

<<C++与面向对象程序设计实践教程>>

图书基本信息

书名：<<C++与面向对象程序设计实践教程>>

13位ISBN编号：9787030266569

10位ISBN编号：7030266560

出版时间：1970-1

出版时间：科学出版社

作者：左伍衡，胡同森 编

页数：250

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>

前言

C++语言具有全面兼容C和支持面向对象的特点。

由于面向对象方法的突出优点，目前它已经成为应用最广泛的面向对象的程序设计语言之一。

针对以培养应用型人才为主要目标的普通高等院校相关专业的程序设计语言教学的实际情况，结合多年的一线教学经验积累，我们特别编写了本书。

本书在体系结构的确定和内容的选择上，坚持了定位明确、由浅入深、循序渐进、点面结合的方针，书写上力求简单明白，通俗易懂，所以本书的可读性、实用性和先进性俱佳，既适合教学，又适合自学。

本书各章节都提供了典型例题，以提高读者的编程能力和技巧。

书中所有程序示例均在visualC++6.0下进行了调试并运行通过。

本书分为基础实践篇和综合实践篇两大部分。

其中基础实践篇主要按照课程内容分为九章，每章包含知识要点、典型例题、典型习题与上机实验四个模块。

第1章，绪论：概要介绍了C++的发展历程、特点，C++程序结构及编程环节。

第2章，数据类型、表达式与语句：概要介绍了C++语言的基础知识，如数据类型、表达式和基本语句及其使用。

第3章，函数与程序结构：概要介绍了C++语言的函数和程序结构的相关概念及其运用。

第4章，数组、结构体与指针：总结了数组、结构体与指针等常用的用户自定义类型的知识点及其运用。

第5章，类和对象：概述了面向对象设计方法的核心概念——类和对象及其成员以及如何利用类来解决具体问题。

第6章，继承与派生：概述了类的继承特性，总结了不同继承方式下的基类成员的访问控制问题、构造函数和析构函数以及多重继承问题。

第7章，多态性：概述了多态性的概念及类型、函数重载、运算符重载、虚函数、纯虚函数以及抽象类等知识点及其简单运用。

第8章，模板：概述了函数模板、类模板以及STL相关概念及其简单运用。

第9章，流与异常处理：概述了输入流、输出流及异常处理相关概念及其简单运用。

综合实践篇主要根据题目的应用领域类型分为三大类：数值计算、数据管理和图形处理。

基础实践篇中的上机实验和综合实践篇的综合实践任务都根据难易程度提供必做题和选做题两种类型，以供不同学习基础的学生自行选择。

附录部分包括主流开发环境（平台）使用介绍、上机报告与综合实践报告样例，以方便学生参考、模仿和学习。

内容概要

“C++语言程序设计”或“面向对象程序设计”是一门实践性要求非常强的课程。作为该课程的全面而系统的实践教材，《C++与面向对象程序设计实践教程》由浅入深、循序渐进地介绍了C++语言的主要概念、知识点及其简单应用和综合应用。

全书内容主要包括基础实践（第1章至第9章）和综合实践（第10章至第12章）两大部分，其中基础实践篇每章包含知识要点、典型例题、典型习题与上机实验四个模块；综合实践篇包括数值计算、数据管理、图形处理三章，均为综合性实验，可作为课程设计、大型实验素材和综合实训的指导。

附录部分包括主流开发环境（平台）使用介绍、上机报告与综合实践报告样例，以方便广大读者参考、模仿和学习。

《C++与面向对象程序设计实践教程》既可作为相关课程的辅助教材，也可作为相关课程的课程设计、大型实验以及实训/培训的主教材，还可供广大C++爱好者参考。

书籍目录

前言 基础实践篇 第1章 绪论 1.1 知识要点 1.2 典型例题 1.3 典型习题 1.4 上机实验 第2章 数据类型、表达式与语句 2.1 知识要点 2.2 典型例题 2.3 典型习题 2.4 上机实验 第3章 函数与程序结构 3.1 知识要点 3.2 典型例题 3.3 典型习题 3.4 上机实验 第4章 数组、结构体与指针 4.1 知识要点 4.2 典型例题 4.3 典型习题 4.4 上机实验 第5章 类和对象 5.1 知识要点 5.2 典型例题 5.3 典型习题 5.4 上机实验 第6章 继承与派生 6.1 知识要点 6.2 典型例题 6.3 典型习题 6.4 上机实验 第7章 多态性 7.1 知识要点 7.2 典型例题 7.3 典型习题 7.4 上机实验 第8章 模板 8.1 知识要点 8.2 典型例题 8.3 典型习题 8.4 上机实验 第9章 流与异常处理 9.1 知识要点 9.2 典型例题 9.3 典型习题 9.4 上机实验 综合实践篇 第10章 数值计算 10.1 典型案例分析 10.2 综合实践任务 第11章 数据管理 11.1 典型案例分析 11.2 综合实践任务 第12章 图形处理 12.1 典型案例分析 12.2 综合实践任务 附录 附录A 编程环境使用简介 A.1 Borland C++ Builder集成编程环境 A.2 MinGW Developer Studio集成编程环境 A.3 Microsoft Visual C++集成编程环境 附录B 上机实验报告样例 附录C 综合实践报告样例 参考文献

章节摘录

1.构造函数 构造函数是一种特殊的成员函数，它主要用来为对象分配空间，给它的数据成员赋值（初始化），并执行对象的其他内部管理操作。

构造函数的名字与它所在的类名相同，它可以有参数，并允许重载。

当一个类中含有多个构造函数时，编译程序为了确定调用哪一个构造函数，需要把对象中使用的参数与构造函数的参数表进行比较，这个过程与普通重载函数相同。

说明：构造函数可以有参数，但没有返回值，且不能用void。

类定义中不一定要构造函数。

构造函数可以有缺省参数。

在重载没有参数和有缺省参数的构造函数时，注意避免产生二义性。

在类定义中，可以用其他类的对象作为成员，称为对象成员。

通常把含有对象成员类叫做组合类。

组合类中的构造函数是一个成员初始化列表。

2.析构函数 析构函数也是类的特殊成员函数，函数名为类名前加一个波浪号（~）。

它没有参数，也没有返回值，而且不能重载，即在一个类中只能有一个析构函数。

析构函数执行与构造函数相反的操作，通常用于释放分配给对象的存储空间，当程序退出类对象的作用域时，或者当对一个类指针使用运算符delete时，将自动执行析构函数。

和构造函数一样，如果不定义析构函数，编译程序将常调用一个缺省的析构函数。

对大多数类来说，缺省的析构函数就能满足要求，如果在一个对象完成其操作之前还需要做一些内部处理，则，应显式地定义析构函数。

5.1.4 静态成员 用类定义多个对象时，每个对象中都含有该类成员的副本，当一个类的所有对象都需要共享某些数据时，通常把这些共享数据定义为类的一部分，并加上关键字static作为静态成员。

用static声明的成员叫做静态成员，它具有与非静态成员不同的特性。

非静态成员：类的每个对象中都有各自不同的副本，通过该类的对象进行存取。

静态成员：整个类所有对象只存在一个副本，即使没有定义对象，也可以访问静态成员。

说明：静态成员实际上是在类中定义的一种全局变量（或函数），因此使用静态成员可以减少对全局变量的需要，而静态数据成员的主要用途是定义类的各个对象所公用的数据。

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>