普通高等教育“十一五”国家级规划教材

21世纪计算机科学与技术实践型教程

丛书主编 陈明

谢书良 编著

C++程序设计任务导引教程

清华大学出版社

普通高等教育“十一五”国家级规划教材

谢书良 编著

C++程序设计任务导引教程

清华大学出版社
北京

21世纪计算机科学与技术实践型教程
丛书主编 陈明

内 容 简 介

本教材是为从未学习过编程又期望能简易掌握编程的读者编写的入门教材。

全书共分 13 章，第 1 章至第 5 章介绍结构化编程，第 6 章至第 10 章介绍对象化编程，第 11 章至第 13 章介绍可视化编程，内容涵盖了 C、C++ 、VC++ 的主要内容。第 13 章是体现本书特色的一章，分别介绍单数据表和多数据表的“学生成绩管理系统”的设计过程和完整代码，为最后进行“课程实践”提供了两个可视化程序设计的工程样例。

本书按任务导引教学方法进行编写，十分注重可读性和可用性。用任务来带基础知识，既保持了知识的系统性，又使学习目的比较明确，学习效果容易检验，在激发读者学习程序设计应用知识和训练程序设计能力方面有较好的作用。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。

本书可作为各级各类高等院校涉及程序设计的相关专业开设“C++ 程序设计”课程或“C++ 工程实践”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

C++ 程序设计任务导引教程 / 谢书良编著 . --北京：清华大学出版社，2012.6
(21 世纪计算机科学与技术实践型教程)
ISBN 978-7-302-27608-1

Ⅰ. ①C…　Ⅱ. ①谢…　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(2011)第 271244 号

责任编辑：谢　琛　郑　涛
封面设计：常雪影
责任校对：时翠兰
责任印制：杨　艳

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：北京市人民文学印刷厂
装 订 者：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：26.75　　**字　　数**：614 千字
版　　次：2012 年 6 月第 1 版　　**印　　次**：2012 年 6 月第 1 次印刷
印　　数：1～3000
定　　价：43.00 元

产品编号：044114-01

《21世纪计算机科学与技术实践型教程》

编辑委员会

《21世纪计算机科学与技术实践型教程》

序

21世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制定科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编　陈明

2005年1月于北京

前　言

当前，IT 技术迅猛发展，日新月异。在计算机应用日益广泛的形势下，软件的概念和程序设计的应用知识已逐渐成为人们渴求的新目标。如果说数学是“培养抽象思维的工具”，物理学是“培养逻辑思维的工具”，那么，程序设计则是“培养计算思维的工具”。有人预言，到 2050 年“计算思维”将成为全人类的主要思维方式，“计算思维”的精髓是“程序思维”，仅从此点而言，对于绝大多数理工类的高校学生来说，学一点程序设计基础和应用知识十分必要。

C++ 语言是目前广泛使用的一种程序设计语言。本教材，涵盖了 C++ 面向过程程序设计、C++ 面向对象程序设计和 VC++ 可视化程序设计基础内容。

本教材采用“任务导引法”思路进行编写，即在教师的具体指导之下，引发学生的学习兴趣和学习动机。这样既有利于维护教学内容的体系，也便于检测教学进程的效果。

考虑到各级各类学校对教材的要求有所不同，本教材采取了多项措施将教学起点降至最低，以帮助没有编程基础的初学者顺利迈进编程神殿的大门。同时，根据“学以致用”的原则，特别强化了“综合应用”这一章，考虑到本教材主要在低年级使用，可视化程序设计部分只围绕着简易的数据库编程进行，以此作为可视化编程的入门级锤炼。教材选择了“学生成绩管理系统”为示范项目，用以培养兴趣，激发创意，为读者今后继续学习有关内容打好基础。

“多思考，勤上机”是学好程序设计课程的关键，为了强化实践教学，本教材前 10 章的实验对每次上机的目的、内容等项目均有明确的要求；后 3 章对实践能力的要求相对要高，则以“课程实践”的形式进行。考虑到使用简便，仍建议用 VC++ 6.0 作为上机环境；教材介绍了如何将程序代码向 C++ 新标准转化，因此用 VC++ 8.0(即 VC++ 2005)或 VC++ 9.0(即VC++ 2008)作为上机环境也未尝不可。

前 12 章均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。提供全部题目的答案，以便于师生参考。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使用。

用“任务导引”的方式将面向过程程序设计、面向对象程序设计和可视化程序设计的一部分内容合编一书，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，此教材虽已进行过多年的试用，进行过反复修改，不足之处仍在所难免，请使用者不

吝赐教，使其不断完善。

学习编程需要细心、耐心、恒心，只有有志气、有毅力的人，才能饱尝编程带来的愉悦。

作者的电子邮箱地址为：xiening0705@126.com，欢迎来函联系。

谢书良

2011年12月

目　　录

第 1 章　程序设计概述

学习要点

- 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。
- 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，几乎可以不假思索地说出它们的平均值是 4；如果这两个数是 23 763 965 432 和 8 456 234 445 446 456，它们的平均值是多少？那只能由计算机去完成。

不管是通过什么方式，人和计算机的计算都是按照如下步骤。

(1) 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，必须由人来指导它。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，这个编制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。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、模块化编程。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为止。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。

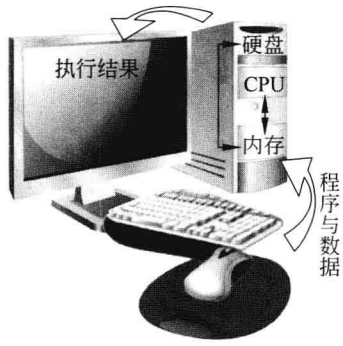

图 1-1　计算机工作过程

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 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”这一根本思想。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。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。

下面这段程序是用我们通常使用的 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”等多种说法，再加上用英语、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，是可望而不可及的事。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 的程序如下所示：

```
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，或许我们能猜个大概。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？那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，但还全然不像我们所期望的那样，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，每一行都直接对应上例的三行代码。以后你有机会学习、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？下面先介绍著名的 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感受到了高级语言的魅力。

[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。

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，几乎成了任何其他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

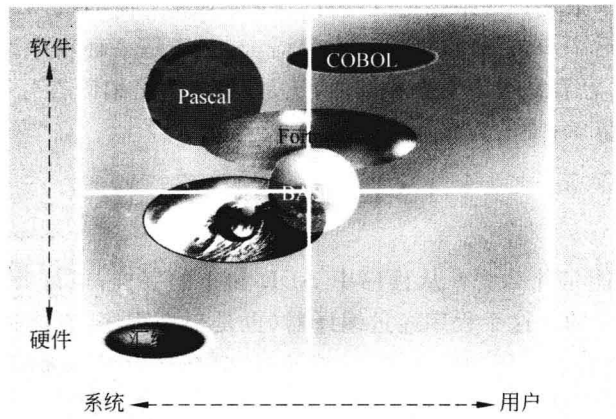

图 1-2 几种程序设计语言与软件、硬件、系统、用户的关系

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 语言是易学、易用、发展快、变化大、覆盖面广的初学者的入门语言。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。我国在二十世纪八十年代以前，也几乎都是选择 BASIC 语言作为程序设计课程的入门语言。

到了二十世纪九十年代，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，所以学起来比较费劲，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，便是大名鼎鼎的“面向对象”。

在面向对象被广泛接受之前，流行的是结构化程序设计思想。C、Pascal 等语言都是结构化的。面向对象的程序设计思想对程序设计的影响可谓翻天覆地，整个计算机界的思维都因其而改变，程序设计更加贴近现实世界，更加符合人类固有的习惯。C++ 借助 C 语言的影响力，携面向对象的威力，其份额迅速增大。在 20 世纪 90 年代，曾有民间组织