

## <<面向对象程序设计>>

### 图书基本信息

书名：<<面向对象程序设计>>

13位ISBN编号：9787310035007

10位ISBN编号：7310035003

出版时间：2010-8

出版时间：南开大学出版社

作者：赵宏 主编

页数：344

字数：566000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;面向对象程序设计&gt;&gt;

## 前言

目前, 不仅许多计算机专业和多数软件学院的程序设计课程选择了C++作为程序设计的第一门语言, 越来越多的理工科专业也把C++作为计算机基础课, 一方面是由于C++是应用最广的面向对象语言, 另一方面是由于它有利于初学程序设计的学生学习一般的编程技巧。

南开大学理工科公共计算机基础课程“面向对象程序设计”, 选用的就是C++语言, 该课程的目标是培养学生基本的程序设计能力。

然而, 公共计算机基础课课时有限, 近几年的教学经验表明, 把C++语言作为高级语言程序设计的教学语言对于教师和学生都是有难度的, 主要的问题是C++的规模和C++程序的复杂程度, 往往使刚刚步入大学的学生感到困难重重。

教材的编写一直是一个艰巨而具有探索性的工作。

面对非计算机专业的理工科学生, 如何使初学者在有限的课时内打下良好的程序设计基础, 目前仍有许多值得探索的地方。

由于他们不再有相关的后续课程, 进一步提高程序设计能力的学习机会较少, 公共计算机基础课教材的编写更是一个挑战。

国内外同类教材主要是面向计算机专业的学生, 即使是为非计算机专业的理工科学生编写的教材, 也仅仅是对专业教材的简单取舍, 较少地考虑非计算机专业的理工科学生学习程序设计的需求和特点, 在内容和程序实例的选取上, 在文字叙述上存在不足, 在教材的编写风格上也过于传统, 较难激发学生的学习兴趣。

《高等院校非计算机专业教材·面向对象程序设计: C++高级语言(附CD-ROM光盘1张)》是南开大学教材建设资助立项教材。

编著者力求做到: (1) 适应21世纪课程体系和教学内容改革方向要求, 抓住授课对象是非计算机专业的本科一年级学生的特点, 注重内容的选取和章节的安排。

力求做到学习内容循序渐进; 文字叙述简单、易于理解; 全书难易得当、重点突出, 适合非计算机专业学生和C++语言自学者学习程序设计语言时使用。

(2) 注重语法规则的清晰讲解和配备完整的程序实例。

加强程序实例的选择和比例配置, 训练和培养学生分析解决问题的思想和能力。

(3) 根据初学者容易出错的地方, 给出了大量的提示、提问和学习指导, 适合课堂教学和自学。

(4) 理论与实际相结合。

同时为《高等院校非计算机专业教材·面向对象程序设计: C++高级语言(附CD-ROM光盘1张)》编写了配套的实习指导和习题集, 根据各章节的内容给出相应的上机实习内容; 强化并丰富相应的习题, 以逐步提高学生的程序设计能力, 从而能够使用C++高级语言解决实际的问题。

## <<面向对象程序设计>>

### 内容概要

高级程序设计语言C++是在C语言基础上的一种能够进行面向对象程序设计和传统过程化程序设计的语言，是C语言的超集。

本书较详细地介绍了

C++语言的基础知识，通过大量的程序实例，针对初学者容易出现错误和困惑的地方提供了大量的附注，帮助读者更好地理解C++的基本概念和技术。

全书共分19章，分别介绍了程序设计的基本概念、C++程序的基本组成，数据类型、常量和变量，运算符、表达式和语句，程序控制结构，函数初步与变量的存储类型，数组，指针和引用，字符串，函数，构造数据类型，编译预处理，类与对象，继承，多态性，运算符重载，输入/输出流，文件，模板，MFC入门等内容。

本书是专门为高等院校非计算机专业C++高级语言程序设计课程编写的教材，面向C++初学者，不求读者已经熟悉相关的编程概念和有C语言方面的背景知识。

本书也适合自学者使用。

## &lt;&lt;面向对象程序设计&gt;&gt;

## 书籍目录

- 第1章 初识C++
  - 1.1 程序设计(Programming)的基本概念
  - 1.2 高级程序设计语言C++
  - 1.3 第一个C++程序
  - 1.4 小结
  - 1.5 学习指导
- 第2章 数据类型、常量和变量
  - 2.1 数据在计算机中存储的基本概念
  - 2.2 基本数据类型
  - 2.3 常量
  - 2.4 变量
  - 2.5 小结
  - 2.6 学习指导
- 第3章 运算符、表达式和语句
  - 3.1 运算符和表达式
  - 3.2 类型转换
  - 3.3 语句
  - 3.4 小结
  - 3.5 学习指导
- 第4章 程序控制结构
  - 4.1 顺序结构
  - 4.2 选择结构
  - 4.3 循环结构
  - 4.4 转向语句
  - 4.5 应用实例
  - 4.6 小结
  - 4.7 学习指导
- 第5章 函数初步与变量的存储类型
  - 5.1 函数的基本概念
  - 5.2 函数的定义
  - 5.3 函数的传值调用及函数原形
  - 5.4 变量的存储类型、作用域和生存期
  - 5.5 小结
  - 5.6 学习指导
- 第6章 数组
  - 6.1 数组的概念
  - 6.2 数组的声明
  - 6.3 数组的初始化
  - 6.4 数组的访问
  - 6.5 小结
  - 6.6 学习指导
- 第7章 指针和引用
  - 7.1 指针的概念
  - 7.2 指针变量的声明、初始化和访问
  - 7.3 指针与数组

## &lt;&lt;面向对象程序设计&gt;&gt;

- 7.4 指针的运算
- 7.5 指向指针的指针
- 7.6 cot指针
- 7.7 堆内存分配
- 7.8 引用
- 7.9 小结
- 7.10 学习指导
- 第8章 字符串
  - 8.1 字符串的概念
  - 8.2 字符串的输入输出
  - 8.3 指针与字符串
  - 8.4 常用的字符串函数
  - 8.5 小结
  - 8.6 学习指导
- 第9章 函数
  - 9.1 函数的调用机制
  - 9.2 函数的递归调用
  - 9.3 带默认形参值的函数
  - 9.4 函数的作用域
  - 9.5 内联函数
  - 9.6 函数重载
  - 9.7 函数指针
  - 9.8 函数与指针
  - 9.9 函数与引用
  - 9.10 小结
  - 9.11 学习指导
- 第10章 构造数据类型
  - 10.1 结构体
  - 10.2 枚举
  - 10.3 类型重定义typedef
  - 10.4 小结
- 第11章 编译预处理
  - 11.1 宏定义
  - 11.2 条件编译
  - 11.3 文件包含和多文件结构
  - 11.4 小结
- 第12章 类与对象
  - 12.1 类和对象的基本概念
  - 12.2 类的声明与定义
  - 12.3 对象
  - 12.4 构造函数和析构函数
  - 12.5 拷贝构造函数
  - 12.6 类的静态成员
  - 12.7 友元
  - 12.8 this指针
  - 12.9 string类
  - 12.10 应用实例

## &lt;&lt;面向对象程序设计&gt;&gt;

- 12.11 小结
- 12.12 学习指导
- 第13章 继承
  - 13.1 什么是继承
  - 13.2 派生类的定义
  - 13.3 函数重定义
  - 13.4 保护(protected)成员
  - 13.5 派生类的继承方式
  - 13.6 派生类的构造函数与析构函数
  - 13.7 类型兼容
  - 13.8 多重继承
  - 13.9 应用实例
  - 13.10 小结
  - 13.11 学习指导
- 第14章 多态性
  - 14.1 多态性的概念
  - 14.2 虚函数
  - 14.3 抽象类
  - 14.4 应用实例
  - 14.5 小结
  - 14.6 学习指导
- 第15章 运算符重载
  - 15.1 运算符重载的概念
  - 15.2 运算符重载的方法
  - 15.3 运算符重载的规则
  - 15.4 特殊运算符的重载
  - 15.5 类类型转换
  - 15.6 小结
- 第16章 输入/输出流
  - 16.1 输入/输出流简介
  - 16.2 输入/输出流的成员函数
  - 16.3 输入/输出的格式控制
  - 16.4 自定义数据类型的输入/输出
  - 16.5 小结
- 第17章 文件
  - 17.1 文件与流
  - 17.2 文件的打开与关闭
  - 17.3 文本文件的顺序读写
  - 17.4 二进制文件的顺序读写
  - 17.5 文件的随机读写
  - 17.6 小结
- 第18章 模板
  - 18.1 函数模板
  - 18.2 类模板
  - 18.3 小结
- 第19章 MFC入门
  - 19.1 认识MFC

## <<面向对象程序设计>>

- 19.2 一个简单的MFC应用程序
- 19.3 常用控件
- 19.4 MFC应用程序开发实例
- 19.5 小结

## &lt;&lt;面向对象程序设计&gt;&gt;

## 章节摘录

计算机程序 (Program) 的作用是指示计算机进行必要的计算和数据处理从而帮助我们解决特定的问题。

计算机语言要处理两个概念—数据和算法，数据是程序使用和处理的信

息。面对问题，需要找出解决问题的方法，我们把这种能够在有限的步骤内解决问题的过程和方法称为算法 (Algorithm)。

程序设计是指设计、编制、调试程序的方法和过程，是寻找算法并用计算机能够理解的语言表达出来的一种活动。

1.1.1 程序设计方法 20世纪60年代末期随着“软件危机”的出现，程序设计方法的研究开始受到重视。

结构化程序设计方法 (Structured Programming, SP) 是程序设计历史中最早提出的方法。

70年代中后期，针对结构化程序设计在进行大型项目设计时存在的缺陷，提出了面向对象程序设计 (Object Oriented Programming, OOP) 方法。

20年来面向对象程序设计方法的大量研究工作，逐步替代了传统的结构化程序设计方法，成为目前最重要的程序设计方法。

1. 结构化程序设计 (SP) SP方法的核心是将程序模块化，主要通过使用顺序、分支 (选择) 和循环 (重复) 等三种基本结构，形成具有复杂层次的结构化程序。

SP方法采用“自顶向下，逐步求精”的设计思想，其理念是将大型的程序分解成小型和便于管理的任务，如果其中的一项任务仍然较大，就将它分解成更小的任务。

程序设计的过程就是将程序划分成为小型的、易于编写的模块。

程序的模块功能独立，只使用三种基本结构，具有单一出口和入口，增加了模块的独立性，可以像搭积木一样根据需要使用不同的模块。

C语言的设计有助于使用结构化程序设计的方法，程序员开发程序单元 (称为函数) 来表示各个任务模块。

图1.1是采用结构化程序设计的程序结构示意图。

<<面向对象程序设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>