

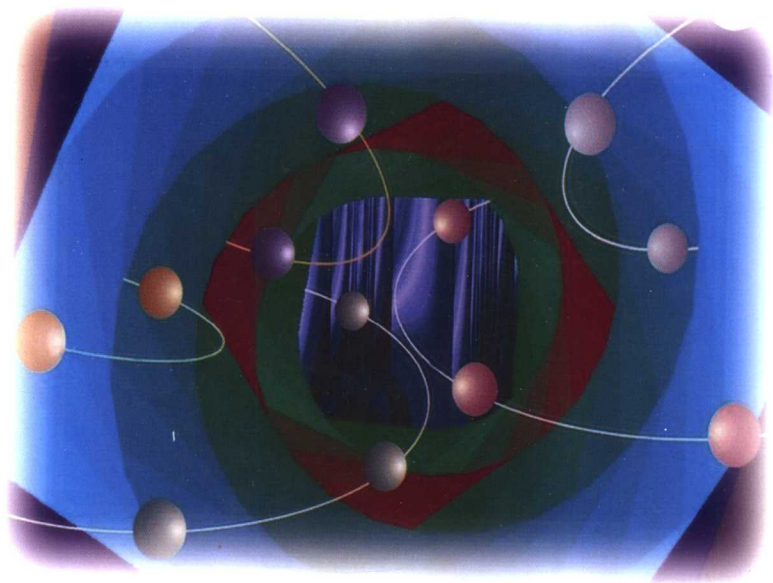
清华大学计算机基础教育课程系列教材



清华大学计算机基础教育课程系列教材

C++ 语言程序设计 (第3版) 学生用书

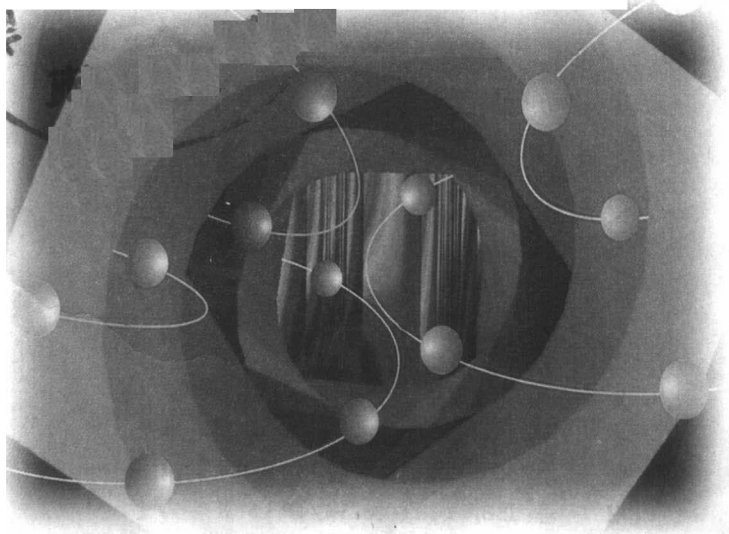
郑莉 傅仕星 张瑞丰 编著



清华大学出版社

C++语言程序设计 (第3版) 学生用书

郑莉 傅仕星 张瑞丰 编著



清华大学出版社
北京

内 容 简 介

本书是在原《C++ 语言程序设计习题与实验指导》一书的基础上编写的,作为与《C++ 语言程序设计(第3版)》配套的学生用书。

本书首先给出一个简要的“学习指南”,其余章节与主教材《C++ 语言程序设计(第3版)》相对应,每章内容分为3部分:第一部分是“要点导读”,主要是为自学读者指明学习重点,建议学习方法。第二部分是“实验指导”,每章都有一个精心设计的实验,与《C++ 语言程序设计(第3版)》相应章内容配合,使读者在实践中达到对主教材内容的深入理解和熟练掌握。每一个实验都包括“实验目的”、“实验任务”和“实验步骤”。第三部分是习题解答,给出了《C++ 语言程序设计(第3版)》各章习题的参考答案,共198道习题。

本书附录中还给出3个非常实用的附录——“Visual C++ 开发环境简介”、“ASCII 码表”和“STL 中通用算法函数原型及说明”。

本书适合作为高等院校程序设计课程的教辅书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

C++ 语言程序设计(第3版)学生用书/郑莉,傅仕星,张瑞丰编著 —北京:清华大学出版社,2004.5
(清华大学计算机基础教育课程系列教材)

ISBN 7-302-08575-7

I. C… II. ①郑… ②傅… ③张… III. C语言—程序设计—高等学校—教学参考资料 IV. TP312
中国版本图书馆CIP数据核字(2004)第039433号

出版者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:王敏稚

印 装 者:三河市印务有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:14 字数:320千字

版 次:2004年6月第1版 2004年6月第1次印刷

书 号:ISBN 7-302-08575-7/TP·6151

印 数:1~6000

定 价:18.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

PJS 115/02

3. 适应面广: 该系列教材照顾了理、工、文等各种类型专业的教材要求。

4. 立体配套: 为适应教学模式、教学方法和手段的改革, 该系列教材中多数都配有习题集和实验指导、多媒体电子教案, 有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践, 凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善, 不断更新, 为我国高校计算机基础教育做出新的贡献。



注: 周远清, 曾任教育部副部长, 原清华大学副校长、计算机专业教授。

前言

计算机程序设计是一门实践性很强的课程,因此仅仅通过阅读教科书或听课是不可能完全掌握的,学习程序设计的最重要环节就是实践。对于自学读者来说,更多一重困难,就是在学习和实践过程中缺乏指导。

凡是学习程序设计的人,往往有这样的感觉:看书或听课时,对老师讲的和书上写的内容基本上能够理解,但是当需要自己编程时却又无从下手。相信每一个讲授程序设计课程的教师都有过这样的经历:有些问题,尽管在课堂上再三强调,反复举例,学生还是不能够完全理解,上机时更是错误百出。应该说,这是学习过程中的必然现象。

要想能够把书本上的知识变为自己所具有的能力,需要的是实践、实践、再实践。在实践环节中,起主导作用的是学习者自己,旁人是无法代劳的,也不能期望有什么一蹴而就的捷径。但是由于学生在实践过程中不能随时随地得到指导,因此花费时间较多,总感觉程序设计课作业负担太重,有的学生甚至因为花四五个小时调不通一个简单的程序而失去学习兴趣。像 C++ 这样面向对象的程序设计语言学习起来尤其如此。

本书是在原《C++ 语言程序设计习题与实验指导》一书的基础上编写的,笔者编写本书作为与《C++ 语言程序设计(第 3 版)》配套的学生用书,目的就在于为读者的学习提供一些指导,为提高读者的编程能力助一臂之力,使读者在实践的过程中少一些曲折和彷徨,多一些成功的乐趣。本书出版之前已经在清华大学“计算机软件技术基础”和“C++ 语言程序设计”课中试用,取得了良好的教学效果。

本书首先给出一个简要的“学习指南”,其余章节与主教材《C++ 语言程序设计(第 3 版)》相对应,每章内容分为 3 部分:第一部分是“要点导读”,主要是为自学读者指明学习重点,建议学习方法;第二部分是“实验指导”,每章都有一个精心设计的实验(每个实验都给出参考学时),与《C++ 语言程序设计(第 3 版)》相应章内容配合,使读者在实践中达到对主教材内容的深入理解和熟练掌握。每一个实验都包括“实验目的”、“实验任务”、“实验步骤”和“答案与提示”,实验的完整参考程序可从网上下载(稍后详述);第三部分是习题解答,给出了《C++ 语言程序设计(第 3 版)》各章习题的参考答案,共 198 道习题。每个题目可能有多种解法,这里仅给出一种参考解法。大部分题目是编程题,在解答中给出了主要程序段的源程序清单,有时不是完整的程序,如果需要运行这些语句,只需将它们插入调试程序即可。

这些习题解答和实验内容,不仅可以指导读者上机练习,也可以由教师选作例题在课堂上演示,使教学内容更加丰富。如果读者没有足够的时间一一做全部习题和实验,可以将剩下的题解作为例题阅读也不失为一种好的选择。

本书中的全部程序都在 Windows 环境下 Visual C++ 系统中测试通过。

为了给自学读者以更多的帮助,作者将自己在清华大学的教学资源(包括例题源代码、电子教案等)向中国国内读者免费开放。郑莉在清华大学授课的教学资源位于“清华大学网络学堂”,网址是:<http://learn.tsinghua.edu.cn/>。读者请在“本学期课程”栏目中查找郑莉的“C++ 语言程序设计”(如果首页未列出,请单击“更多”),找到后单击课程名,然后以用户名“GUEST”、密码“guest”登录,即可阅读、下载教学资源。在作者不授课的学期,读者可以查找以往课程,下载学习资源。

参加本书编写的还有董渊、张文举、孟威、田荣牌、孟鸿利、杨兴朋、王绚,在此表示感谢。

感谢读者选择使用本书,欢迎您对本书内容提出意见和建议。作者的电子邮件地址:zhengli@mail.tsinghua.edu.cn,来信标题请包含“C++ book”。

作者

2004年4月于清华大学

目 录

《C++ 语言程序设计(第3版)》学习指南	1
第1章 绪论	3
要点导读	3
实验一 VC++ 6.0 开发环境应用入门(2学时)	4
习题解答	6
第2章 C++ 简单程序设计	9
要点导读	9
实验二 C++ 简单程序设计(4学时)	10
习题解答	14
第3章 函数	34
要点导读	34
实验三 函数的应用(2学时)	34
习题解答	36
第4章 类与对象	44
要点导读	44
实验四 类与对象(4学时)	45
习题解答	47
第5章 C++ 程序的结构	57
要点导读	57
实验五 C++ 程序的结构(2学时)	57
习题解答	59
第6章 数组、指针与字符串	67
要点导读	67
实验六 数组、指针与字符串(4学时)	68
习题解答	69
第7章 继承与派生	87
要点导读	87
实验七 继承与派生(4学时)	88
习题解答	89

第8章 多态性	100
要点导读.....	100
实验八 多态性(2学时).....	100
习题解答.....	101
第9章 群体类和群体数据的组织	114
要点导读.....	114
实验九 群体类和群体数据(4学时).....	115
习题解答.....	116
第10章 泛型程序设计与 C++ 标准模板库	134
要点导读	134
实验十 标准模板库的应用(2学时)	134
习题解答	135
第11章 流类库与输入输出	160
要点导读	160
实验十一 流类库与输入输出(2学时)	160
习题解答	162
第12章 异常处理	169
要点导读	169
实验十二 异常处理(2学时)	169
习题解答	170
第13章 MFC 库与 Windows 程序开发概述	175
要点导读	175
实验十三 创建 Windows 应用程序(2学时)	175
习题解答	179
附录 A Visual C++ 开发环境简介	180
附录 B ASCII 码表	193
附录 C STL 中通用算法函数原型及说明	195

《C++ 语言程序设计(第3版)》学习指南

《C++ 语言程序设计(第3版)》是针对初学程序设计语言的读者编写的入门教材,预期的读者主要有这样几种:初学程序设计的自学读者、以此为C++课程教材或参考书的在校学生、以此为参考资料的C++程序员、C++课程教师。针对教师,另外编写了配套的教师用书。对有经验的程序员而言,自然也不必遵循什么学习指导。因此在这里主要是对前两种读者提出一些学习建议,首先给出学习本套教材的总体建议,在本书的后续各章中还会有详细的导读。

一、主教材《C++ 语言程序设计(第3版)》的学习方法

自学读者在阅读教材时,应该边阅读、边实践。有条件的应该坐在计算机前,边阅读边亲自编写每一个例题程序,如果对于某些概念、语法存有疑问,应该立即编写程序予以验证。在完全理解了主教材内容以后,再开始做实验和习题。

对于在校学习C++课程的学生,应按照教师讲课的进度,提前预习教材。所谓预习,并不是要完全看懂,如果都看懂了,就不必听课了。预习的目的是大致浏览一下新的内容,了解哪些是难点、重点,将疑问记下来,听课时就比较主动。上课之后要及时复习,然后再写作业。复习时要边看书边看笔记,这时一定要认真阅读书上的内容,要完全搞懂。教师可能不会在课堂上讲解书中的每一个例题,对于教师课上讲的例题和书上的例题,课后复习时都要阅读并上机实践,达到完全理解,要能够自己独立编写例题程序,还要尝试用不同的方法解决问题。做到这些以后,再开始写作业。

有的读者对于以上建议会不以为然,但这是大多数初学者达到事半功倍的途径。我在清华大学教C++课就一直是这样要求学生的,也曾经有学生在我讲完了上述要求之后,立刻站起来反驳:你的要求太麻烦了,又要预习又要复习,我们有那么多课程要学,没有时间。但是后来的无数事实证明,想省时间的多半欲速则不达,甚至出现“夹生饭煮不熟”的情况,而一步一个脚印往前走的,走得最从容,总体上花的时间也最短。当然,这只是针对大多数读者的一般的建议,每个人还要根据自己的情况选择适合自己的方法。

二、学生用书的使用方法

每学习一章主教材内容,都应该及时通过实验和习题巩固知识,提高实践能力。学生用书中的实验是针对主教材每一章的重点内容设计的最基本的实践任务,有详细的实验指导,很容易入手,应该首先完成。完成实验之后,可以根据自己的时间和教师的要求,选择部分或全部习题来做。

本书给出了全部习题的答案,这是为了方便没有教师指导的自学读者。但是不少读者在没有深入思考之前就急于看答案,这是有害无益的,这样做不仅不能真正提高自己的编程能力,还会扼杀自己的创造性思维能力。有些学生在邻近考试的时候,就来诉苦:书上的例题和习题解答都能看懂,可是自己写程序就不会下手。仔细一问,这些都是平时急

于看习题答案的。自学读者纷纷来信表示欢迎习题解答,而大多数教师(包括我自己)都不希望学生看到习题解答。这个矛盾困扰了我很久,始终没有找到两全的解决方案。所以只好在这里给予建议。

当然,有些章的习题较多,如果你没有时间全部做完,也可以将一部分习题解答作为例题来学习。

三、关于编程能力的困惑

学完本套教材之后,许多读者都会遇到这样的困惑: C++ 语言学会了,但是面对实际问题还是不知道该如何写程序,这是为什么? 每个学期末都有很多学生向我问这种问题。于是每学期最后一节课,我都要举这样的例子: 我们都是以中文为母语的,对中文的掌握可谓精通了。但是你是否有能力用中文写出某个项目的实施方案和某个企业的发展规划? 恐怕大多数学生做不到,因为只掌握语言是不够的,还需要有相关的专业知识和工作经验。

编写程序的道理也是相同的,就是要用程序语言将需要解决的问题和解决问题的方案描述清楚。仅仅掌握 C++ 语言是不够的,还需要学习解决各类问题的专门方法。为此很多程序语言教材(包括本套教材)都声称: 不仅介绍语言本身而且介绍分析问题和解决问题的方法。但是实事求是地讲,这些都只是介绍分析问题和解决问题方法的皮毛。如果只读一本薄薄的书,就什么程序都会写了,学校里还要开设那么多基础课和专业课干什么? 比如高等数学、计算方法、数据结构、软件工程等。学会一门高级语言只是掌握了一种描述工具,要真正具备较强的分析问题和解决问题的能力,要学的东西还很多,除了认真学习、勤奋实践以外没有捷径可走。所以初学者要给自己定一个现实的目标,本套书主教材的第 9、10 章介绍了一些基本的对群体数据的管理方法和类库中相应的算法,第 13 章简单介绍了 Windows 图形用户界面程序的设计方法,就是为了使读者掌握一些基本的解决问题方法,能够运用 C++ 语言编写程序解决一些简单问题,并为读者今后继续学习相关课程打开一扇窗户。

第 1 章

绪论

要点导读

本章作为全书的开篇,旨在使读者初步了解面向对象的程序设计语言的由来,初步了解面向对象的程序设计思想的基本特点,概要地了解面向对象的软件开发方法,为后续章节的学习奠定基础。

为什么需要首先有一个初步和概要的了解呢?一方面,这是为了在以后的学习中具体接触到每一个新的概念、语法时都能够清楚地认识到它在面向对象的方法中、在 C++ 语言中的地位和作用是什么;另一方面,是希望读者在一开始就能够认识到,面向对象的思想与人类所习惯的思维方式是一致的,虽然 C++ 语言比面向过程的语言(如 C 语言)要复杂许多,但是 C++ 设计者的目的是为了把事情变得更简单,而不是故弄玄虚将事情搞得更复杂。事实上,正是由于 C++ 语法的复杂性,使得它的表现能力更强,程序员用 C++ 写程序的时候能够更容易、更灵活地实现各种功能。

读者在阅读 1.1~1.3 节时会感觉很多问题理解不透,这是正常的。因为需要学完本教材的全部内容,才能对 C++ 语言和面向对象的方法有一个全面的认识。而本章在一开始就给出了一个全面介绍,虽然尽量使用通俗的语言,但是肯定仍有一些问题是读者现在不能完全理解的。对此读者不必深究,对 1.1~1.3 节的内容阅读后有个大致的了解就行。

1.4 节介绍了信息的表示与存储,这是程序设计的基本知识,是必须掌握的基础。建议读者认真学习,完全掌握。不过有些读者可能会觉得这些知识在编程中并没有直接使用,不学这一节好像也不影响学习编程。但实际上,没有这些基础知识,会影响你对程序的理解。当然,如果觉得枯燥,也可以先略过这一节,待以后遇到疑问时,再来学习。因此有的教师在讲课时也略过这一节,留给学生自学,作者本人就是这样做的。

1.5 节简单介绍了程序的开发过程和一些术语,不必死记硬背,最好结合实验来体会。

本章的主要实验任务是学会使用一种 C++ 开发工具,本书的实验用的是 Visual C++ 6.0 开发环境。认真完成这一实验很重要,了解开发环境的基本功能,是完成以后各章实验的基础。

实验一 VC++ 6.0 开发环境应用入门(2学时)

一、实验目的

- (1) 了解 Visual C++ 6.0 的特点。
- (2) 熟悉 Visual C++ 6.0 的开发环境。
- (3) 学习用 Visual C++ 6.0 编写标准的 C++ 控制台程序。

二、实验任务

使用 Visual C++ 建立一个非图形化的标准 C++ 程序,编译、运行例 2-1,即以下程序:

```
#include <iostream >
using namespace std;
void main(void)
{
    cout << "Hello! \n";
    cout << "Welcome to C++! \n";
}
```

三、实验步骤

1. 启动 Visual C++ 6.0 开发环境

➤ 从“开始”菜单中选择“程序”| Microsoft Visual Studio 6.0 | Microsoft Visual C++ 6.0,显示 Visual C++ 6.0 开发环境主窗口。

2. 创建一个项目

➤ 单击 File 菜单中的 New 选项显示 New(新建)对话框(如图 1-1 所示)。

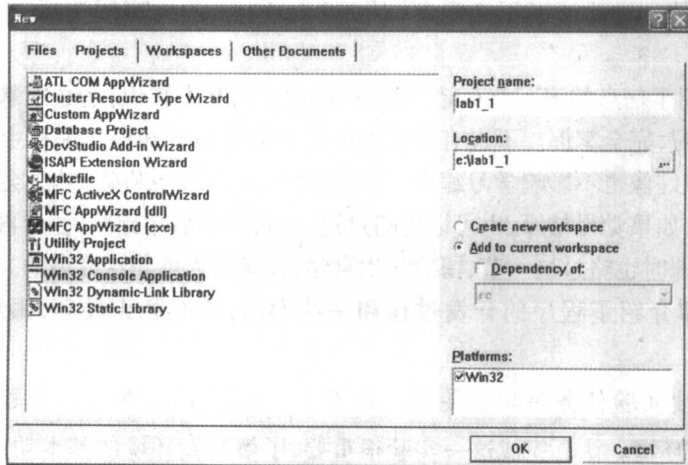


图 1-1 New 对话框

➤ 单击 Projects 标签,在 Projects(项目)选项卡中,选择 Win32 Console Application (Win32 控制台应用程序)。在 Location(位置)文本框中指定一个路径,在 Project Name(项目名称)文本框中为项目输入一个名字“lab1_1”,单击 OK(确定)按钮。

➤ 在弹出的 Win32 Console Application-Step 1 of 1 对话框中选择 An empty project 单选项,然后单击 Finish(完成)按钮(如图 1-2 所示)。

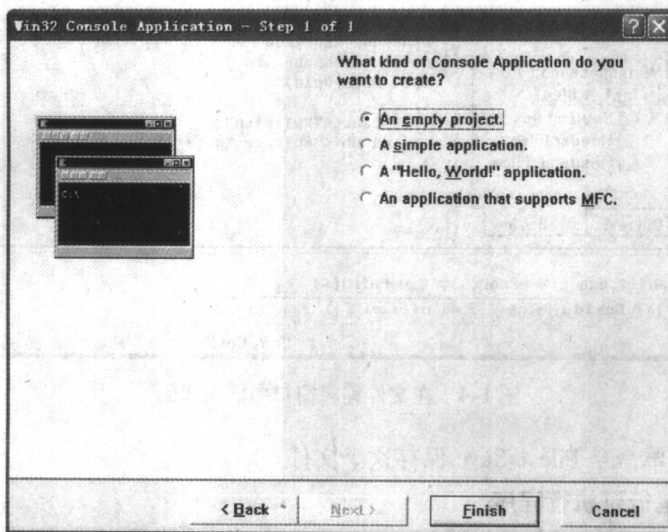


图 1-2 创建控制台应用程序第一步

➤ 最后在 New Project Information 对话框中单击 OK 按钮,完成项目的建立。

3. 建立 C++ 源程序文件

➤ 选择菜单命令 Project | Add to Project | New,弹出 New 对话框。如图 1-3 所示。

➤ 在 New 对话框的 Files 选项卡中选择 C++ Source File,并填入文件名称“lab1_1”,单击 OK 按钮,完成新建 C++ 源程序文件。

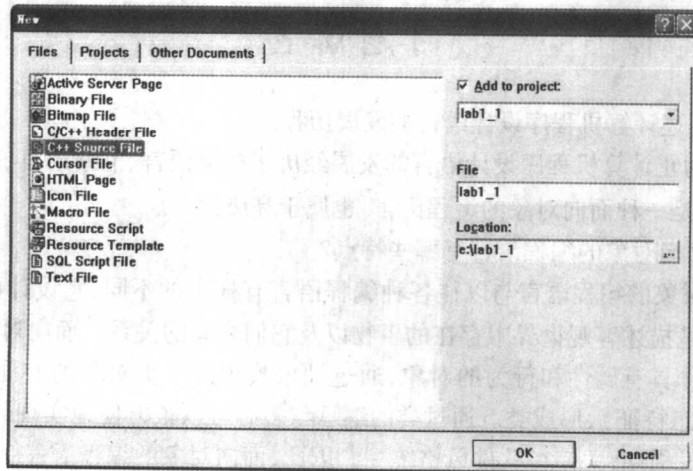


图 1-3 添加新的源程序文件

4. 编辑 C++ 源程序文件内容

- 在文件编辑窗口中输入代码(如图 1-4 所示)。

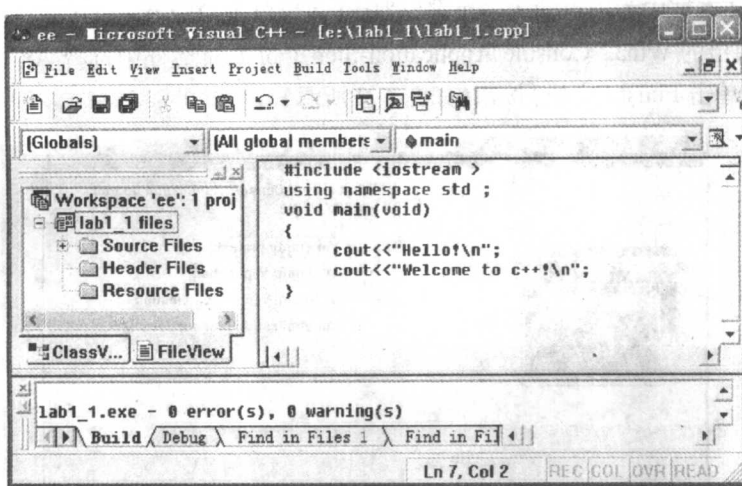


图 1-4 在文件编辑窗口中输入代码

- 选择菜单命令 File | Save 保存这个文件。

5. 建立并运行可执行程序

- 选择菜单命令 Build | Build lab1_1.exe 建立可执行程序。

如果你正确输入了源程序,此时便成功地生成了可执行程序 lab1_1.exe。如果程序有语法错误,则屏幕下方的状态窗口中会显示错误信息。根据这些错误信息对源程序进行修改后,重新选择菜单命令 Build | Build lab1_1.exe,建立可执行程序。

- 选择菜单命令 Build | Execute lab1_1.exe 运行程序,观察屏幕的显示内容。

6. 关闭工作空间

- 选择菜单命令 File | Close Workspace 关闭工作空间。

习题解答

【1-1】 简述计算机程序设计语言的发展历程。

解: 迄今为止计算机程序设计语言的发展经历了机器语言、汇编语言、高级语言等阶段。C++ 语言是一种面向对象的编程语言,也属于高级语言。

【1-2】 面向对象的编程语言有哪些特点?

解: 面向对象的编程语言与以往各种编程语言有根本的不同,它设计的出发点就是为了能更直接地描述客观世界中存在的事物以及它们之间的关系。面向对象的编程语言将客观事物看作具有属性和行为的对象,通过抽象找出同一类对象的共同属性(静态特征)和行为(动态特征),形成类。通过类的继承与多态可以很方便地实现代码重用,大大缩短了软件开发周期,并使得软件风格统一。因此,面向对象的编程语言使程序能够比较直接地反映问题域的本来面目,软件开发人员能够利用人类认识事物所采用的一般思维

方法来进行软件开发。C++语言是目前应用最广的面向对象的编程语言。

【1-3】 什么是结构化程序设计方法？这种方法有哪些优点和缺点？

解：结构化程序设计的思路是：自顶向下、逐步求精；其程序结构是按功能划分为若干个基本模块；各模块之间的关系尽可能简单，在功能上相对独立；每一模块内部均是由顺序、选择和循环3种基本结构组成；其模块化实现的具体方法是使用子程序。结构化程序设计由于采用了模块分解与功能抽象，自顶向下、分而治之的方法，从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务，便于开发和维护。

虽然结构化程序设计方法具有很多的优点，但它仍是一种面向过程的程序设计方法，它把数据和处理数据的过程分离为相互独立的实体。当数据结构改变时，所有相关的处理过程都要进行相应的修改，每一种相对于老问题的新方法都要带来额外的开销，程序的可重用性差。

由于图形用户界面的应用，程序运行由顺序运行演变为事件驱动，使得软件使用起来越来越方便，但开发起来却越来越困难，对这种软件的功能很难用过程来描述和实现，使用面向过程的方法来开发和维护都将非常困难。

【1-4】 什么是对象？什么是面向对象方法？这种方法有哪些特点？

解：从一般意义上讲，对象是现实世界中一个实际存在的事物，它可以是有形的，也可以是无形的。对象是构成世界的一个独立单位，它具有自己的静态特征和动态特征。面向对象方法中的对象是系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位，由一组属性和一组行为构成。

面向对象的方法将数据及对数据的操作方法放在一起，作为一个相互依存、不可分离的整体——对象。对同类型对象抽象出其共性，形成类。类中的大多数数据只能用本类的方法进行处理。类通过一个简单的接口与外界发生关系，对象与对象之间通过消息进行通信。这样，程序模块间的关系更为简单，程序模块的独立性、数据的安全性就有了良好的保障。通过实现继承与多态性，还可以大大提高程序的可重用性，使得软件的开发和维护都更为方便。

面向对象方法所强调的基本原则就是直接面对客观存在的事物进行软件开发，将人们在日常生活中习惯的思维方式和表达方式应用在软件开发中，使软件开发从过分专业化的方法、规则和技巧中回到客观世界，回到人们通常的思维。

【1-5】 什么叫做封装？

解：封装是面向对象方法的一个重要原则，就是把对象的属性和服务结合成一个独立的系统单位，并尽可能隐蔽对象的内部细节。

【1-6】 面向对象的软件工程包括哪些主要内容？

解：面向对象的软件工程是面向对象方法在软件工程领域的全面应用，它包括面向对象的分析(OOA)、面向对象的设计(OOD)、面向对象的编程(OOP)、面向对象的测试(OOT)和面向对象的软件维护(OOSM)等主要内容。

【1-7】 简述计算机内部的信息可分为几类？

解: 计算机内部的信息可以分成控制信息和数据信息两大类;控制信息可分为指令和控制字两类;数据信息可分为数值信息和非数值信息两类。

【1-8】 什么叫二进制? 使用二进制有何优点和缺点?

解: 二进制是基数为2, 每位的权是以2为底的幂的进制, 遵循逢二进一的原则, 基本符号为0和1。采用二进制码表示信息, 有如下几个优点: 易于物理实现; 二进制数运算简单; 机器可靠性高; 通用性强。其缺点是它表示数的容量较小, 二进制表示同一个数较其他进制需要更多的位数。

【1-9】 请将以下十进制数值转换为二进制和十六进制补码:

- (1) 2 (2) 9 (3) 93
(4) -32 (5) 65535 (6) -1

解:

- (1) $(2)_{10} = (10)_2 = (2)_{16}$
(2) $(9)_{10} = (1001)_2 = (9)_{16}$
(3) $(93)_{10} = (1011101)_2 = (5D)_{16}$
(4) $(-32)_{10} = (11100000)_2 = (E0)_{16}$
(5) $(65535)_{10} = (11111111\ 11111111)_2 = (FFFF)_{16}$
(6) $(-1)_{10} = (11111111\ 11111111)_2 = (FFFF)_{16}$

【1-10】 请将以下数值转换为十进制:

- (1) $(1010)_2$ (2) $(10001111)_2$ (3) $(01011111\ 11000011)_2$
(4) $(7F)_{16}$ (5) $(2D3E)_{16}$ (6) $(F10E)_{16}$

解:

- (1) $(1010)_2 = (10)_{10}$
(2) $(10001111)_2 = (143)_{10}$
(3) $(01011111\ 11000011)_2 = (24515)_{10}$
(4) $(7F)_{16} = (127)_{10}$
(5) $(2D3E)_{16} = (11582)_{10}$
(6) $(F10E)_{16} = (61710)_{10}$

【1-11】 简要比较原码、反码、补码等几种编码方法。

解: 原码: 将符号位数字化为0或1, 数的绝对值与符号一起编码, 即所谓“符号-绝对值表示”的编码。

正数的反码和补码与原码表示相同。

负数的反码与原码有如下关系:

- 符号位相同(仍用1表示), 其余各位取反(0变1, 1变0)。
- 补码由该数反码的最末位加1求得。