骤，以计算机能够理解的语言表达出来，就称为“程序”。程序是要计算机完成某项工作的代名词，是对计算机工作规则的描述。

计算机软件是指挥计算机硬件的，没有软件，计算机是什么事也做不了，而软件都是由各种程序构成的，程序是软件的灵魂。

2. 程序设计

人们要利用计算机解决实际问题，首先要按照人们的意愿，借助计算机语言，将解决问

题的方法、公式、步骤等编写成程序，然后将程序输入到计算机中，由计算机执行这个程序，完成特定的任务，这个设计和书写程序的整个过程就是程序设计。简言之，为完成一项工作的规则的过程设计称为程序设计，从根本上说，程序设计是人的智力克服客观问题的复杂性的过程。

程序设计是根据给出的具体任务，编制一个能正确完成该任务的计算机程序。计算机程序是有序指令的集合，或者说是能被计算机执行的具有一定结构的语句的集合。

目前程序设计方法主要有面向过程的结构化程序设计和面向对象程序设计，本书只介绍面向过程的结构化程序设计。面向过程的结构化程序设计的主要思想是自顶向下、逐步求精、模块编程。即当一个程序十分复杂时，可以将它拆分成一系列较小的子程序，直到这些子程序易于理解结束。每个子程序是一个独立模块，每个模块又可继续划分为更小的子模块，程序具有一种层次结构。采用自顶向下逐步求精的设计方法就是先设计顶层，然后逐渐深入，逐层细分，逐步求精，直到整个问题可用程序设计语言明确地描述出来为止。结构化程序设计采用单入口单出口的控制方式，程序由顺序、分支选择、循环三种基本控制结构组成。这三种基本结构的特点都是：每一种结构只有一个入口和一个出口，任何一个问题都可以用这三种基本结构实现，任何复杂的程序都可以分解为由这三种基本结构组成。

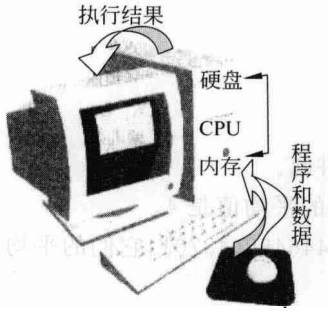


图 1.1

结构化程序设计使得程序结构清晰，可读性好，在出现问题时，便于查错，易于修改，提高了程序设计的质量。

图 1.1 所示的是一个简化了的计算机工作过程示意图，计算机的实际工作过程当然比这复杂得多，但它还是完整地体现了其基本工作原理，尤其体现了“软件指挥硬件”这一基本思想。在整个过程中，如果没有软件程序，计算机什么也干不了，可见软件程序多么重要。而且，如果软件程序编得好，计算机就运行得快且结果正确；程序编得不好，则可能需要运行很久才出结果，且结果还不一定正确。程序是软件的灵魂，CPU、显示器等硬件必须由软件指挥，否则它们只是一堆没有灵性的工程塑料与金属的混合物。在这里就是要教会读者怎样用编程语言又快又好地编写程序(软件)。

计算机直接能够读懂的语言是机器语言，也叫做机器代码，简称机器码。这是一种纯粹的二进制语言，用二进制代码来代表不同的指令。

下面这段程序是用通常使用的 x86 计算机的机器语言编写的，功能是计算 1+1。

```
10111000
00000001
00000000
00000101
00000001
00000000
```

这段程序看起来像“天书”。在用按钮开关和纸带打孔的方式向计算机输入程序的时

代，程序员编写的都是这样的程序。很明显，这种程序编起来很费力气，很难读懂。从那时候起，让计算机能够直接懂得人的语言就成了计算机科学家们梦寐以求的目标。

有人想出了这样的办法，编一个可以把人类的语言翻译成计算机语言的程序，这样计算机就能读懂人类语言了。这说起来容易，做起来难。就拿计算 1+1 来说，人们可以用“1+1 等于几”，“算一下 1+1 的结果”，“1+1 得多少”……多种说法，再加上英语、法语、日语、韩语、俄语等来描述。如果想把这些都自动转换成上面的机器码，是可望而不可即的事。所以人们退后一步，打算设计一种中间语言，它还是一种程序设计语言，但比较容易翻译成机器代码，且容易被人学会和读懂，于是诞生了“汇编语言”。

用汇编语言计算 1+1 的程序如下：

```
MOV  AX, 1
ADD  AX, 1
```

这个程序的功能是什么呢？从程序中 ADD 和 1 的字样，或许能猜个大概。没错，它还是计算 1+1 的。这个程序经过编译器（也是一个程序，它能把 CPU 不能识别的语言翻译成 CPU 能直接识别的机器语言）编译，就会自动生成前面的程序。这已经是很大的进步了，但并不理想。这里面的 MOV 是什么含义？好像是 Move 的缩写。这里的 AX 又代表什么？这是一个纯粹的计算机概念。从这个小程序，能看出汇编语言虽然已经开始贴近人类的语言，但还全然不像人们所期望的那样，里面还有很多计算机固有的东西必须要学习。它与机器语言的距离很近，每行程序都直接对应上例的三行代码。以后有机会学习、使用汇编语言，到那时将学到更多有关计算机内部的知识。

程序设计语言要无限地接近自然语言，所以它注定要不停地发展。此时出现了一道分水岭，人们把机器语言和汇编语言称为低级语言，把以后发展起来的语言称为高级语言。低级语言并不比高级语言“低级”，而是说它与计算机（硬件）的距离的级别比较低。高级语言高级到什么程度呢？这里先介绍一个很著名的 BASIC 语言，看它是怎样完成 1+1 计算的。

用 BASIC 语言计算并显示 1+1 的内容如下：

```
PRINT 1+1;
```

英文 PRINT 的中文意思是打印。比起前两个例子，它确实简单了不少，而且功能很强。前两个例子的计算结果只保存在 CPU 内，并没有输出给用户。这个例子直接把计算结果显示在屏幕上，它才是真正功能完备的程序，从这个例子相信你已经体会到了高级语言的魅力吧。

[阅读材料] 目前较流行的几种高级语言简介

因为高级语言易学、易用、强大，所以它发展很快，其种类之多完全可以用“百花争艳”来形容。据一位民间人士耗时多年的不完全统计，目前已经有超过 2500 种计算机语言，其中绝大多数都是高级语言。

BYTE 杂志创刊 20 周年特刊里一篇题为“程序设计语言发展简史”的文章中，列举了 1946 年起到 1995 年为止，在社会上产生一定影响的程序设计语言约有四十几种，实际上目

前广泛流行的只有几种。图 1.2 是日本计算机教育家三田典玄先生提供的一幅这几种程序设计语言与软件、硬件、系统、用户四大领域的关系图。

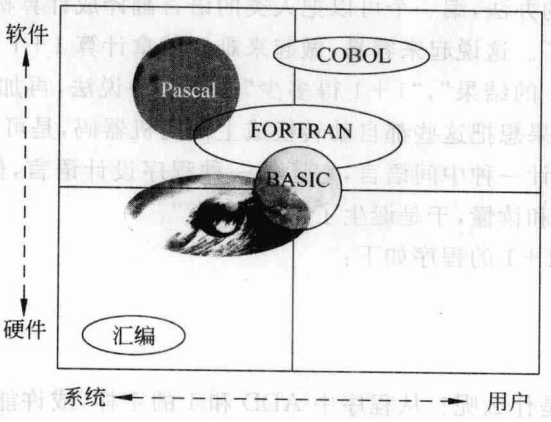


图 1.2

由图中不难看出，BASIC 语言正好处于中间位置，它兼顾了软件、硬件、系统和用户四大领域的要求，几乎成了其他任何一种语言无法替代的语言。几种常用的高级语言，在漫长的发展过程中互相渗透，互相借鉴，逐步形成了你中有我、我中有你的局面，其中变化最大的首当 BASIC 语言，它最初是从 FORTRAN 语言脱胎而来，以后又陆续吸收了其他语言的诸多特点。国外一些计算机教育家早已注意到了这一点，他们发现已掌握 BASIC 的人可以很快地学会任何一门（哪怕是很晦涩的）程序设计语言。都公认 BASIC 语言易学、易用、发展快、变化大、覆盖面广，是初学者常用的入门语言。美国、日本等发达国家都选择 BASIC 作为程序设计课程的入门语言，不无道理。我国在 20 世纪 80 年代以前，也几乎都是选择 BASIC 语言作为程序设计课程的入门语言。

到了 20 世纪 90 年代，BASIC 逐步被 QBASIC 所替代。QBASIC 是纯粹的面向过程的解释型语言，内容又过于烦琐，特别是它不能编译，从而不能脱离系统运行，使用十分不便。当前，在我国几乎清一色地选择了 C 语言作为程序设计课程的入门语言。C 语言是面向过程的编译型语言，从图 1.2 中不难看出，C 语言所处的位置是偏向硬件和系统的，由于 C 语言距离机器语言比其他高级语言更近，因此也有人把 C 语言看成是唯一的介于低级语言和高级语言之间的中级语言。它距离自然语言比较远，其思维还是顺应计算机而不是顺应人的，所以学起来比较困难，用起来也不简单，虽然很适合程序员使用，但对于缺乏计算机语言基础的程序设计课程的初学者来说，其难度之大可想而知。

在当今世界，C 语言除其固守的系统软件开发阵地和历史延续下来的大型软件外，几乎仅在小型的且追求运行效率的软件和嵌入式软件开发方面还有一定空间。况且，就这一点点空间也正在逐渐缩小。取而代之的是 C++、Java 和 C# 以及一些很专门化的语言。

C++ 从名字上看就知道其与 C 语言有着不解的渊源。C++ 几乎完全兼容 C 语言的语法，且所有流行的 C++ 编译器，都能编译 C 程序，所以很多人认为 C++ 就是 C 语言的升级。在很多场合，它俩也被放在一起，称为 C/C++。这个“++”里第二个加号加上的，便是大名