

## <<C++程序设计实践教学>>

### 图书基本信息

书名：<<C++程序设计实践教学>>

13位ISBN编号：9787560923666

10位ISBN编号：7560923666

出版时间：2001-2

出版时间：华中科技大学出版社

作者：马光志

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>

## <<C++程序设计实践教程>>

### 前言

C++是由Bjarne Stroustrup于1986年在AT&T贝尔实验室开发的。

开发这一语言的目的在于通过数据封装减小程序变量的副作用，从而降低程序的复杂性并提高程序的可靠性。C++是C语言的直接扩展，C++的多继承机制能更好地描述对象的属性和行为，因此，C++特别有助于开发大型软件系统。

到目前为止，C++的语法和语义还在发展变化，已基本具备面向对象语言的全部特点。

许多教材往往用大量篇幅介绍C或C++的开发环境，而没有全面深刻地介绍类和面向对象的程序设计。

许多人在学完C++后，觉得C和C++没有多大区别，体会不到面向对象的程序设计的优点，这与教材忽视概念的应用背景介绍有很大关系。

本书全面深刻地介绍C++的类，不重复介绍C语言的有关概念，所有章节均以类为中心，由浅入深地逐步展开，尽量避免前后参照和相互关联，力图使初学者理解而不是死记概念。

在详细介绍有关概念的基础上，本书通过精心设计的例题，具体说明有关概念的使用方法。

绝大部分例题能在Borland C++ Ver 3.1上运行，少数例题必须在5.0甚至更高版本上才能运行。

## <<C++程序设计实践教学>>

### 内容概要

类、友元、封装、继承、重载、多态、引用、内联、友元、模板、异常、断言、虚函数、抽象类、静态成员、成员指针、名字空间、流及类库等等。

为了使本书介绍的对象建模技术更具实战性，本书按照面向对象的系统分析与设计步骤，完整地介绍了一个对象建模实例，并用C++语言进行了面向对象的程序设计。

为方便学生自学，本教材配有《学习指导与习题详解》光盘一片，其中对教材中的所有习题给出了详尽地解答。

并提供了若干套模拟题以帮助学生进行学习自测。

还附上了教材中例题、习题解答中的所有源代码，便于学生上机调试。

本书内容新颖、通俗易懂、注重理论与实践相结合，既可作为高等院校计算机及其相关专业的教材，又可作为c++初学者和高级程序设计人员的参考书。

## &lt;&lt;C++程序设计实践教程&gt;&gt;

## 书籍目录

第1章 引论1.1 程序设计语言1.2 程序编译技术1.3 面向对象的语言及程序设计1.4 面向对象的基本概念1.5 C++语言的特点1.6 C++的程序结构练习题第2章 C++的变量、类型及函数2.1 声明及定义2.2 类型定义2.3 引用类型2.4.函数参数2.5 函数内联练习题第3章 C++的类3.1 类的声明及定义3.2 访问权限3.3 内联及位段3.4 new和delete3.5 隐含参数this3.6 对象初始化3.7 类的存储空间练习题第4章 作用域及成员指针4.1 作用域4.2 名字空间4.3 成员指针4.4 const、volatile和mutable4.5 引用对象练习题第5章 静态成员与友元5.1 静态数据成员5.2 静态函数成员5.3 静态成员指针5.4 成员友元5.5 普通友元练习题第6章 单继承类6.1 单继承类6.2 派生控制6.3 成员访问6.4 构造与析构6.5 父类和子类6.6 派生类的存储空间练习题第7章 虚函数7.1 虚函数7.2 虚析构函数7.3 抽象类7.4 友元、绑定7.5 类的存储空间练习题第8章 多继承类8.1 多继承类8.2 虚基类8.3 派生类成员8.4 构造与析构8.5 类的存储空间练习题第9章 运算符重载9.1 概述9.2 运算符函数参数9.3 赋值与调用9.4 强制类型转换9.5 重载new和delete9.6 表运算实例练习题第10章 模板10.1 函数模板10.2 模板函数10.3 类模板10.4 模板类及覆盖10.5 内存回收实例练习题第11章 异常处理与断言11.1 异常处理11.2 catch顺序11.3 异常接口11.4 异常类型11.5 异常对象的析构11.6 断言练习题第12章 C++流及类库12.1 流类概述12.2 输出流12.3 输入流12.4.文件流12.5 串流处理第13章 对象分析与设计13.1 概述13.2 对象模型13.3 对象分析技术13.4 对象设计与实现13.5 一个实例13.6 对象实现附录A C++运算符表附录B ASCII字符码表参考文献

## &lt;&lt;C++程序设计实践教程&gt;&gt;

## 章节摘录

1.2 程序编译技术程序设计语言的翻译方式通常有两种，即解释方式和编译方式。

在解释方式下，由解释器读入高级语言程序，并逐条解释和执行程序的每条语句；在编译方式下，由编译程序读入高级语言程序，并编译成某种等价的目标语言程序，然后由操作系统装入和执行目标语言程序。

就程序最终的执行速度而言，用编译方式实现要比用解释方式实现快。

程序设计语言可以用多种方式实现。

例如，BASIC语言既可用解释方式实现，也可用编译方式实现，C和C++通常只以编译方式实现，以下重点介绍编译技术。

1.2.1 编译技术高级语言程序通常要经过预处理、词法分析、语法分析、代码生成和模块连接等环节，才能被编译成可被计算机执行的目标语言程序。

每个编译环节的输出通常作为下一编译环节的输入，该输出通常以文件的形式存放在磁盘上。

编译结束时，通常只保留目标文件和目标语言程序，中间文件在编译结束后会被自动删除。

对于C和C++编译程序而言，预处理过程通常要完成由#define定义的宏替换、由#include定义的文件包含以及由#if。

等定义的条件编译，最终生成不包含#define、#include及其他编译命令的文本文件。

在预处理过程中，注解、多余的空格或空行均会被删除。

词法分析的任务是分析识别出各种有意义的词法记号，它以预处理程序输出的文本文件作为输入，词法记号是高级语言程序中所使用的各种单词，包括保留字、标识符、操作符、分隔符以及各种常量。

识别的单词连同其他相关信息一起存放在符号表中，供语法分析程序检查语言的语法结构时使用。

语法分析程序根据程序设计语言的形式规范检查程序语法的正确性。

程序设计语言的形式规范严格地描述了变量声明、函数及各种语句的语法结构。

语法分析程序分析源程序的程序语义，其输出将作为代码生成程序的输入。

代码生成程序根据已经识别的程序语义生成和源程序等价的某种形式的中间代码，这种代码可以是一种需要再次解释的代码，也可以是一种带有连接信息的特定计算机的机器指令代码。

## <<C++程序设计实践教学>>

### 编辑推荐

《C++程序设计实践教学》内容新颖、通俗易懂、注重理论与实践相结合，既可作为高等院校计算机及其相关专业的教材，又可作为C++初学者和高级程序设计人员的参考书。

<<C++程序设计实践教学>>

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>