

<<C++程序设计实验教程>>

图书基本信息

书名：<<C++程序设计实验教程>>

13位ISBN编号：9787302249405

10位ISBN编号：7302249407

出版时间：2011-3

出版时间：清华大学出版社

作者：魏英

页数：212

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++程序设计实验教程>>

内容概要

本书是姜学锋主编的《C++程序设计》的配套实验教程。

《C++程序设计实验教程》分为4部分，其中详细介绍了开发工具的使用方法和程序调试技术。实验内容按课程教材和教学大纲要求设计，分验证型实验和设计型实验，突出综合性实验，并结合算法、数据结构知识设计了一些有难度的实验题目。

《C++程序设计实验教程》还包括课程设计专题实验内容，其目的是使读者能够训练应用程序开发，获取设计C++程序项目的初步知识和工程经验，掌握高级编程技术，为后续专业学习和职业发展打下坚实的实践基础。

本书的作者长期从事计算机基础教学和软件开发科研工作，具有丰富的教学经验和软件开发经验。全书贯彻“精讲多练、提升技能、开拓设计”的教学理念，精心策划、准确定位、结构清晰、语言通俗易懂，内容由浅入深、实验循序渐进。验证型实验体现“学”，设计型实验体现“用”，课程设计体现“提升和开拓”，核心目标是技能和计算思维能力训练。

本书适合作为高等学校各专业程序设计课程的实验教材，可以独立设课，也可作为自学者学习参考用书。

<<C++程序设计实验教程>>

书籍目录

第1章 开发环境及上机操作

1.1 C++语言开发环境简介

1.1.1 编译器和连接器

1.1.2 C++语言编译器

1.1.3 集成开发环境(IDE)

1.1.4 快速应用开发(RAD)工具

1.2 Visualc++6.0开发环境及上机操作

1.2.1 Visualc++6.0简介

1.2.2 启动和退出Visualc++6.0

1.2.3 配置Visualc++6.0

1.2.4 Visualc++6.0开发环境和基本菜单

1.2.5 建立和编辑源程序

1.2.6 编译、连接和运行

1.3 Code Blocks+GCC+GGB开发环境及上机操作

1.3.1 Code Blocks简介

1.3.2 下载Code Blocks

1.3.3 安装Code Blocks

1.3.4 配置Code Blocks

1.3.5 Code Blocks开发环境和基本操作

第2章 程序调试技术

2.1 概述

2.2 程序调试的方法

2.2.1 单步法

2.2.2 断点法

2.3 常见编译系统调试功能

2.3.1 单步

2.3.2 断点

2.3.3 观察

2.3.4 控制

2.4 Visualc++6.0调试方法

2.4.1 语法排错

2.4.2 调试设置

2.4.3 单步调试

2.4.4 快步调试

2.4.5 断点调试

2.4.6 动态调试

2.4.7 数据观察

2.4.8 远程调试

2.4.9 宏调试

2.5 Code Blocks调试方法

2.4.1 语法排错

2.4.2 调试设置

2.4.3 调试举例

第3章 基础实验内容

3.1 实验指导

<<C++程序设计实验教程>>

3.2 实验内容及安排

3.2.1 实验1 C++语言程序初步及输入输出

3.2.2 实验2 选择结构

3.2.3 实验3 循环结构

3.2.4 实验4 函数与预处理前令

3.2.5 实验5 数组

3.2.6 实验6 指针、引用与函数

3.2.7 实验7 结构体与函数

3.2.8 实验8 类与对象

3.2.9 实验9 继承与派生

3.2.10 实验10 运算符重载与标准库

第4章 课程设计

4.1 API接口方法

4.1.1 查看与设置开发环境的路径参数

4.1.2 库的包含和链接

4.1.3 开发环境配置举例

4.2 实验内容及安排

4.2.1 实验1 常用算法

4.2.2 实验2 数值计算

4.2.3 实验3 界面编程

4.2.4 实验4 图形输出、事件处理与对话框

4.2.5 实验5 图形编程

4.2.6 实验6 多媒体编程

4.2.7 实验7 网络编程

4.2.8 实验8 数据库编程

附录A 常见编译错误信息

A.1 Visual C++6.0错误信息概述

A.2 Visual C++6.0编译错误信息列表

参考文献

<<C++程序设计实验教程>>

章节摘录

1.1 C++语言开发环境简介 1.1.1 编译器和连接器 计算机是按照计算机指令自动工作的，计算机的工作过程就是指令的执行过程。

让计算机执行什么样的工作，得到什么样的结果的过程本质上就是编写什么样指令的过程。

在计算机发展早期，编写计算机指令是一件非常复杂的事情，后来人们逐步设计出了各种高级语言，大大简化了指令（程序，指令的集合）设计的难度，并且提高了程序生产效率。

编译器是将一种计算机语言翻译为另一种计算机语言的程序。

编译器将源语言（Source Language）编写的程序（简称源程序）作为输入，编译成用目标语言（Target Language）编写的等价程序。

源程序一般为高级语言（High-level Language），例如C、C++等。

而目标语言则是汇编语言或目标机器的目标代码（Object code，有时也叫做机器代码Machine Code）。

编译器可以生成用在与编译器本身所在的计算机和操作系统（或平台）相同的环境下运行的目标代码，这种编译器叫“本地编译器”；编译器也可以生成用来在其他平台上运行的目标代码，这种编译器叫做交叉编译器，交叉编译器在生成新的硬件平台时非常有用。

编译器有两种方式可以执行高级语言程序：一是通过解释程序；二是通过编译、连接生成执行代码。

第一种方式，解释程序能够直接执行高级语言源程序。

这种方式非常方便，但是效率不高，而且没有安装解释程序的计算机不能执行，例如Java语言等就是采用解释方式。

第二种方式，使用编译器，将高级语言源程序编译、连接成为执行代码，也就是二进制的机器指令，从而允许用户直接执行程序，C语言、C++语言等就是这样的方式。

尽管经过编译过程后，高级语言源程序转换成二进制的执行代码了；但在大多数的操作系统上，执行这些执行代码是按“进程”方式管理的，因此，这些二进制的执行代码还需要增加与进程和操作系统相关的执行代码，这个过程就称为“连接”。

完成这种连接工作的程序称为“连接器”。

⋮

<<C++程序设计实验教程>>

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>