

<<C++高级程序设计>>

图书基本信息

书名：<<C++高级程序设计>>

13位ISBN编号：9787302292845

10位ISBN编号：7302292841

出版时间：2012-9

出版时间：清华大学出版社

作者：田秀霞 编

页数：446

字数：708000

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++高级程序设计>>

内容概要

C++程序设计语言同时支持面向过程和面向对象编程,《21世纪高等学校规划教材·计算机科学与技术: C++高级程序设计》以C++标准为蓝本,以实际案例为支撑来阐述学生学习过程中晦涩难懂的概念,并将多年来教学过程中学生遇到的难以掌握的知识点以重点突出或错误解释的方式列出。整本书的语言通俗易懂,每个章节的开始都以一个短小有趣的引发好奇心的说明引导学生进入本章节内容的学习。

整个内容按照循序渐进的方式展开,引领学生从C++的面向过程编程逐步过渡到C++的面向对象编程,实现学习点上的由易到难的顺利过渡。

本书内容包括:绪论、数据类型与基本输入输出、控制语句、函数与预处理、数组、指针与引用、自定义数据类型、类、静态成员与友元、继承与派生、多态性、输入输出流、模板和异常处理。

《21世纪高等学校规划教材·计算机科学与技术: C++高级程序设计》实例丰富,有配套的《C++高级程序设计实验与习题指导》并为广大学习者或教师提供了教程课件和实例源代码供下载。本教程可以分两学期来讲,第一学期建议48学时,其中实验16学时,第二学期48学时,其中实验16学时。

《21世纪高等学校规划教材·计算机科学与技术: C++高级程序设计》适用于高校计算机专业程序设计课程教学,也适用于自学成才的读者,在本书的指导下,读者可以逐步由不入门到初级程序员到高级程序员,并从中体会到学习编程的无限乐趣,为成为一名真正的程序员打下坚实的基础。

<<C++高级程序设计>>

书籍目录

第1章 绪论

- 1.1 程序设计语言的演变
- 1.2 C++程序设计语言的发展
- 1.3 面向过程程序设计
- 1.4 面向对象程序设计
- 1.5 计算机信息存储与表示
 - 1.5.1 数据在计算机中的表示
 - 1.5.2 不同形式数据的表示方法
- 1.6 C++程序的编程流程
- 1.7 编译一个简单的程序
- 1.8 C++的词法和规则
 - 1.8.1 C++的字符集
 - 1.8.2 词与词法规则
- 1.9 C++编译环境
 - 1.9.1 VC++6.0的集成开发环境
 - 1.9.2 第一个标准C++程序

本章小结

习题1

第2章 数据类型与基本输入输出

- 2.1 数据类型基本概念
 - 2.1.1 C++数据类型种类
 - 2.1.2 整型和实型
 - 2.1.3 数值的范围
 - 2.1.4 有符号数和无符号数
 - 2.1.5 有符号数和无符号数的范围区别
 - 2.1.6 本章用到的关键字
- 2.2 C++基本数据类型
 - 2.2.1 整型、实型和字符型
 - 2.2.2 布尔型和无类型
 - 2.2.3 为数据类型起别名
 - 2.2.4 sizeof的使用
- 2.3 常量
 - 2.3.1 整型常量
 - 2.3.2 字符常量和字符串常量
 - 2.3.3 逻辑常量
 - 2.3.4 枚举常量
 - 2.3.5 实型常量
 - 2.3.6 地址常量
- 2.4 符号常量 ,
 - 2.4.1 为什么需要符号常量
 - 2.4.2 const与#define的比较
- 2.5 变量
 - 2.5.1 变量概述
 - 2.5.2 变量声明
 - 2.5.3 赋值语句

<<C++高级程序设计>>

- 2.5.4 变量的初始化
- 2.6 输入和输出
 - 2.6.1 cout输出流
 - 2.6.2 cin输入流
- 2.7 运算符和表达式
 - 2.7.1 C++语言的运算符种类
 - 2.7.2 算术运算符
 - 2.7.3 赋值运算符
 - 2.7.4 关系运算符
 - 2.7.5 逻辑运算符
 - 2.7.6 位运算符
 - 2.7.7 条件运算符
 - 2.7.8 逗号运算符
 - 2.7.9 运算符优先级及结合性
- 本章小结
- 习题2
- 第3章 控制语句
 - 3.1 顺序控制语句
 - 3.1.1 表达式语句
 - 3.1.2 函数调用语句
 - 3.1.3 复合语句
 - 3.1.4 空语句
 - 3.1.5 赋值语句
 - 3.2 选择控制语句
-
- 第4章 函数与预处理
- 第5章 数组
- 第6章 指针与引用
- 第7章 自定义数据类型
- 第8章 类
- 第9章 静态成员与友元
- 第10章 继承与派生
- 第11章 多态性
- 第12章 输入输出流
- 第13章 模板
- 第14章 异常处理
- 参考文献

章节摘录

版权页：插图：对自定义函数的调用与对预定义函数的调用遵循一样的语法规则，那就是要进行正确的函数调用，首先需要给出相应的函数名如sum，然后给出用圆括号括起来的两个实参，如本例中的两个实参分别是整型的s和num，它们中存放的实际值的大小取决于程序运行时的键盘输入，最后函数调用结束后会返回一个整型的自然数之和，即函数值。

该函数值可以像任何整型表达式一样参加整型数据可以参加的运算。

如下都是正确的调用。

```
int m=sum ( 5 , 6 ) ; int k=200+sum ( 1 , 4 ) ; int d=2*m+sum ( S , num ) ; // s , num同上述程序中的含义
```

程序4_2.cpp中包括在注释函数定义中的粗体部分为函数定义。

函数定义是描述该函数如何实现指定的功能，返回指定类型值的部分。

它主要包括两个部分：函数头和函数主体（功能段）。

从sum函数的定义可以看到，函数定义的语法和main函数一样，唯一不同的是自定义函数不是由系统自动调用的，而main函数则是执行C++程序的唯一入口。

函数头和函数声明相同，只是函数头后面没有分号。

4.3.2 函数返回值 函数声明不能确定函数的返回值是什么，能够确定函数返回值的是通过在函数主体中添加return语句实现的。

函数的返回值在执行return语句时确定。

也就是说，如果需要从被调用函数带回一个函数值（以便在主调函数中使用），被调函数中必须包含return语句，如程序4_2.cpp中的return语句：return total；由关键字return和一个表达式构成，total的值作为函数调用的值返回。

表达式的类型应该和函数返回类型一致，如果不一致，那么以返回类型为标准进行转换。

例如，如果一个函数的返回类型为整型，但是函数中执行如下return语句：return 36.78；那么最后作为函数调用返回值的是36，即函数值为36。

为了避免混淆，也可以为return语句的表达式添加圆括号，这样使得程序更加容易阅读，如：return (total)；return语句后面一般不再写其他语句，即使写，程序也不再执行，因为程序一旦执行return语句就会终止当前的函数调用，即从当前的函数返回上一级主调用函数。

<<C++高级程序设计>>

编辑推荐

<<C++高级程序设计>>

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>