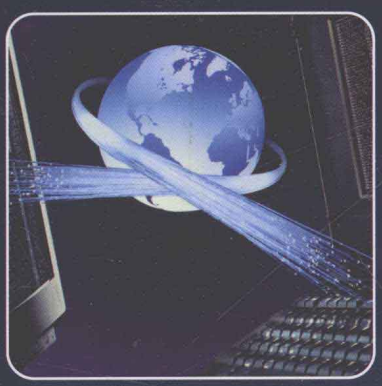
高等教育计算机学科“应用型”规划教材

C/C++程序设计教程

——面向对象分册（第2版）

郑秋生 主 编
王黎明 主 审



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等教育计算机学科“应用型”规划教材

C/C++程序设计教程——面向对象分册（第2版）

郑秋生　主　编
刘凤华　王文奇　李晓宇　副主编
王黎明　主　审

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

《C/C++程序设计教程》系列教材分为面向过程和面向对象两个分册。

面向对象分册详细阐述了 C++语言中面向对象程序设计的语法和思想。主要内容包括类和对象、继承与派生、多态性、输入/输出流、异常处理及命名空间、模板、标准模板库 STL 介绍及应用，以及面向对象程序设计实例。书中通过流行的 UML 工具描述 C++类，内容讲解清晰、实例丰富，力避代码复杂冗长，注重程序设计思想。简短的实例和 UML 图特别有助于初学者更好地理解、把握解决问题的精髓，帮助读者快速掌握面向对象程序设计的基本方法。

本书的特点是实例丰富，重点突出，叙述深入浅出，分析问题透彻，既有完整的语法，又有大量的实例，突出程序设计的思想和方法，将 C 语言程序设计和 C++程序设计有机地统一。特别适合作为计算机学科各应用型本科、专科的 C 语言程序设计和 C++程序设计的教材，也可作为其他理工科各专业的教材及相关技术人员的自学参考书。

图书在版编目（CIP）数据

C/C++程序设计教程. 面向对象分册/郑秋生主编. —2 版. —北京：电子工业出版社，2012.1
高等教育计算机学科“应用型”规划教材
ISBN 978-7-121-15489-8

Ⅰ. ①C… Ⅱ. ①郑… 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 263272 号

策划编辑：何 况
责任编辑：李 蕊
印 刷：涿州市京南印刷厂
装 订：涿州市桃园装订有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：18.75 字数：480 千字
印 次：2012 年 1 月第 1 次印刷
印 数：4 000 册 定价：30.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

编 委 会

前　言

本书的主要作者都是有着丰富教学经验的一线教师，从事C/C++程序设计课程教学多年，深知学生在学习 C++程序设计这门课程后，对程序设计方法、算法设计、调试程序、习题解答的茫然和问题，因此本书在介绍理论知识、相关概念和语言语法时，始终强调其在程序设计中的作用，使语言语法与程序设计相结合。同类书籍大部分偏重于对语言语法和概念的介绍，虽然在书中有针对语法和知识点的程序实例，但学生对每章内容在实际程序设计中的作用缺乏了解，而本书每章后都附有针对性较强的应用实例分析，尽可能使初学者在学习每章的内容后即能独立设计程序、解决实际问题，而不至于无从下手。本书有以下五个鲜明特点：

第一，改变了传统的教学模式。先讲C语言程序设计，再讲C++对C语言的扩展、面向对象的程序设计。本教材将 C/C++语言的学习很好地融在一起，让读者把面向过程和面向对象的程序设计方法有机地结合在一起。面向过程和面向对象两分册都统一使用 Visual C++ 6.0 编译器。

第二，改变了传统教材以语言、语法学习为重点的缺陷，本教材从基本的语言、语法学习上升到程序的“设计、算法、编程、调试”层次。为了让学生更好地掌握程序开发思想、方法和算法，书中提供了大量简短精辟的代码，有助于初学者学习解决问题的精髓。在每章后都有一节关于程序综合设计的内容，有一个或多个较大的程序，以帮助学生更好地掌握程序设计方法和解决实际问题的能力。

第三，教材强调程序的设计方法，大量例题配有流程图、N-S 图和 UML 图，即突出程序的算法和设计，而不仅是语法和编程，培养学生程序设计能力和程序调试技能，养成好的编程习惯，为专业程序员的培养打下良好的基础。

第四，培养学生面向对象程序设计的能力，引导学生建立程序设计的大局观，帮助学生掌握从客观事物中抽象出 C++类的方法。通过系统的学习，使学生的编程能力上一个台阶，具备解决复杂问题的程序设计能力。

第五，根据当前实际大型软件项目开发的需要，加大了异常处理、模板等内容，新增 STL 标准模板库，并通过流行的 UML 工具设计 C++类。

本教材编写充分考虑了目前应用型本科 C/C++程序设计课程教学的实际情况和存在的问题。第一，学生在大一阶段的基础课程较多，不可能投入过多的精力来学习本门课程；第二，学生对这门课学习的期望值很高，但对学习时可能遇到的困难估计不足；第三，学生现有的上机实践条件大大改善，特别有利于贯彻先进的精讲多练的教学思想；第四，学生学会了语言的语法，仍不具备解决实际问题的能力，学生的程序设计、算法设计、编程、调试的能力相对较差。本教材作者正是考虑了学生的这些实际问题，从而精心编写了这一套面向应用型本科的 C/C++程序设计教程，特别适合于分两个学期系统讲授 C/C++程序设计。第 1 学期讲授面向过程分册，第 2 学期讲授面向对象分册。

本面向对象分册共分 8 章，第 1 章到第 3 章主要阐述面向对象程序设计的重要概念，包括类和对象、继承与派生、多态性；第 4 章介绍输入/输出流技术；第 5 章主要介绍异常的概

念、异常的产生及异常的处理机制；第 6 章和第 7 章介绍模板和 STL 标准模板库；第 8 章主要讲述面向对象的分析与设计方法，以实例的形式详细介绍如何用 C++进行程序设计。

为了方便使用本教材的教师备课，我们还提供了配套的电子教案，公开放在网站上，供任课教师自由下载使用。相信我们多年的教学经验会对广大师生的教和学有所帮助。建议本分册的教学学时为 60 学时，其中理论教学为 44 学时，课内上机实践为 16 学时，课外上机不少于 32 学时。

本教材的编写得到了河南省计算机学会的大力支持，组织了河南多所高校编写了高等教育计算机学科“应用型”系列教材。参编本教材的高校有中原工学院、郑州大学、河南科技大学、郑州轻工业学院。

本书由郑秋生任主编，第 1 章和附录由王海龙和夏敏捷编写，第 2 章由罗菁和潘惠勇编写，第 3 章由李晓宇编写，第 4 章由冀治航编写，第 5 章由郑秋生编写，第 6 章和第 7 章由王文奇编写，第 8 章由刘凤华编写。全书最终由郑秋生修改并统稿。郑州大学王黎明老师为本书提出改进意见，在此谨向他们表示衷心的感谢。

由于编者水平有限，时间仓促，书中难免有错，敬请广大读者批评指正，在此表示感谢。作者 E-mail：zqs@zzti.edu.cn。

编　者

2011 年 9 月

第 2 版改版说明

本书第 1 版是由河南省计算机学会和电子工业出版社组织教学一线教师，共同组织编写的一套面向应用型本科教学的教材，属于高等教育计算机学科应用型规划教材，教材充分体现了教师多年的教学研究成果和教学经验，具有较好的质量和社会影响力。2007 年出版后，在郑州大学、中原工学院、河南科技大学、郑州轻工业学院和河南工程学院等河南省高校推广使用，得到任课教师和读者的普遍欢迎。通过电子工业出版社和新华书店的努力，被国内的许多高校广泛使用，通过三年的使用，该教材的任课教师、学生和读者都给予了许多的肯定、鼓励，也提出了许多有意义的建议、意见和再版的意愿，在此代表所有作者感谢读者对教材的厚爱和关心。

为了更好地推进高等学校本科教学质量与教学改革工程，结合重点专业建设、精品课程和精品教材建设目标，以及培养学生的工程能力和创新能力的需要，对教材进行修订再版。第 2 版根据教学中发现的问题和读者的建议反馈，以及为了达到提升学生的程序设计能力、调试能力的目的再次进行了编写。第 2 版的主要变化表现在：

第 1 章到第 4 章对部分程序例题做了改变，更接近实际应用。

第 5 章增加了异常处理中各种情况的说明，同时简化了对特定情况（如没有捕捉异常和捕捉所有异常）的描述。

第 6 章完善了函数模板语法的说明，减少了函数模板和模板函数的区别描述。

第 7 章强化了对 vector 和 list 容器使用方式的描述，简化了关于容器的成员函数和结构的说明。

第 8 章用图书管理系统取代原来的电梯调度管理系统作为面向对象程序设计的综合性实例。对于初学者来说，原来的实例过于生涩、不好理解，修改后的实例浅显易懂、条理性更强。

教材中的语法和程序代码遵循 C99 国际标准，以适应 C/C++语言的发展。例如，按照 C99 标准的要求，main 函数的返回类型一律指定为 int 型，并在 main 函数的末尾加一个“return 0;”。

修订了第 1 版的一些印刷错误和不准确的表述方法。修改、增加了一些例题、习题和章节内容，希望本版能更加适合教师的教学。

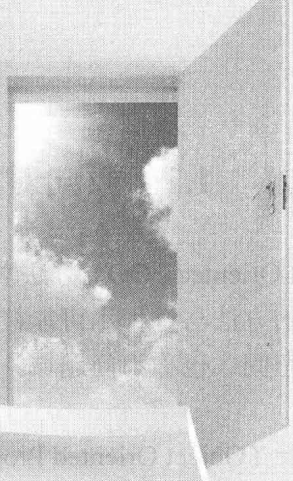
编　者

2011 年 9 月

人们纠结于财务问题的主要原因在于，他们在学校里待了很多年，却对金钱一无所知。结果便是他们学会了怎样为金钱工作，而不是让金钱为他们工作。

——财商专家罗伯特·清崎

目　录



第1章 类和对象

C++是一种面向对象程序设计语言，为面向对象技术提供了全面的支持。学习C++首先要认识类和对象，掌握它面向对象的特性和实现面向对象的方法。类是实现面向对象程序设计的基础，也是C++封装的基本单元，对象是类的实例。本章主要介绍类和对象的基本概念、类的定义和使用，以及类的一些特性。

通过本章学习，应该重点掌握以下内容：

- 面向对象程序设计的基本特点
- 类和对象的定义与使用
- 构造函数和析构函数
- 拷贝构造函数和堆对象、对象数组
- 友元函数和友元类
- 类的静态成员
- string类对象

1.1 面向对象程序设计概述

面向对象（Object Oriented，OO）方法学的出发点和基本原则是尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界、解决问题的方法与过程，也就是使描述问题的问题空间（也称为问题域）与实现解法的解空间（也称为求解域）在结构上尽可能一致。

面向对象程序设计（Object Oriented Programming，OOP）是软件系统设计与实现的方法，这种方法既吸取了结构化程序设计的绝大部分优点，又考虑了现实世界与面向对象空间的映射关系，所追求的目标是将现实世界的问题求解尽可能的简单化。在自然界和社会生活中，一个复杂的事物总是由很多部分组成的。例如，一个人由姓名、性别、年龄、身高、体重等特征描述；一个自行车由车轮、车身、车把等部件组成；一台计算机由主机、显示器、键盘、鼠标等部件组成。当人们生产一台计算机时，并不是先生产主机，然后生产显示器，最后生产键盘、鼠标，即不是顺序执行的。而是分别生产主机、显示器、键盘、鼠标等，最后把它们组装起来。这些部件通过事先设计好的接口连接，以便协调工作。例如，通过键盘输入可以在显示器上显示字或图形。这就是面向对象程序设计的基本思路。

1.2 面向对象程序设计的基本概念

面向对象程序设计方法提出了一些全新的概念，如类和对象、封装、继承和多态，下面分别讨论这几个概念。

1.2.1 类

类是面向对象程序设计语言的基本概念。在现实生活中，人们常把众多的事物归纳并划分为若干种类型，这是认识客观世界常用的思维方式。例如，人们把载人数量为 5～7 人的，各种品牌使用汽油或柴油的四个轮子的汽车统称为小轿车，也就是说，从众多的具体车辆中抽象出小轿车类。再例如，把一所高校所有在校的各个班级和各个专业的本科生、研究生统称为学生，可以从众多的具体在校人员中抽象出学生类。

对事物进行分类时，依据的原则是抽象，将注意力集中在与目标有关的本质特征上，而忽略事物的非本质特征，进而找出这些事物的所有共同点，把具有共同性质的事物划分为一类，得到一个抽象的概念。日常生活中的汽车、房子、人、衣服等概念都是人们在长期的生产和生活实践中抽象出来的概念。

面向对象方法中的“类”，是具有相同属性和行为的一组对象的集合，它为属于该类的全部对象提供了抽象的描述，其内部包括属性和行为两个主要部分。

1.2.2 对象

对象是现实世界中一个实际存在的事物，它可以是看得见、摸得到的物体（如一本书），也可以是无形的（如一份报告）。对象是构成现实世界的一个独立单位，它具有自己的静态特征（可以用某种数据来描述）和动态特征（对象所表现出来的行为或具有的功能）。例如，

张三是现实世界中一个具体的人，他具有身高和体重（静态特征），能够思考及做运动（动态特征）。

面向对象方法中的对象，是描述系统中某一客观事物的一个实体，它是构成系统的一个基本单位。对象由一组属性和一组行为构成。属性是用来描述对象静态特征的数据项，而行为是用来描述对象动态特征的操作序列。类和对象的关系就像模具与产品之间的关系，一个属于某类的对象称为该类的一个实例，如张三就是人这个类的一个实例，或是这个类的具体表现。

1.2.3 封装与数据隐藏

封装是指将数据和代码捆绑在一起，从而避免外界的干扰和不确定性。在 C++中，封装是通过类来实现的。类是描述具有相同属性和方法的对象的集合，定义了该集合中每个对象所共有的属性和方法。封装也是面向对象方法中的一个重要原则，它把对象的属性和行为结合成一个独立的系统单位，并且尽可能地隐藏对象的内部细节。这里有两层含义：第一是把对象的全部属性和全部行为结合在一起，形成一个不可分割的独立单元；第二是信息隐蔽，也就是尽可能隐蔽对象的内部细节，对外部世界形成一个边界或屏障，只保留有限的公用的对外接口，使之与外部世界发生联系。

1.2.4 继承

继承（inheritance）是面向对象程序设计能够提高软件开发效率的重要原因之一，也是软件规模化的一个重要手段。特殊类的对象拥有其一般类的全部属性和行为，称为特殊类对一般类的继承。

继承具有重要的现实意义，它简化了人们对于现实世界客观事物的认识和描述。例如，人们认识了汽车的特征之后，再考虑小轿车时，因为知道小轿车也是汽车，于是认为小轿车具有汽车的全部一般特征，从而可以把精力用于发现和描述小轿车不同于一般汽车的独有的那些特征。

软件的规模化生产是影响软件产业发展的重要因素，它强调软件的复用性，也就是程序不加修改或进行少许修改，就可以用在不同的地方。继承对于软件的复用具有重要意义，特殊类继承一般类，本身就是软件复用。不仅如此，如果将开发好的类作为构件放到构件库中，在开发新系统时可以直接使用或继承使用。

1.2.5 多态性

面向对象的通信机制是消息，面向对象技术是通过向未知对象发送消息来进行程序设计的。当一个对象发出消息时，对于相同的消息，不同的对象具有不同的反应能力。这样，一个消息可以产生不同的响应效果，这种现象称为多态性。

在操作计算机时，“双击鼠标左键”这个操作可以很形象地说明多态性的概念。如果发送消息“双击鼠标左键”，不同的对象会有不同的反应。例如，“文件夹”对象收到双击消息后，其产生的操作是打开这个文件夹；而“可执行文件”对象收到双击消息后，其产生的操作是执行这个文件；如果是音乐文件，会播放这个音乐；如果是图形文件，会使用相关工具软件打开这个图形。很显然，打开文件夹、播放音乐、打开图形文件需要不同的函数体，但是在这里，它们可以被同一条消息“双击鼠标左键”来引发，这就是多态性。面向对象程序设计

通过继承和重载两种机制实现多态性。

多态性是面向对象程序设计的一个重要特征。它减轻了程序员的记忆负担，使程序的设计和修改更加灵活，程序员只需要记住有限的接口就可以完成各种所需要的操作，程序中可以用简单的操作完成同一类体系中不同对象的操作。

1.2.6 消息

面向对象技术的封装使对象相互独立，各个对象要相互协作实现系统的功能则需要对象之间的消息传递机制。消息是一个对象向另一个对象发出的服务请求，进行对象之间的通信，也可以说是一个对象调用另一个对象的方法（method）或称为函数（function）。

通常，把发送消息的对象称为发送者，把接收消息的对象称为接收者。在对象传递消息中只包含发送者的要求，它指示接收者要完成哪些处理，但并不告诉接收者这应该如何完成这些处理。接收者接收到消息后要独立决定采用什么方式完成所需的处理。同一对象可接收不同形式的多个消息，产生不同的响应；相同形式的消息可送给不同的对象，不同的对象对于形式相同的消息可以有不同的解释，做出不同的响应。

在面向对象设计设计中，对象是节点，消息是纽带。应注意不要过度侧重如何构建对象及对象间的各种关系，而忽略对消息（对象间的通信机制）的设计。

1.3 类和对象的定义

前面介绍了类与对象的概念：类是对象的抽象，而对象是类的具体实例（instance）。C++提供了对数据结构和方法的封装与抽象，称为类（class）。类是C++封装的基本单元。可以将类理解成一种新的数据类型，这种数据类型中封装了数据的内容和对数据内容的操作。在使用结构体类型时，先定义一个结构体类型，再定义这种结构体类型的变量。类和类对象的使用情况与结构体类型相同，先定义一个类类型，然后用这个类去定义若干个该类类型的对象，也就是说对象是一种类类型对应的变量。

1.3.1 类的定义

类是用户自己定义的新类型。如果在程序中用到类类型，必须先根据自己的需要进行声明。类定义包含两部分：数据成员和成员函数（又称函数成员）。数据成员说明类的属性，成员函数是对类数据成员操作的类内函数，又称为方法。

类定义的一般格式为：

```
class  类名
{
  public :
    数据成员和成员函数实现
  protected :
    数据成员和成员函数实现
  private :
    数据成员和成员函数实现
};
```

说明：

（1）定义一个类时，使用关键字 class；类名必须是一个合法的变量名，C++中习惯用 C 或 T 开头，如 CStudent。

（2）一个类包括类头（class head）和类体（class body）两部分。class <类名>，称为类头。

（3）大括号中定义的是类的数据成员和成员函数，称为类体。类定义结束要用“;”号结束。

（4）关键字 public（公有）、protected（保护）和 private（私有）称为成员访问限定符（number access specifier）。用访问限定符声明各个数据成员和成员函数的访问权限。

在类声明中，三种访问限定符可以按任意次序出现，也就是说可以先声明私有成员，再声明公有成员；也可以先声明公有成员，再声明私有成员。三种访问限定符也可以多次出现，但是一个成员只能具有一种访问权限。在书写时通常习惯将公有类型放在最前面，这样便于阅读，因为它们是外部访问时所要了解的。

被声明为公有（public）的成员提供了与外界的接口功能。公有成员可以被本类中的成员使用和访问，还可以被类作用域外的其他函数使用。

被声明为私有（private）的成员是封装在类内部的，只能被该类的成员和该类友元函数或友元类访问，任何类以外的函数对私有成员的访问都是非法的。

被声明为保护（protected）的成员，访问权限介于私有和公有之间，类的成员可以访问保护成员，而类以外的其他函数不能访问保护成员，但该类的继承类可以访问。不同访问权限关系图如图 1.1 所示。

关于访问权限，可以举例说明如下：一个家庭住宅，客厅是公有的，家庭成员和外面的客人都可以进入。卧室是私有的，只有主人能进入自己的卧室，但如果主人有一个好朋友（友元），这个好朋友也被允许进入主人的卧室（友元可以访问私有成员）。这个家庭的存款可以定义为保护权限的成员，主人自己可以使用，外人不得使用，但主人的子女（主人的继承类）也可以使用这些存款（继承类可访问保护成员）。

一般情况下，一个类的数据成员应该声明为私有成员，这样封装性较好。一个类应该有一些公有的成员函数，作为对外的接口，否则其他代码无法访问类。

根据上面类定义的规则，定义编制一个求方程 $ax^2+bx+c=0$ 根的程序。假设这个类的名字为 CFindRoot，至少需要将方程的系数作为 CFindRoot 类的数据成员，可以将它们设计成 float 型。

为了方便，除了将方程系数设为类的数据成员之外，还将方程的根 x1 和 x2，以及用来作为判定条件的 delta 设计成类的属性，并且将方程的两个根设为 double 型，将 delta 设为 float 型。

成员函数 SetData 用来设置方程系数的值，并且计算出 delta 的值。求 delta 需要使用库函数 sqrt，该函数在头文件<cmath>中定义，只要包含它即可。

成员函数 Find 用来求方程的根，Display 函数则用来输出结果。

按以上描述画出 CFindRoot 的类图，如图 1.2 所示。