

普通高等教育“十二五”规划教材
高等院校计算机系列教材

成 稿 译 文

本书是根据教育部《计算机专业人才培养方案》和《计算机专业教学大纲》的要求，结合我国计算机专业教学改革的实际，参照国外著名教材编写而成的。

本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机工作的工程技术人员参考。本书共分10章，主要内容包括：C++语言概述、C++语言的基本语法、C++语言的数据类型、C++语言的运算符、C++语言的表达式、C++语言的语句、C++语言的函数、C++语言的数组、C++语言的指针、C++语言的字符串、C++语言的模板、C++语言的异常处理、C++语言的命名空间、C++语言的线程、C++语言的进程、C++语言的互斥、C++语言的信号量、C++语言的死锁、C++语言的同步、C++语言的异步、C++语言的并行、C++语言的分布式、C++语言的云计算、C++语言的大数据、C++语言的物联网、C++语言的智慧城市、C++语言的智能交通、C++语言的智能制造、C++语言的智能医疗、C++语言的智能教育、C++语言的智能家居、C++语言的智能农业、C++语言的智能工业、C++语言的智能服务、C++语言的智能生活、C++语言的智能社会、C++语言的智能未来。

C++ 程序设计教程

主 编 瞿绍军 刘 宏

副主编 (C++) 瞿绍军 刘宏

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

主 审 王 伟

华中科技大学出版社
中国·武汉

内 容 简 介

本书紧密结合目前高等学校计算机教学的发展趋势,将 ACM 国际大学生程序设计竞赛的相关内容引入教材,对学生养成良好的编程习惯和编程思维,提高分析和解决问题的能力大有帮助,这是本书的创新之处。

全书共分 12 章,各章节内容由浅入深、相互衔接、前后呼应、循序渐进。第 1~6 章介绍了 C++ 程序设计的基础、函数与程序结构、数组与字符串、指针、结构体与共用体、ACM 国际大学生程序设计竞赛相关知识和竞赛中的输入/输出等;第 7~11 章介绍了 C++ 面向对象的特性,包括类与对象及封装性、类的深入、运算符重载和模板、继承性、多态性;第 12 章介绍了 C++ 输入/输出流和 ACM 国际大学生程序设计竞赛中的文件输入/输出;附录 A 列出了 ASCII 码对照表;附录 B 列出了 C/C++ 与标准 C++ 头文件对照表,方便学习和参考;附录 C 介绍了 Linux、Unix 下编译和调试 C++ 程序的方法。

与本书配套的教材《C++ 程序设计教程习题答案和实验指导》提供了本课程的实验内容、习题参考答案和模拟试卷。所有习题和程序均按照 ACM 国际大学生程序设计竞赛要求进行设计,并进行了严格的测试,验证了程序的正确性。本课程设有专门的课程学习网站,具有丰富的学习参考资料,所有的习题均可在学习网站(<http://acm.hunnu.edu.cn/online/C++/Index.htm>)进行测试。

本书特别适用做计算机专业和相关专业的教材,作为 ACM 国际大学生程序设计竞赛入门教材,也可作为各类考试培训和 C++ 程序设计自学教材。

图书在版编目(CIP)数据

C++ 程序设计教程/瞿绍军 刘 宏 主编. —武汉:华中科技大学出版社,2010.8
ISBN 978-7-5609-6365-5

I. C… II. ①瞿… ②刘… III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2010)第 127796 号

C++ 程序设计教程

瞿绍军 刘 宏 主编

责任编辑:熊 慧

封面设计:范翠璇

责任校对:周 娟

责任监印:熊庆玉

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:华中科技大学惠友文印中心

印 刷:武汉首壹印务有限公司

开 本:787mm×1092mm 1/16

印 张:17.5

字 数:408 千字

版 次:2010 年 8 月第 1 版第 1 次印刷

定 价:25.80 元



华中出版

本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前 言

C++语言是目前最流行的面向对象的程序设计语言之一。它既支持传统的面向过程的程序设计方法,也支持面向对象的程序设计方法。它是Linux和Unix下编程的最主要的语言之一,也是嵌入式开发最常用的编程语言之一。C++全面兼容C,熟悉C的程序员仅需学习C++语言的面向对象特征,就可很快地用C++编写程序。

本书是一本通过编程实践引导学生掌握C++程序开发的教材。我们组织了多位长期从事程序设计、数据结构、面向对象程序设计和计算机算法设计课程教学的老师参与本书编写,其中部分老师还是本校的ACM国际大学生程序设计竞赛集训队的教练和指导老师,他们都有着丰富的教学和编程经验。本书在编写过程中力求将复杂的概念用简洁、通俗的语言来描述,做到深入浅出、循序渐进,从而使学生能体会到学习编程的乐趣。

本书将ACM国际大学生程序设计竞赛引入课程学习之中,使学生从编程入门开始就养成良好的编程习惯和编程思维,强化学生对实际问题分析能力和解决问题能力的培养,激发学生对编程的兴趣,达到以教学促竞赛,以竞赛强化教学的目的。

ACM国际大学生程序设计竞赛是由国际计算机界具有悠久历史的权威性组织ACM学会(Association for Computing Machinery)主办的,是目前世界上公认的规模最大、水平最高、参与人数最多的大学生程序设计竞赛之一,其宗旨是使大学生能通过计算机充分展示自己分析问题和解决问题的能力。现在各个高校都非常重视计算机程序设计竞赛。

与本书配套的教材《C++程序设计教程习题答案和实验指导》提供了本课程的实验内容、习题参考答案和模拟试卷。所有习题和程序均按照ACM国际大学生程序设计竞赛要求进行设计,编者进行了严格的测试,验证了程序的正确性。

参与本书编写的人员有瞿绍军、刘宏、邱建雄、谢中科、何迎生、张楚才、戴经国、唐赞玉、张引琼、张历卓、彭华。

本书吸收了国内外近几年出版的同类教材的优点,内容丰富,特别适合用做计算机专业和相关专业的教材,作为ACM国际大学生程序设计竞赛入门教材,也可作为各类考试培训和C++自学教材。

为方便教师的教学,本书配有电子教案和源程序,您在使用过程中有任何疑问可发邮件与我们联系(Email:powerhope@tom.com)。

编 者

2010年3月

目 录

第 1 章 C++ 语言概述	(1)
1.1 C++ 语言简介	(1)
1.1.1 C++ 语言的发展	(1)
1.1.2 C++ 语言的特点	(1)
1.2 C++ 程序基本结构	(2)
1.3 C++ 程序的开发环境	(3)
1.3.1 Turbo C++	(4)
1.3.2 Visual C++	(6)
1.3.3 Dev-C++	(9)
1.4 ACM 国际大学生程序设计竞赛	(11)
1.4.1 ACM 国际大学生程序设计竞赛简介	(11)
1.4.2 竞赛规则	(12)
1.4.3 自动评测系统	(13)
1.4.4 竞赛学习资源	(14)
习题 1	(15)
第 2 章 C++ 语言编程基础	(16)
2.1 C++ 语言词法	(16)
2.1.1 注释	(16)
2.1.2 关键字	(17)
2.1.3 标识符	(17)
2.1.4 常量	(18)
2.1.5 变量	(18)
2.1.6 运算符	(18)
2.1.7 分隔符	(19)
2.2 基本数据类型	(19)
2.2.1 整型	(20)
2.2.2 浮点型	(22)
2.2.3 字符型	(23)
2.2.4 布尔型	(24)
2.2.5 类型转换	(24)

2.3 运算符与表达式	(25)
2.3.1 算术运算符及其表达式	(26)
2.3.2 赋值运算符及其表达式	(27)
2.3.3 关系运算符及其表达式	(28)
2.3.4 逻辑运算符及其表达式	(28)
2.3.5 位运算符	(29)
2.3.6 条件运算符	(30)
2.3.7 运算符的优先级	(31)
2.4 流程控制语句	(31)
2.4.1 C++语句	(31)
2.4.2 if 语句与条件选择控制	(33)
2.4.3 条件表达式	(38)
2.4.4 switch 语句与多项选择	(38)
2.4.5 while 语句	(40)
2.4.6 do...while 语句	(40)
2.4.7 for 语句	(41)
2.4.8 break 语句和 continue 语句	(42)
2.4.9 循环嵌套	(44)
2.4.10 return 语句	(45)
2.4.11 程序设计综合举例	(45)
2.5 ACM-ICPC 中的输入/输出	(48)
习题 2	(50)
第 3 章 函数与程序结构	(52)
3.1 函数与程序结构概述	(52)
3.2 函数的定义与声明	(53)
3.2.1 函数的定义	(53)
3.2.2 函数声明与函数原型	(55)
3.3 函数参数和函数调用	(56)
3.3.1 函数形式参数和实际参数	(56)
3.3.2 函数的返回值	(57)
3.3.3 函数调用	(57)
3.4 函数的嵌套与递归调用	(58)
3.4.1 函数的嵌套调用	(58)
3.4.2 递归调用	(58)
3.5 变量作用域和存储类型	(60)
3.5.1 局部变量与全局变量	(60)

3.5.2 静态存储变量和动态存储变量	(60)
3.6 内联函数	(62)
3.7 重载函数与默认参数函数	(63)
3.7.1 重载函数	(63)
3.7.2 默认参数函数	(63)
3.8 编译预处理	(64)
3.8.1 文件包含	(64)
3.8.2 宏定义	(64)
3.8.3 条件编译	(65)
习题 3	(66)
第 4 章 数组与字符串	(67)
4.1 数组的概念	(67)
4.2 数组的定义	(68)
4.2.1 一维数组	(68)
4.2.2 二维数组	(74)
4.3 数组作为函数的参数	(78)
4.3.1 用数组元素做函数实参	(78)
4.3.2 用数组名做函数参数	(79)
4.3.3 用多维数组名做函数参数	(81)
4.4 数组应用举例	(82)
4.5 字符串	(90)
4.5.1 字符串概念	(90)
4.5.2 字符串函数	(93)
4.5.3 字符串应用举例	(96)
习题 4	(99)
第 5 章 指针	(102)
5.1 指针的概念	(102)
5.2 指针变量	(102)
5.2.1 指针定义	(102)
5.2.2 指针运算符	(103)
5.2.3 引用变量	(104)
5.2.4 多级指针与指针数组	(106)
5.2.5 指针与常量限定符	(108)
5.3 指针与数组	(109)
5.3.1 指针与一维数组	(109)
5.3.2 指针与二维数组	(114)

5.3.3 指针与字符数组	(116)
5.3.4 指针与函数	(117)
5.4 指针运算	(121)
5.5 动态存储分配	(123)
5.5.1 new 操作符	(123)
5.5.2 delete 操作符	(124)
习题 5	(125)
第 6 章 结构体与共用体	(128)
6.1 结构体	(128)
6.1.1 结构体的声明	(128)
6.1.2 结构体变量的引用及初始化赋值	(130)
6.2 嵌套结构体	(131)
6.3 结构体数组	(132)
6.3.1 结构体数组的定义和初始化	(133)
6.3.2 结构体数组成员的引用	(134)
6.4 结构体指针	(135)
6.4.1 指向结构体变量的指针	(135)
6.4.2 指向结构体数组的指针	(136)
6.4.3 用结构体变量和指向结构体变量的指针作为函数参数	(138)
6.5 链表的基本操作	(140)
6.5.1 链表基本知识	(140)
6.5.2 内存动态管理函数	(141)
6.5.3 建立链表	(142)
6.5.4 输出链表	(144)
6.5.5 对链表的删除操作	(145)
6.5.6 对链表的插入操作	(146)
6.5.7 对链表的综合操作	(148)
6.6 共用体	(150)
6.6.1 共用体的概念	(150)
6.6.2 共用型变量的定义	(151)
6.6.3 共用型变量的引用	(152)
6.6.4 共用型数据的特点	(153)
6.6.5 共用型变量的应用	(154)
6.7 枚举类型	(155)
6.8 用 typedef 定义	(158)
习题 6	(159)

第 7 章 类与对象及封装性	(161)
7.1 类的抽象	(161)
7.2 类的定义与对象的生成	(161)
7.3 构造函数和析构函数	(166)
7.4 构造函数的重载	(169)
7.5 对象指针	(171)
习题 7	(172)
第 8 章 类的深入	(174)
8.1 友元函数	(174)
8.2 对象传入函数的讨论	(178)
8.3 函数返回对象的讨论	(181)
8.4 拷贝构造函数	(184)
8.5 this 关键字	(188)
习题 8	(188)
第 9 章 运算符重载和模板	(191)
9.1 使用成员函数的运算符重载	(191)
9.2 友元运算符函数	(195)
9.3 重载关系运算符	(200)
9.4 进一步考查赋值运算符	(201)
9.5 重载 new 和 delete	(203)
9.6 重载[]	(205)
9.7 重载其他运算符	(208)
9.8 模板	(211)
9.8.1 模板的概念	(211)
9.8.2 函数模板	(212)
9.8.3 类模板	(213)
习题 9	(216)
第 10 章 继承性	(218)
10.1 对继承性的理解	(218)
10.2 类的继承过程	(218)
10.3 基类访问控制	(220)
10.4 简单的多重继承	(225)
10.5 构造函数/析构函数的调用顺序	(226)
10.6 给基类构造函数传递参数	(227)
10.7 访问的许可	(229)
10.8 虚基类	(231)

习题 10	(233)
第 11 章 多态性	(235)
11.1 指向派生类型的指针	(235)
11.2 虚函数	(237)
11.3 继承虚函数	(239)
11.4 多态性的优点	(240)
11.5 纯虚函数和抽象类	(241)
习题 11	(244)
第 12 章 输入/输出流	(245)
12.1 C++ 的输入/输出	(245)
12.2 标准输入/输出流	(246)
12.3 文件操作与文件流	(247)
12.4 串流类	(250)
12.5 格式控制	(251)
12.5.1 格式控制符	(251)
12.5.2 流对象的成员函数	(253)
12.6 ACM-ICPC 中的文件输入/输出	(255)
习题 12	(258)
附录 A ASC II 码对照表	(259)
附录 B C/C++ 与标准 C++ 头文件对照表	(260)
附录 C Linux、Unix 下编译 C++ 程序	(261)
参考文献	(265)

第1章 C++语言概述

[本章主要内容] 本章主要介绍 C++ 语言的发展、特点, 面向对象程序设计的基本概念, C++ 程序的基本结构, 几种常用的 C++ 开发环境, 以及 ACM 国际大学生程序设计竞赛的基本情况。

1.1 C++语言简介

1.1.1 C++语言的发展

C 语言是贝尔实验室于 20 世纪 70 年代初研制出来的, 后来又被多次改进, 并出现了多种版本。C 语言既具有高级语言的特点, 即表达力丰富, 可移植性好, 又具有低级语言的一些特点, 即能够很方便地实现汇编级的操作, 目标程序效率较高。刚开始形成的 C 语言受到那些想建立更快、更有效的代码的程序员欢迎。有一位名为 Bjarne Stroustrup 的人却不满足于仅仅是快速生成代码, 他想创建面向对象的 C 语言。他开始对 C 语言的内核进行必要的修改, 使其能满足面向对象模型的要求, C++ 从此产生。

C++ 语言自诞生以来, 经过开发和扩充, 已成一种完全成熟的编程语言。现在 C++ 已被 ANSI、BSI、DIN、一些国家标准机构和 ISO 定为标准。

C++ 标准演变了许多年。C++ 模板则是近几年来对此语言的一种扩展。模板是根据类型参数来产生函数和类的机制的, 有时也称模板为“参数化的类型”。使用模板, 可以设计一个对许多类型的数据进行操作的类, 而不需要为每个类型的数据建立一个单独的类。标准模板库 (standard template library, STL) 和微软的活动模板库 (active template library, ATL) 都是基于这个 C++ 语言扩展的。

C++ 标准可分为两部分: C++ 语言本身和 C++ 标准库。C++ 标准库提供了标准的输入/输出、字符串、容器 (如矢量、列表和映射等)、非数值运算 (如排序、搜索和合并等) 和对数值计算的支持。应该说, C/C++ 包含了相对少的关键字, 很多最有用的函数都来源于库, 用 C++ 标准库实现容器和部分算法就是标准模板库。

1.1.2 C++语言的特点

C++ 语言之所以被人们广泛认可, 是因为它具有许多先进的技术特点。

1. 优越的性能

性能包含两方面: 算法速度和机器代码效率。通俗地讲, 算法是指解决问题的一种方法或一个过程。严格地定义, 算法是由若干条指令组成的无穷序列。例如, 一个算法定义为获取一个字符串, 计算字符串中的字符个数, 并作为结果返回的过程。所以在编程之前必须设计算法, 编写一个快速程序的第一个步骤是设计良好的算法, 以最少的操作步骤得出问题的答案。第二个步骤是选择语言, 这也影响程序的速度。

从性能的角度考虑,用汇编语言编写程序是最佳的选择,它是计算机能理解的自然语言。但是,几乎没有人用汇编语言编写完整的程序,因为这样做极其乏味。另一个选择是用C语言编写。然而,由 Visual C++ 提供的所有工具都产生 C++,而不是 C。使用 Visual C++ 的向导可以生成大量的实用代码,而不必人工编写代码。从编写程序的难易程度和程序的性能综合考虑,C++是最佳的选择。

C++性能良好,因为它被编译为机器代码。对于 VBScript 和 Java 等语言,代码在运行时由程序解释,而且每次运行程序时都要将代码转换为机器码,这样做效率比较低。不仅仅是已编译过的 C++程序运行得较快,而且微软 C++ 编译器已存在多年。这意味着微软的编译器程序员已经把许多优点集中到编译器上,以至于它能产生非常高效的机器码。

2. 全面兼容 C 语言

C++语言保持了C语言简洁、高效和接近汇编语言等特点,同时对C语言的类型系统进行了改革和扩充。C++对C的兼容性体现在许多C代码的程序不必修改就可被C++所使用。为保持这种兼容性,C++也支持面向过程的程序设计。

3. 支持面向对象的方法

C++是一种支持面向对象的程序设计语言(oriented object programming, OOP)。C++代码以类的形式组成,使得应用程序的开发变得十分容易。C++面向对象技术的特征主要体现为封装性、继承性和多态性,本书在以后的章节中将详细介绍。

1.2 C++程序基本结构

在开始学习使用C++语言编程之前,应该了解C++源程序的基本结构,以及如何书写、编译和运行C++程序,以便建立一个总体的印象。

用C++语言编写应用程序,从开始到最后得到结果,需要经过3个过程,即编写源程序、编译和运行。

1. 编写源程序

一个简单的C++应用程序如例1.1所示。

【例 1.1】 一个简单的C++应用程序。

```
/*-----
HelloWorld.cpp
-----*/

#include<iostream>
using namespace std;
int main()
{
    cout<<"HelloWorld"<<endl; //输出一个字符串
    return 0;
}
```

通过这个程序,可以看到C++应用程序还是比较简单的,结构并不复杂。编写C++

程序时必须遵循 C++ 语言的编程原则。一个简单的 C++ 应用程序的基本格式遵循以下几点规定。

(1) C++ 程序是无格式的纯文本文件,可以用任何文本编辑器(例如,记事本、写字板)来编写 C++ 程序。

(2) C++ 程序(源代码)保存为文件时,建议使用默认扩展名 .cpp。文件名最好有一定提示作用,能使人联想到程序内容或功能。

(3) 每个 C++ 程序都由一个或多个函数组成,函数则是具有特定功能的程序模块。对于一个应用程序来讲,还必须有一个 main 函数,且只能有一个 main 函数。该函数标志着执行应用程序时的起始点。其中,关键字 int 表示 main 函数返回整型值。

(4) 任何函数可以有多条语句。例 1.1 的 main 函数有一条语句:

```
cout<<"HelloWorld"<<endl;
```

该语句用来在屏幕上输出一个“HelloWorld”字符串。cout 是 C++ 的一个对象,可通过它的操作符“<<”向标准输出设备输出信息。return 0 是返回语句。

(5) C++ 程序中的每条语句都要以分号“;”结束(包括以后程序中出现的类型说明等)。

(6) 为了增加程序的可读性,程序中可以加入一些注释行,例如,用“//”开头的行。

(7) 在 C++ 程序中,字母的大小写是有区分意义的,因此 main、Main、MAIN 都是不同的名称。

小提示:编写程序的过程中和编译程序前,请及时存盘,以避免意外导致输入的源程序丢失。

2. 编译

当 C++ 程序编写完成后,必须经过 C++ 编译器把 C++ 源程序编译成扩展名为 .obj 的目标文件,然后使用连接工具将目标文件连接为扩展名为 .exe 的应用程序。在 C++ 集成环境中,往往可以通过 Build 命令一次完成这两个步骤。

3. 运行

根据目的的不同,运行可分为应用运行、测试运行和调试运行。应用运行是指程序正式投入使用后的运行,目的是通过程序的运行完成预先设定的功能,从而获得相应的效益。测试运行是指应用运行前的试运行,是为了验证整个应用系统的正确性,如果发现错误,应进一步判断错误的原因和产生错误的大致位置,以便加以纠正。调试运行则是专门为验证某段程序的正确性而进行的,运行时,通过输入一些特定的数据,观察程序是否产生预期的输出。如果没有产生预期的输出,则通过程序跟踪方法,观察程序是否按预期的流程运行,程序中的某些变量的值是否如预期的那样改变,从而判定出错的具体原因和位置,再加以纠正。

1.3 C++ 程序的开发环境

目前,比较流行的 C++ 程序集成开发环境有基于 Windows 平台的 Microsoft Visual C++、Microsoft Visual Studio 2008、Borland C++ Builder 和 Dev-C++, 基于

DOS 平台的 Turbo C++ 和 Borland C++，基于 Windows 平台和 Linux 及 Unix 下的 Eclipse、Codeblocks 和 Netbeans 等集成开发环境 (IDE)。下面对 Turbo C++、Visual C++ 和 Dev-C++ 开发环境的使用作简要介绍。

1.3.1 Turbo C++

Turbo C++ 3.0 软件是 Borland 公司在 1992 年推出的强大的 C 语言程序设计与 C++ 面向对象程序设计的集成开发工具。它只需要修改一个设置选项，就能够在同一个集成开发环境下设计和编译以标准 C 和 C++ 语法设计的程序文件。

1. 启动 Turbo C++

当 Turbo C++ 成功安装后，将自动在其安装目录下建立一个 BIN 子目录，该子目录下的 TC.exe 为 Turbo C++ 的启动程序，运行该程序可进入 Turbo C++ 集成开发环境，如图 1.1 所示。

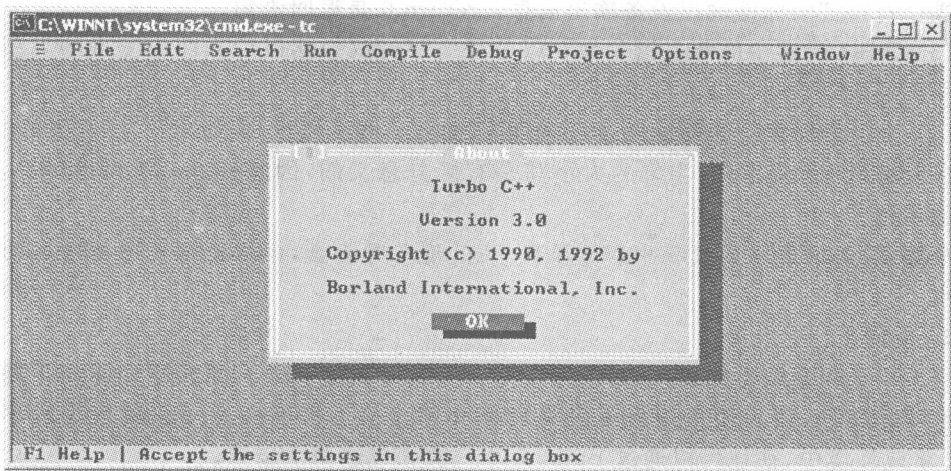


图 1.1 Turbo C++ 启动界面

2. Turbo C++ 集成开发环境

进入 Turbo C++ 集成开发环境后，可通过“File”菜单下的“New”子菜单新建 C++ 程序。图 1.2 演示了例 1.1 所示程序经编辑后并被保存为 first.cpp 的情况。

3. C++ 程序编译和连接

在 Turbo C++ 集成环境中将源程序编辑完毕并且保存后，可以先使用 Alt+C 组合键打开“Compile”菜单，选择“Compile”子菜单（或者使用快捷键 Alt+F9）对源程序进行编译，如图 1.3 所示，得到目标文件，然后选择“Link”子菜单将目标文件连接为可执行程序文件。如果出现错误，则检查、修改程序，然后重复上述步骤。

4. C++ 程序执行和结果查看

C++ 程序在 Turbo C++ 集成环境中编译、连接成功之后，可以使用 Alt+R 组合键打开“Run”菜单，选择“Run”子菜单（或者使用 Ctrl+F9 快捷键）来运行程序，如图 1.4 所示。

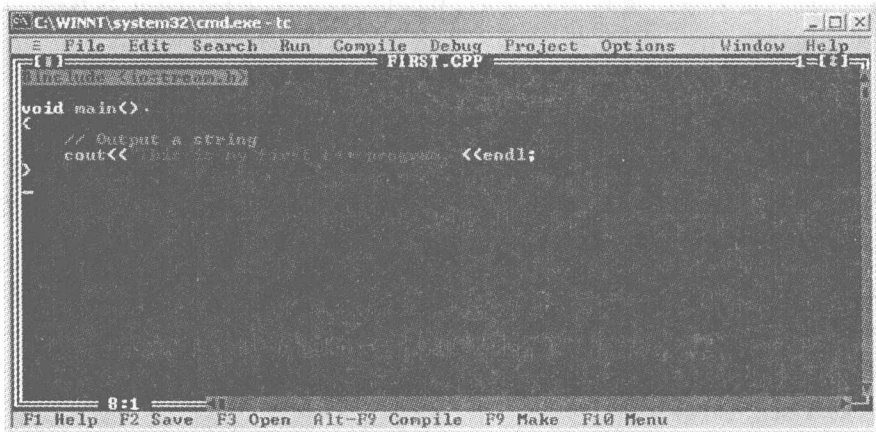


图 1.2 Turbo C++程序编辑器

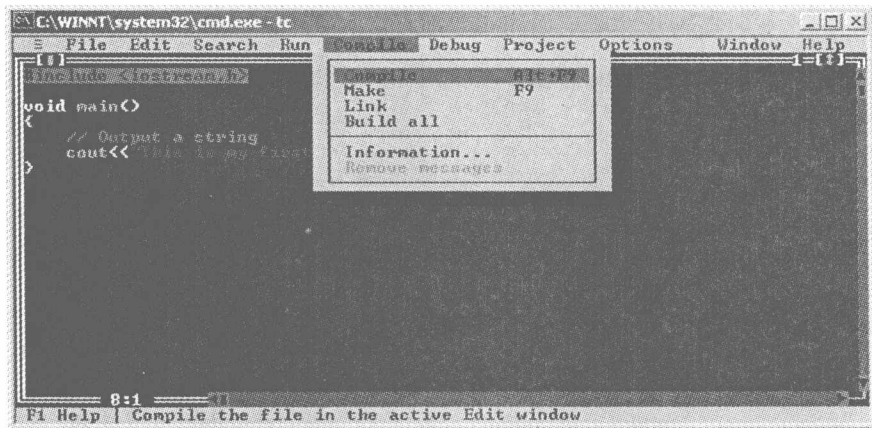


图 1.3 C++程序的编译和连接

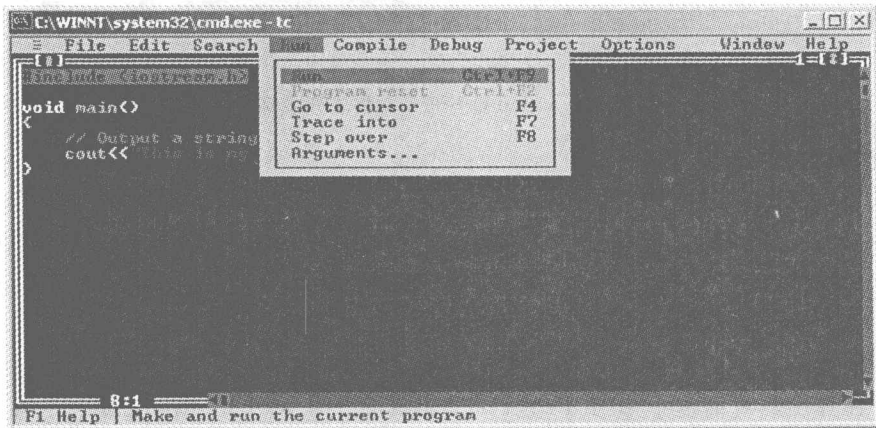


图 1.4 Turbo C++运行菜单

程序运行后,结果输出窗口需要使用“Windows”菜单下的“User Screen”子菜单查看,或者通过快捷键 Alt+F5 查看,然后按任意键返回集成环境界面。

1.3.2 Visual C++

Visual C++是美国微软公司最新推出的可视化 C++开发工具,是目前计算机开发者首选的 C++开发环境。它支持最新的 C++标准,它的可视化工具和开发向导使 C++应用开发变得非常方便快捷。

Visual C++已经从 Visual C++ 1.0 发展到最新的 Visual Studio 2008 版本。本节以 Visual C++ 6.0 为背景简单介绍 Visual C++的使用方法,Visual Studio 2008 的使用将在《C++程序设计教程习题答案和实验指导》中介绍。

1. 启动 Visual C++

当 Visual C++成功安装后,在 Windows 桌面依次选择“开始”→“所有程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”,可以启动 Visual C++ 6.0。Visual C++ 6.0 的集成开发环境如图 1.5 所示。

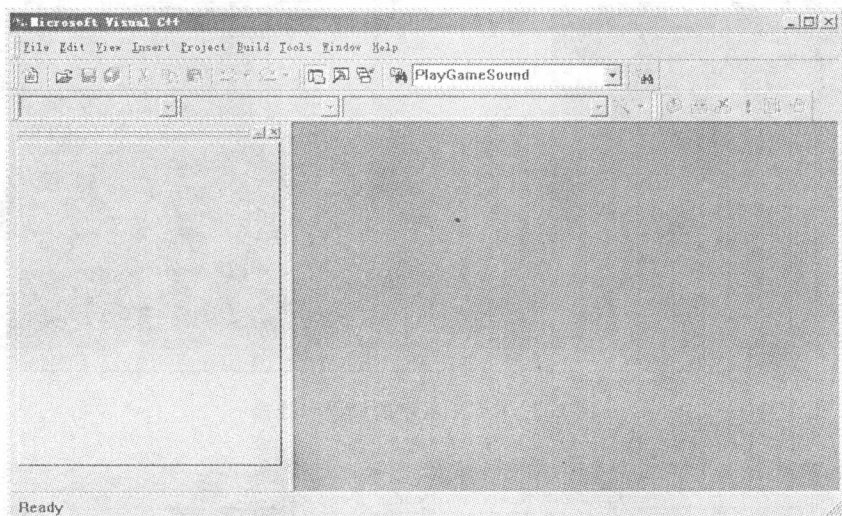


图 1.5 Visual C++ 6.0 集成开发环境

2. 创建工程

在 Visual C++环境中,开发应用程序的第一步是创建一个工程。Visual C++用工程组织维护应用程序。工程文件保存了与工程有关的信息。每个工程都保存在自己的目录中。每个工程目录包括一个工作区文件(.dsw)、一个工程文件(.dsp),至少一个 C++程序文件(.cpp)和 C++头文件(.h)。

(1) 依次单击“File”→“New”,如图 1.6 所示。

(2) 在弹出的对话框中单击“Projects”选项卡,选中“Win32 Application”,在“Project name”文本框中输入工程名,在“Location”文本框中选择工程保存的位置,单击“OK”按钮,如图 1.7 所示。

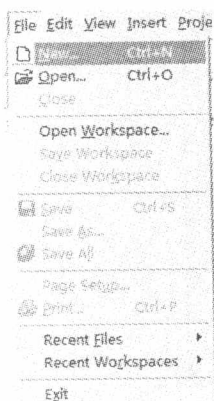


图 1.6 “File”菜单

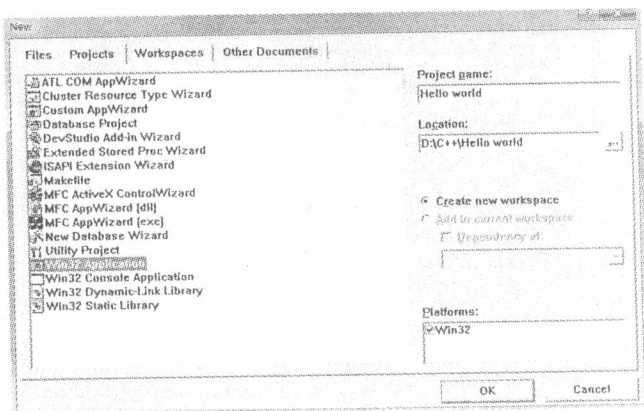


图 1.7 Visual C++ 6.0 向导

(3) 此时出现如图 1.8 所示的“Win32 Application-Step 1 of 1”对话框,选择“An empty project”单选项,单击“Finish”按钮。

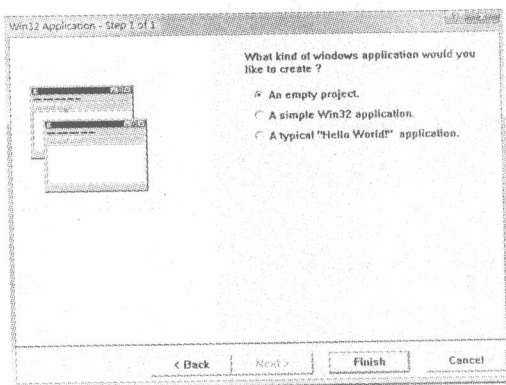


图 1.8 控制台工程向导

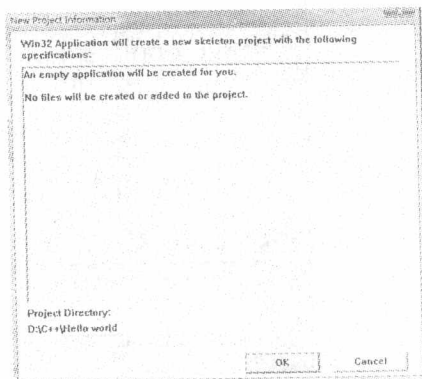


图 1.9 工程信息

(4) 出现如图 1.9 所示的“New Project Information”对话框,单击“OK”按钮,完成工程的创建。

3. 编辑 C++ 源程序

(1) 单击“File”→“New”,在弹出的对话框中单击“Files”选项卡,选中“C++ Source File”,选中“Add to project”复选框,在“File”文本框中输入文件名,然后在“Location”文本框中选择文件保存的位置(用缺省值即可,即和工程保存在同一位置)。最后单击“OK”按钮,如图 1.10 所示。

(2) 在编辑区输入如图 1.11 所示代码。输入完毕后,单击“File”菜单下的“Save”子菜单,保存代码。

4. 编译和运行

(1) 单击工具栏(见图 1.12)中的“Compile”图标或选择“Build”菜单下的“Compile...”子菜单,或按快捷键 Ctrl+F7,如图 1.13 所示。

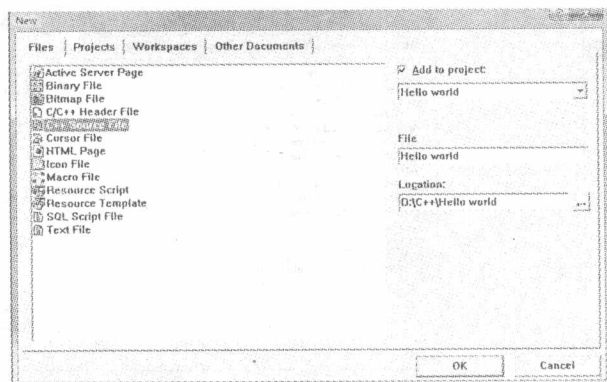


图 1.10 新建 C++ 源程序文件

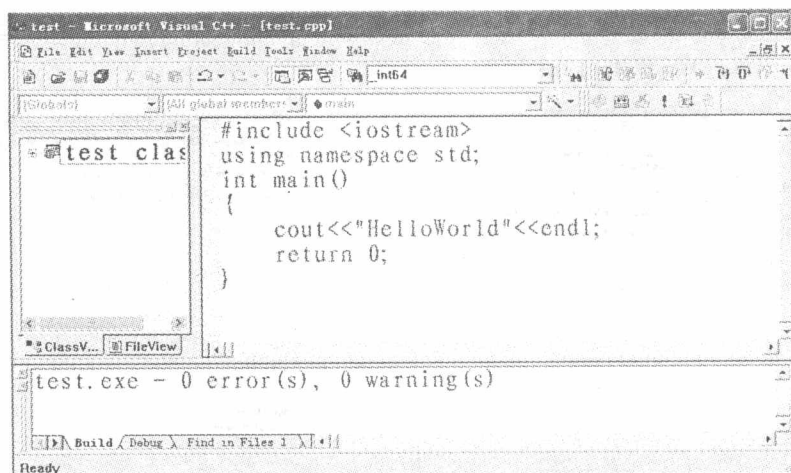


图 1.11 编辑 C++ 源程序



图 1.12 工具栏

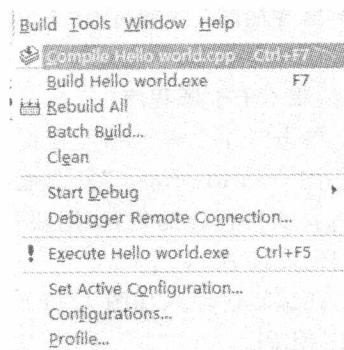


图 1.13 “Build”菜单

(2) 如果编译成功,单击工具栏(见图 1.12)中的“Build”图标,或选择“Build”菜单下的“Build...”子菜单或按快捷键 F7。