普通高等教育“十一五”国家级规划教材 计算机系列教材

程序设计基础
——以C++为例

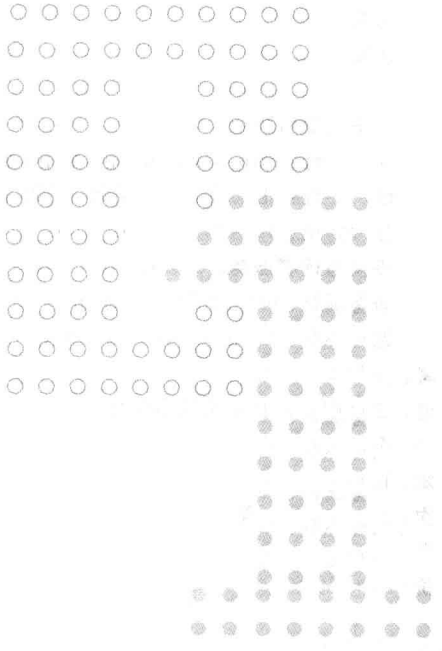
虞 歌 编著

清华大学出版社

计算机系列教材

虞 歌 编著

程序设计基础
——以C++为例



清华大学出版社
北京

内 容 简 介

本书是以C++语言作为入门语言的程序设计教材，以崭新的思路进行设计和编排。全书以程序设计零起点读者为主要对象，以培养程序设计能力为目标，循序渐进，通过数百个例题，重点讲解程序设计思想和方法，力图将C++语言基础知识介绍和程序设计能力培养完美结合。本书共10章，包括程序设计概述、C++基础、函数、复合数据类型、对象和类、对象和类的进一步学习、继承和多态、文件、常用数据结构以及标准模板库。

本书可作为高等学校学生学习程序设计课程的教材，也可供程序员和编程爱好者参考使用。

图书在版编目(CIP)数据

程序设计基础：以C++为例/虞歌编著. --北京：清华大学出版社，2013
计算机系列教材
ISBN 978-7-302-34228-1

Ⅰ. ①程…　Ⅱ. ①虞…　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(2013)第246344号

责任编辑：白立军　沈　洁
封面设计：常雪影
责任校对：时翠兰
责任印制：刘海龙

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**：32　　**字　　数**：778千字
版　　次：2013年12月第1版　　**印　　次**：2013年12月第1次印刷
印　　数：1～2000
定　　价：49.00元

产品编号：055834-01

随着信息产业的迅速发展，软件人才的需求量也越来越大。程序设计是软件人才必备的基础知识和技能。

程序设计基础是一门理论与实践密切相关，以培养学生程序设计能力为目标的课程。如何消除学生学习程序设计的畏难情绪，使学生顺利进入程序设计的大门，逐步掌握程序设计思想和方法，提高实践动手能力，是本课程教学的难题。

程序设计既是科学，也是艺术。学习程序设计是一件非常辛苦的事情，要有非常强的耐心和实践精神，需要花费大量的时间，不可能一蹴而就，必须从某个起点开始循序渐进。

本书就是一个很好的起点，以程序设计零起点读者为主要对象，采用标准C++语言(C++ 03)作为程序设计的描述语言，并加入了最新C++语言标准(C++ 11)的部分内容。C++语言是目前业界使用最广泛的程序设计语言，作者确信选用C++语言作为程序设计基础课程的教学语言是正确的选择。在多年的教学实践中，作者深深感到C++语言的灵活和高效，能够带给软件开发者无尽想象的空间，同时也深深感到讲授C++语言过程中面临的困难和挑战，意识到在程序设计基础课程中讲授C++语言并不是那么容易的。C++语言是一门复杂的程序设计语言，是为软件开发者设计的，而非为初学者设计的，向初学者讲授C++语言时必须很好地控制其固有的复杂性。

尽管目前有关学习C++语言的书籍很多，但学习C++语言仍然让大多数初学者心存畏惧。作者一直从事程序设计方面的教学和科研工作，主讲过多门程序设计课程，积累了丰富的教学经验。结合自己学习和使用C++语言的经验和感悟，以程序设计为主线，通过数百个例题，简洁通俗地讲解程序设计思想和方法，并穿插介绍相关的语言知识，循序渐进培养学生的程序设计能力。本书对那些渴望掌握C++语言而又心存畏惧的初学者是一个很好的选择。

教学改革的重点之一，就是要抓学生实践动手能力的培养。学生的能力是决定就业的根本，而就业率又是体现教育质量的重要指标。作为国内首家服务外包本科学院以及教育部、商务部在江苏、浙江两省开展地方高校计算机学院培养服务外包人才试点工作单位，我们实施了程序设计课程的教学改革，在教学内容、教学方法、教学手段和考核方式上，基本形成了比较完整的体系，目的就是培养学生的程序设计能力，适应社会对软件服务外包人才培养的需求。本书源于教学改革和教学实践，体现了程序设计教学改革的成果。

全书共有10章，各章内容安排如下：第1章程序设计概述，介绍程序设计基本概念和C++语言的基础知识。第2章C++基础，介绍标识符、数据类型、运算符、表达式、语句、标准库函数的使用、指针、引用以及程序设计错误、程序设计风格。第3章函数，内容包括函数定义、函数调用、函数声明、内联函数、函数默认参数、函数重载、函数模板、递归函数、指针和函数、引用和函数、Lambda表达式以及异常处理、程序结构。第4章复合数据类型，内容包括数组的基本概念、数组的声明和使用、数组作为函数参数、数组和指针的关系、排序和查找、

array数组、动态内存分配以及字符串。第5章对象和类，内容包括面向对象程序设计、类、构造函数和析构函数、静态成员、类作用域、对象作为函数参数、对象数组、异常类以及编写多文件程序。第6章对象和类的进一步学习，内容包括拷贝构造函数、this指针、对象组合、友元函数和友元类、运算符重载以及类模板。第7章继承和多态，内容包括继承的概念、单继承、多继承、多态的概念、虚函数、抽象类。第8章文件，内容包括文件的基本概念、文件操作、文件输入输出以及文件定位。第9章常用数据结构，内容包括数组类Vector、栈类Stack、链表以及队列类Queue。第10章标准模板库，内容包括标准模板库的概念、容器、容器适配器、迭代器、函数对象以及算法。

写给教师

跟各位尊敬的老师一样，作者从事了多年程序开发和教学工作，深知教学的艰辛，为教学方法费劲了心机，为教学效果伤透了脑筋，为学生的学习成绩摇头惋惜……所有这一切，从主观上说，与教师素质有关、与教学方法有关、与学生的学习态度有关；而在客观上，与教材和教学内容组织更有着重大的关联。编写本书的目的，就是企图从客观的角度，使教师的教学热情有更好的着力点，从而更顺利地完成既定的教学目标，让教师和学生的辛劳能得到应有的回报。

每章后面都有精心设计的实验题，可以据此来布置实验内容，使学生从第1周起就练习编程，并贯穿始终。许多实验题与例题有关联。实验题都有运行结果，较难的实验题有编程提示和程序代码，便于学生检验自己编写的程序。

实验所用操作系统可以是Windows、Mac OS X或UNIX/Linux，建议使用业界流行的C++语言集成开发工具(例如Visual Studio、Xcode或Code::Blocks的较新版本)，支持利用项目来实现程序的多文件组织。彻底抛弃非常陈旧的Visual C++ 6.0。

在讲授过程中，应该完全遵循C++语言标准，避免程序依赖任何特定的计算机、操作系统和编译器。要注重培养学生养成良好的编码风格，强调程序的可读性。在编写程序时，采用一种统一的、良好的编码风格是非常重要的。本书程序全部采用一种符合业界规范的编码风格。

我们所面对的教学对象，绝大多数是第一次接触程序设计的学生，很多学生对学习程序设计有较大的畏难情绪。培养学生程序设计能力不外乎3点：兴趣、天赋和经验。经验可以通过实践加以积累，天赋不容易改变，所以程序设计课程教学唯一能起作用的就是提高学生对编程的兴趣。

以往的程序设计教学多以讲授语法为主，没有引导学生如何合理优美地使用语言来解决实际问题。对初学者来说，语法内容讲得太多，难以在短时间内理解和消化。更何况，在课堂教学中，不可能面面俱到地讲授语法知识。这就造成了多数学生在学习结束后仍不会编程。这种只造砖、不盖房的教学方式，本身就造就了学生这样的学习态度，使学生感觉就

是为了考试过关、为了学分而学习，所以学习兴趣不大，主观能动性缺乏，学过的东西也很少能对将来的就业和工作有帮助，因此，真正能够达到课程期望目标的学生少之又少。

本书略去过多的语法细节和实际很少使用的语言特性，通过“例题→知识点说明→模仿编程→实践提高”的教学方式，让学生首先得到成功的经历，尽快体验成功的喜悦，逐渐喜爱看似枯燥的课程，增强学习的信心，激发求知欲。如果不仅会造砖，而且用造好的砖瓦亲手盖起了漂亮的房子，那么，学生就不会轻易将它们丢弃，而且还会长期拥有、使用和维护它们。这样，学到的东西就会真正成为他们自己的财富，而他们也将由此长期受益。那么，教师顺利、圆满地完成教学目标也就可以期待了。

写给学生

程序设计是一门实践性很强的学科，仅靠记概念、背原理是远远不够的。通常学生在上课时基本都能够听懂，可到了自己动手编程时，往往会觉得无从下手。要解决这个问题没有捷径可走，只有增加实践的时间和数量，熟能生巧，经历的多了，就会慢慢理解程序设计的思想，用过的方法多了，遇到问题才有可能想到解决的思路。

刚开始可能会感觉程序设计很难、很痛苦，可是当第一次编写出了一个小程序，就会觉得无比欣喜，回头看看其实也并不是那么难。解决了一个过去不能解决的问题，就是一种自我实现，就会有成就感，就会发觉自己原来这么有潜力。反过来，这种感觉会更加激发学习的热情，驱使自己去解决更多的难题，实现更多实用的任务，同时这些成果还可以作为提高自己工作效率的工具，岂不两全其美？这个过程也正是一条造就高手之路。所以学习程序设计一定要有信心、耐心和恒心，要实践，实践，再实践。

教学资源

读者要获取本书的相关资源，请访问清华大学出版社网站 http://www.tup.com.cn。

致谢

在本书写作过程中，参考了部分图书资料和网站资料，在此向其作者表示感谢。

本书的出版得到了清华大学出版社的大力支持，在此表示衷心的感谢。

征求建议与批评

感谢读者选择本书。由于作者水平和经验有限，书中难免有不足之处，恳请读者提出宝贵意见和建议，使本书日臻完善。作者联系方式：yuge@hznu.edu.cn。

第 1 章　程序设计概述

人们撰写文档、畅游因特网(Internet)时使用的文字处理器、浏览器都是在计算机上运行的软件,而软件是使用程序设计语言开发出来的。C++ 就是一种流行且功能强大的程序设计语言,很多软件都是用 C++ 开发的。本书将带领读者学习使用 C++ 。

1.1　程序设计基础

1.1.1　程序

1. 程序的定义

广义地说,程序是指为进行某项活动所规定的途径。

我们平时所说的日程安排、会议议程等,都是程序的实例。例如,学校要召开运动会,就需要事先编排好程序,从开幕式到闭幕式,每一项活动的时间、地点、人物、设施、规则、管理、协调等都必须有详细、周密的安排。

2. 程序的执行

程序的执行通常有三种方式。例如,在正常情况下,运动会按照程序所设定的顺序执行,这称为程序的顺序执行方式;如果遇到意外,例如下雨、运动员受伤等,还必须要准备相应的应急程序,也就是说有两套或多套方案供选择执行,这就是程序的选择执行方式;而当一项比赛有多组多人反复进行时,只需要一套程序反复执行即可,这就是程序的循环执行方式。

3. 计算机程序

算法是解决某个问题所需要的方法和步骤。

如果以计算机作为工具解决某个问题,必须将解决问题的方法和步骤(算法)告诉计算机。因为人无法与计算机直接交流,所以必须使用程序将算法表示成计算机能够理解的形式,然后让计算机执行程序来完成指定的任务。

计算机程序就是人们为解决某个问题用计算机可以识别的指令合理编排的一系列操作步骤。

1.1.2　程序设计语言

1. 程序设计语言的定义

程序设计语言,又称编程语言,是编写计算机程序所使用的语言。程序设计语言是人与计算机交互的工具,人要把需要计算机完成的工作告诉计算机,就需要使用程序设计语言编写程序,让计算机去执行。

2. 程序设计语言的发展

没有程序设计语言的支持,计算机无异于一堆废料。由于程序设计语言的重要性,从计

算机问世至今，人们一直在为研制更好的程序设计语言而努力。程序设计语言的数量在不断增加，各种新的程序设计语言不断问世。

程序设计语言的发展过程是其功能不断完善、描述问题的方法越来越贴近人类思维方式的过程。越接近自然语言的程序设计语言，就越“高级”，反之就越“低级”。越低级的语言，学习和使用难度就越大。

程序设计语言包括机器语言、汇编语言和高级语言。汇编语言和机器语言一般被称为低级语言。

① 机器语言是计算机诞生和发展初期使用的语言。机器语言采用二进制编码形式，由0、1组成，如0000010000001110，是计算机唯一可以直接识别、直接运行的语言。机器语言的执行效率高，但不易记忆和理解，编写的程序难以修改和维护，所以现在很少直接用机器语言编写程序。

② 为了减轻编写程序的负担，20世纪50年代初发明了汇编语言。汇编语言和机器语言基本上是一一对应的，但在表示方法上作了根本性的改进，引入了助记符，例如，用ADD表示加法，用MOVE表示传送等。汇编语言比机器语言更加直观，容易记忆，提高了编写程序的效率。计算机不能够直接识别和运行用汇编语言编写的程序，必须通过一个翻译程序将汇编语言转换为机器语言后方可执行。

③ 世界上第一个高级程序设计语言是20世纪50年代由John Backus领导的一个小组研制的FORTRAN语言。高级语言与人们日常熟悉的自然语言和数学语言更接近，便于学习、使用、阅读和理解。高级语言的发明，大大提高了编写程序的效率，促进了计算机的广泛应用和普及。计算机不能够直接识别和运行用高级语言编写的程序，必须通过一个翻译程序将高级语言转换为机器语言后方可执行。常用的高级语言有C、Objective—C、C++、Java和C#等。

3. 语言处理程序

计算机只能直接执行机器语言程序，用汇编语言或高级语言编写的程序都不能直接在计算机上执行。因此计算机必须配备一种工具，它的任务是把用汇编语言或高级语言编写的程序翻译成计算机可直接执行的机器语言程序，这种工具就是“语言处理程序”。语言处理程序包括汇编程序、解释程序和编译程序。

① 汇编程序把汇编语言编写的程序翻译成计算机可直接执行的机器语言程序。

② 解释程序对高级语言编写的程序逐条进行翻译并执行，最后得出结果。也就是说，类似于外文翻译中的口译，解释程序对高级语言编写的程序是一边翻译，一边执行的。下次执行同样的程序时，还必须重新翻译。

③ 编译程序把高级语言编写的程序一次性翻译成计算机可直接执行的机器语言程序。类似于外文翻译中的笔译，翻译一次就可以反复使用，以后每次运行同样的程序时，无需重新翻译。

1.1.3 程序设计

1. 程序设计的定义

程序设计，又称编程，是指编写计算机程序解决某个问题的过程。专业的程序设计人员常被称为程序员。

进行程序设计必须具备四个方面的知识。

(1) 领域知识

这是给出解决某个问题算法的基础。例如,要解决素数判断的问题,必须了解素数的概念以及素数判断的数学方法,这就是领域知识。如果程序员不具备解决某个问题的领域知识,是不可能编写出解决该问题的计算机程序的。

(2) 程序设计方法

程序设计方法是指合理编排计算机程序内部逻辑的方法。程序员在具备领域知识的基础上,还必须掌握某种程序设计方法,运用适当的思维方式,构造出解决某个问题的算法。

(3) 程序设计语言

要使用计算机解决某个问题,程序员还必须掌握某种程序设计语言(如 C++ 语言),运用程序设计语言将算法转换为计算机程序。

(4) 程序设计工具

程序员在设计程序时,为了提高程序设计的效率和程序的质量,通常需要使用某种程序设计工具。例如,在运用 C++ 进行程序设计时,可以使用微软公司的 Visual Studio,苹果公司的 Xcode,也可以使用开源的 Code::Blocks。

2. 程序设计过程

程序设计过程应当包括分析、设计、编码、测试、维护等不同阶段。

① 分析阶段的主要任务是了解问题的背景,理解问题的需求。

② 设计阶段的主要任务是针对问题的需求,设计出解决问题的算法。

③ 编码阶段的主要任务是运用某种程序设计语言,将算法转换为程序。

④ 测试阶段的主要任务是对程序进行严格的测试,保证程序的质量。只有通过测试的程序才能够交付使用。

⑤ 程序在交付使用后,就进入了维护阶段。人都有可能犯错误,程序是人的智力产品,无论经过怎样的严格测试,程序中通常还是会存在错误的。在程序交付使用后,也有可能发现程序中的错误,就需要对程序进行修改。在程序使用过程中,可能会对程序的功能提出新的需求,为了实现新的需求,也需要对程序进行修改。诸如此类的修改工作通常称为对程序的“维护”。

程序设计初学者常犯的错误是拿到问题就开始编程,忽略了程序设计过程中的分析和设计阶段,这会严重影响程序设计的质量。初学程序设计时,教材中的题目都比较简单,从这些题目中很难体会到上述程序设计过程的作用,尽管如此,也不应该直接开始编程,而应该首先分析问题的需求,设计出合理的算法。

3. 程序设计好坏的界定

程序设计有非常严格的语法规则和很强的逻辑顺序,所以需要熟练掌握和深入理解相应的语法规则,并进行大量的逻辑思维训练和编程实践,才能够设计出好用并可靠的计算机程序。

在保证程序正确的前提下,可读、易维护、可移植和高效是程序设计的首要目标。

可读是指程序清晰,具有良好的书写风格,没有太多繁杂的技巧,能使他人容易读懂。可读性是程序维护的基础,如果很难读懂程序,则无法修改程序。

在程序使用过程中,可能会对程序的功能提出新的需求,为了实现新的需求,需要对程