



全国信息化计算机应用技术资格认证推荐教材



翰子昂实训体系第一阶段

C 程序设计案例教程

高寿柏 主编

山东友谊出版社



全国信息化计算机应用技术资格认证推荐教材



C 程序设计案例教程

高寿柏 主编

山东友谊出版社



图书在版编目 (CIP) 数据

翰子昂实训体系第一阶段 / 高寿柏主编. — 济南:

山东友谊出版社, 2008.5

ISBN 978-7-80737-320-9

I. 翰子昂… II. 高… III. 软件开发—人才—培养—教材
IV. TP311.52

中国版本图书馆 CIP 数据核字 (2008) 第 060164 号

翰子昂实训体系第一阶段

C 程序设计案例教程

高寿柏 主编

主 管: 山东出版集团

集团网址: www.sdpress.com.cn

出版发行: 山东友谊出版社

地 址: 济南市胜利大街 39 号 邮政编码: 250001

电 话: 总编室 (0531) 82098756 82098142

发行部 (0531) 82098035 (传真)

印 刷: 济南希尔康印务有限公司

版 次: 2008 年 5 月第 1 版

印 次: 2008 年 5 月第 1 次印刷

规 格: 889mm×1194mm 16 开本

印 张: 10.5

字 数: 210 千字 (本册)

定 价: 360.00 元 (全五册)

(如印装质量有问题, 请与出版社总编室联系调换)

翰子昂实训体系软件理论教材丛书序

本套丛书是翰子昂实训体系在软件项目实训前的技术准备教材，与项目实训教材丛书一起组成了一个较完善的实训教材体系。丛书采用案例教学的形式，带领学员以解决项目问题为目标去学习知识，掌握技术细节。从而使学员在学习知识的同时，掌握整合技术点，为软件开发寻找解决方案的思路与途径。

软件开发要求从业人员熟悉软件开发的行业规范，具有良好的实践基础，有自主学习意识、独立思考的能力。案例教学法就是把软件开发对知识技术的要求带入理论知识点的学习过程，让学员带着要解决的问题去学习知识，逐步形成自己对问题的理解，建立问题与知识点间的联系，形成有效解决方案，从而提高自己分析和解决问题的能力。

案例教学法是一种推动式的教学方法，能创建良好宽松的学习氛围。传统的教学方法是拉动式的，使学员始终处于被动吸收的学习之中，不能培养学员主动思考及独立解决问题的能力。案例教学引进项目实际要求，学员解决问题的过程就是按照自己的理解去寻找相关知识与技术点的过程。在这一过程中，项目的规范与要求是学员要达到的目标，教师指导及相关资料的辅助是学员学习的推动力。项目完成后，学员掌握了问题所涉及的所有知识，形成了自己的解决方案，培养了自己主动思考的学习和工作习惯。

丛书以专题的形式呈现一系列知识结构，在每个专题中又以案例的形式描述了一个完整的实现过程，案例之后是所用知识点的讲解，最后是这个案例的扩展实现。学员在使用丛书时，应首先按照书中的步骤，逐步实现案例，遇到不了解的知识点时，再去参考案例后面的知识点讲解。通过这样主动探求，逐步深入的过程，来实现自己对知识点从感性认识到理性掌握的转变。案例完成后，学员还可以参考案例后的扩展内容进一步深化学习。由此，学员在完成一个案例时，不仅掌握了项目包含的技术知识，还可以了解与之相关的知识与技术。

软件可能是人类做出来的最复杂的东西了，软件开发也是一个复杂的过程，软件开发的讲授就更是复杂的工程了，笔者很高兴看到这套丛书将这—个复杂的过程变得清晰易懂，简单明了。这套丛书是通向正确方向的一条路径。

在丛书成书的思路整理、高校与企业调研、结构设计与编写成书的过程中，得到了许多高校研究与教学人员的大力支持，尤其是华东师范大学职业与成人教育研究所徐国庆博士的指导与建议，使我们能更加清晰地把握丛书定位与方向；

还有更多的企业中资深软件工程师和高校一线教师，亲自参加了技术知识的组织与书稿的写作。在此，对他们辛勤的劳动致以最真诚的感谢。

尽管我们始终追求尽善尽美，以便向您提供最新及最全面的知识与技术，但由于作者水平有限，疏漏之处仍在所难免，恳切希望得到您对本教材的建议。

翰子昂实训体系软件理论教材丛书编委成员，主编：高寿柏、宁东，副主编：任春梅、印晓新。

北京翰子昂教育科技有限公司

网址：<http://www.hands-on.com.cn>

2007 年 12 月

前 言

C 语言程序设计课程的目的是通过运用 C 语言开展程序设计实践,使读者具备应用高级语言开展程序设计的初步能力。本书以案例为主,由案例引导理论,介绍 C 语言的词法和语法。本书以 Microsoft Visual C++ 6.0 为开发工具,讲解了:

专题一: C 语言概述和算法

案例一:通过 Hello world 程序,讲解 C 语言的发展史、C 语言的特点、Microsoft Visual C++ 6.0 开发工具、main 函数。

案例二:以沏茶为例子,讲解算法的概念、算法的特性、流程图和伪代码。

专题二:数据类型、运算符与表达式预处理

案例一: C 语言的基本数据类型、变量和常量、变量的初始化和赋值语句。

案例二: C 语言的表达式、运算符、表达式的赋值和自增/自减表达式。

专题三:输入与输出:格式化输入和输出、字符数据输入和输出。

专题四:选择结构程序设计: C 语言中的关系和逻辑运算符、关系和逻辑表达式、if 语句和 switch 语句。

专题五:循环控制:循环的概念, while、for、do...while 语句。

专题六:数组: C 语言中数组的概念,一维、二维、字符数组的定义和引用。

专题七:函数

案例一:函数的概念,函数的一般形式,函数的调用及返回值。

案例二:函数的参数,变量的作用域,函数的嵌套和递归调用。

专题八:指针

案例一:指针的概念、指针的定义和使用,以及变量的指针和指针的变量。

案例二:数组与指针、字符串与指针。

专题九:结构体:结构体的概念、结构体变量的定义和引用,结构体数组的定义和引用。

专题十:文件:文件概述,文件类型指针,文件的打开、定位、读写、关闭。

本书编委成员,主编:王艳华、道鹏,副主编:王秀青、吴升刚、张永涛。

北京翰子昂教育科技有限公司

2007 年 12 月

目 录

专题一 C 语言概述和算法	1
教学目标	1
案例一 hello world 程序	2
1. 教学目标	2
2. 工作任务	2
3. 相关实践知识	2
4. 相关理论知识	4
4.1 C 语言发展史	4
4.2 C 语言的特点	4
4.3 VC++ 的集成开发环境	5
4.4 main 函数	7
5. 实验	7
6. 课后作业	8
案例二 沏茶	9
1. 教学目标	9
2. 工作任务	9
3. 相关实践知识	9
4. 相关理论知识	11
4.1 算法 (algorithm)	11
4.2 流程图	12
4.3 伪代码	14
5. 提高	15
6. 实验	18
7. 课后作业	21
专题二 数据类型、运算符与表达式预处理	23
教学目标	23
案例一 译码程序	24
1. 教学目标	24
2. 工作任务	24
3. 相关实践知识	24
4. 相关理论知识	26
4.1 数据类型概述	26
4.2 变量和常量	27
4.3 基本数据类型	27
4.4 变量的初始化	30
4.5 赋值语句	30
5. 实验	30
6. 课后作业	31
案例二 自增自减表达式测试程序	33
1. 教学目标	33

2. 工作任务.....	33
3. 相关实践知识.....	33
4. 相关理论知识.....	35
4.1 表达式和算术表达式.....	35
4.2 赋值表达式.....	35
4.3 运算符.....	36
4.4 自增(++)、自减(--)运算.....	38
5. 实验.....	38
6. 课后作业.....	39
专题三 输入与输出.....	41
1. 教学目标.....	41
2. 工作任务.....	41
3. 相关实践知识.....	41
4. 相关理论知识.....	44
4.1 格式化输出函数: printf()	44
4.2 格式化输入函数: scanf().....	46
4.3 字符输入函数: getchar()	46
4.4 字符输出函数: putchar()	47
5. 实验.....	47
6. 课后作业.....	47
专题四 选择结构程序设计.....	49
1. 教学目标.....	49
2. 工作任务.....	49
3. 相关实践知识.....	49
4. 相关理论知识.....	54
4.1 关系运算符.....	54
4.2 关系表达式.....	54
4.3 逻辑运算符.....	55
4.4 逻辑表达式.....	55
4.5 if 语句	55
4.6 switch 语句和 break 语句.....	58
4.7 条件运算符.....	59
5. 实验.....	60
6. 课后作业.....	60
专题五 循环控制.....	63
1. 教学目标.....	63
2. 工作任务.....	63
3. 相关实践知识.....	63
4. 相关理论知识.....	67
4.1 for 语句	67
4.2 while 语句.....	69
4.3 do...while 语句.....	69

4.4 break 语句.....	70
4.5 continue 语句.....	70
4.6 goto 语句.....	70
4.7 循环的嵌套.....	71
5. 实验.....	71
6. 课后作业.....	72
专题六 数组.....	73
1. 教学目标.....	73
2. 工作任务.....	73
3. 相关实践知识.....	73
4. 相关理论知识.....	77
4.1 数组的基本概念.....	77
4.2 数组的定义.....	77
4.3 数组的初始化.....	78
4.4 数组元素的引用.....	79
4.5 字符数组.....	80
5. 实验.....	82
6. 课后作业.....	84
专题七 函数.....	85
教学目标.....	85
案例一 判断素数.....	86
1. 教学目标.....	86
2. 工作任务.....	86
3. 相关实践知识.....	86
4. 相关理论知识.....	89
4.1 函数的概念.....	89
4.2 函数的一般形式.....	90
4.3 函数的调用.....	90
4.4 函数的返回值.....	91
5. 实验.....	92
6. 课后作业.....	92
案例二 汉诺塔问题.....	93
1. 教学目标.....	93
2. 工作任务.....	93
3. 相关实践知识.....	93
4. 相关理论知识.....	96
4.1 形式参数和实际参数.....	96
4.2 函数的嵌套调用.....	97
4.3 函数的递归调用.....	98
4.4 变量的作用域.....	98
5. 实验.....	99
6. 课后作业.....	99

专题八 指针.....	101
教学目标.....	101
案例一 指针.....	102
1. 教学目标.....	102
2. 工作任务.....	102
3. 相关实践知识.....	102
4. 相关理论知识.....	104
4.1 指针简介.....	104
4.2 指针变量的类型说明.....	105
4.3 指针变量的赋值.....	105
4.4 指针运算符.....	106
4.5 指针变量的运算.....	106
5. 实验.....	107
6. 课后作业.....	108
案例二 数组与指针、字符串与指针.....	109
1. 教学目标.....	109
2. 工作任务.....	109
3. 相关实践知识.....	109
3.1 数组与指针案例.....	109
3.2 字符串指针案例.....	111
4. 相关理论知识.....	113
4.1 一维数组的指针.....	113
4.2 指向数组的指针.....	114
4.3 字符串与指针.....	115
5. 实验.....	115
6. 课后作业.....	118
专题九 结构体.....	119
1. 教学目标.....	119
2. 工作任务.....	119
3. 相关实践知识.....	119
4. 相关理论知识.....	123
4.1 结构体的基本概念.....	123
4.2 结构体的定义.....	123
4.3 声明结构体变量.....	124
4.4 结构体变量的初始化.....	125
4.5 访问结构体成员.....	125
4.6 结构体数组.....	125
5. 实验.....	126
6. 课后作业.....	129
专题十 文件操作.....	131
1. 教学目标.....	131
2. 工作任务.....	131

3. 相关实践知识.....	131
4. 相关理论知识.....	139
4.1 文件 (File)	139
4.2 打开和关闭文件.....	140
4.3 顺序读写文件.....	142
4.4 随机读写文件.....	152
4.5 文件检测函数.....	153
5. 实验.....	154
6. 课后作业.....	155

专题一 C 语言概述和算法

教学目标

了解 C 语言发展史

掌握 C 语言特点

熟悉 C 语言的编译和执行环境 Visual C++ 6.0

熟练掌握 main 函数的使用

理解算法，掌握算法的概念和特点，掌握算法设计的基本目标

掌握流程图，会使用流程图描述算法

理解伪代码，会使用伪代码描述算法

案例一 hello world 程序

1. 教学目标

- 1.1 熟练掌握 main 函数的使用；
- 1.2 熟悉 C 语言的编译和执行环境 Visual C++ 6.0。

2. 工作任务

使用 VC++的集成开发环境编写 hello world 程序

3. 相关实践知识

- 1. 在 E 盘上新建文件夹，命名为“VC_EXERCISE”（E:\VC_EXERCISE）。
- 2. 打开 VC++的 IDE 环境，新建一个 C 程序，名为 hello_world.c。如图 1-1:

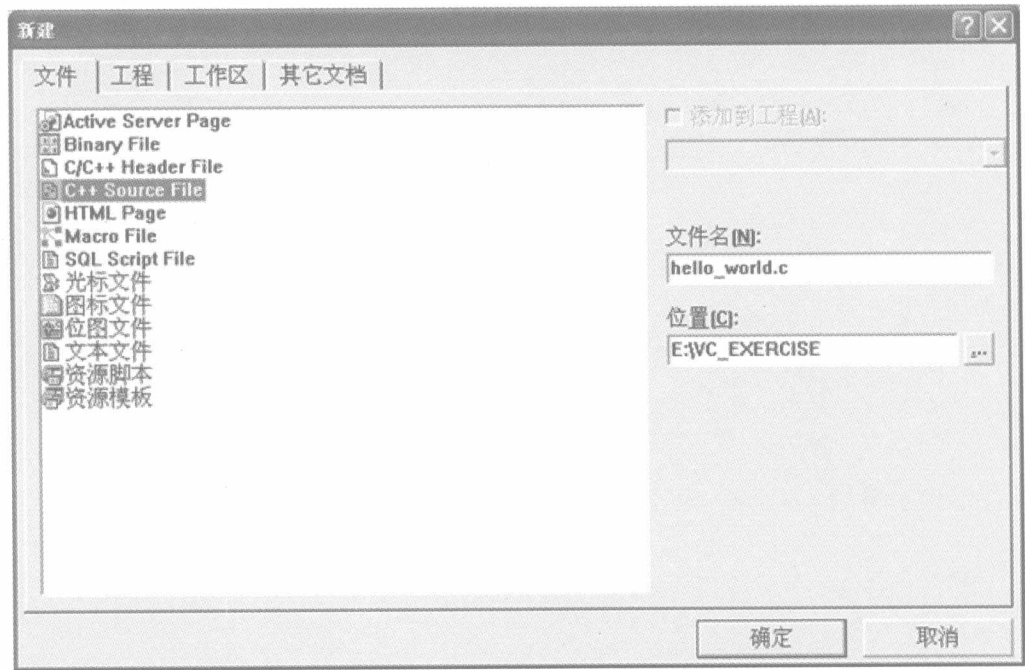


图 1-1 VC++新建文件窗口

- 3. 点击“确定”，进入 VC++的编辑环境，编辑 C 源程序，如下图 1-2 所示:

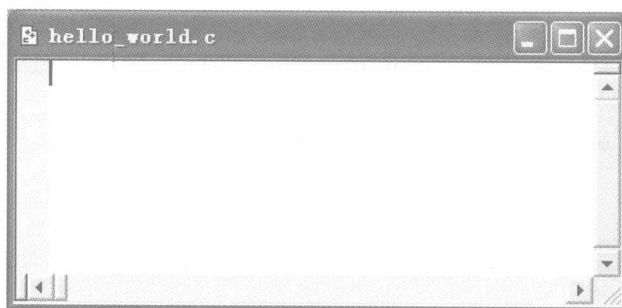


图 1-2 VC++编辑窗口

具体程序代码如下：

```
#include "stdio.h"
void main()
{
    printf("Hello,World!\n");
    printf("Welcome to use Microsoft Word!\n");
}
```



程序代码的一些说明：

程序中用 # 开始的行被称为预处理指令（preprocessing directive），用于与预处理器通讯。程序中 #include 指令使预处理器在代码的当前位置加入标准头文件 stdio.h 的拷贝。

关于预处理指令还将在后续章节详细介绍。

4. 依次选择“编译微型条”中的 Compile 和 Build（分别为第一个和第二个按钮），编译微型条如图 1-3 所示：

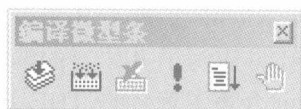


图 1-3 编译微型条

5. 在编译窗口和连接窗口分别出现如下图 1-4 和图 1-5 的界面：

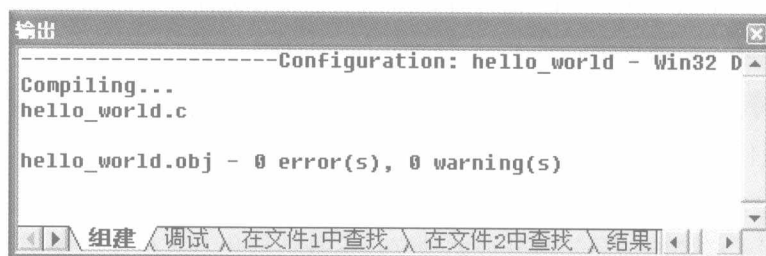


图 1-4 编译窗口

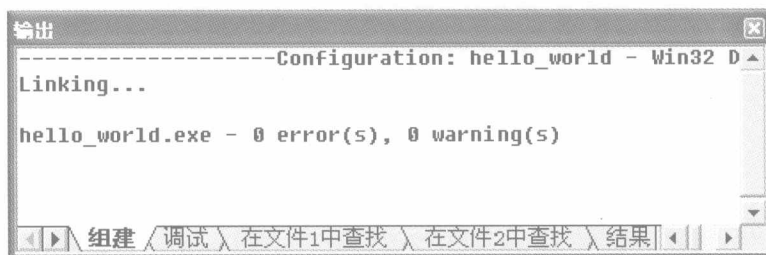


图 1-5 连接窗口

6. 在上面步骤正常执行后，单击“编译微型条”中的“运行”按钮（红色感叹号），出现如下图 1-6 的运行界面：

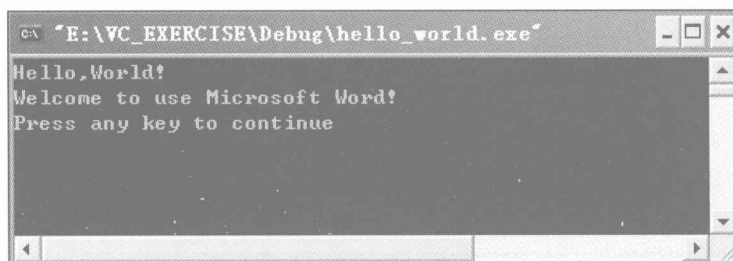


图 1-6 运行界面及运行结果

4. 相关理论知识

4.1 C 语言发展史

C 语言是美国的贝尔实验室在 1969~1973 年开发的。C 语言是由 B 语言衍生而来的，B 语言是贝尔实验室的 Ken Thompson 在 BCPL 语言的基础上开发的，并用它编写了第一个 UNIX 操作系统。BCPL 语言是英国剑桥大学的 Martin Richards 于 60 年代在美国 MIT 时设计的。

1971 年，贝尔实验室的 Dennis Ritchie 扩展了 B 语言（通过增加类型），他称之为 NB，即 New B。在更改了 B 语言的结构，并重写了 B 语言的编译器后，Ritchie 称他的新语言为 C。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《THE C PROGRAMMING LANGUAGE》一书。通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的标准 C 语言。1983 年，出现了许多 C 语言的版本。美国国家标准化协会（ANSI）在 1989 年出版了标准的 C 语言，称之为 ANSI C。C 语言的强大功能和其各方面的优点逐渐为人们所认识，到了八十年代，C 语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到广泛地使用。

4.2 C 语言的特点

C 语言是当代最优秀的程序设计语言之一，它具有如下特点：

1. C 语言是结构化程序设计语言。结构化语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。C 语言是以函数形式提供给用户的，这些函数可方便地调用，并具有多

种循环、条件语句控制程序流向，从而使程序完全结构化。

2. C 语言简洁、紧凑，使用方便、灵活。C 语言一共只有 32 个关键字，9 种控制语句，程序书写形式自由。

3. C 语言既具有高级语言的许多功能又具有低级语言的许多功能，如 C 语言可以像汇编语言一样对位、字节和地址进行操作，即直接对硬件进行操作。

4. C 语言数据结构丰富。C 语言具有各种各样的数据结构，并引入了指针概念，可使程序效率更高。

5. C 语言程序设计自由度大，语法限制不严格。例如，在编译系统中对数组下标越界不做检查。

6. 与汇编语言相比，C 源程序的可移植性好。C 源程序可应用于多种型号的计算机和各种操作系统。

7. C 语言生成目标代码质量高，程序执行效率高。



C 程序要点：

1. 一个 C 语言源程序可以由一个或多个源文件组成。
2. 每个源文件可由一个或多个函数组成。
3. 一个源程序不论由多少个文件组成，都有一个且只能有一个 `main` 函数，即主函数。
4. 源程序中可以有预处理命令（`include` 命令仅为其中的一种），预处理命令通常应放在源程序的最前面。
5. 每一条说明，每一条语句都必须以分号结尾。但预处理命令，函数头和花括号“`}`”之后不能加分号。
6. 标识符、关键字之间必须至少加一个空格以示分隔。若已有明显的分隔符，也可不再加空格来分隔。

4.3 VC++ 的集成开发环境

任何高级语言源程序都要“翻译”成机器语言，才能在机器上运行。“翻译”的方式有两种：一种是解释方式，即对源程序解释一句执行一句；另一种是编译方式，即先把源程序“翻译”成目标程序（用机器代码组成的程序），再经过连接装配后生成可执行文件，最后执行可执行文件而得到结果。

C 语言是一种编译型的程序设计语言，它将源程序翻译成目标程序（机器代码）。运行一个 C 程序，从输入源程序开始，要经过编辑源程序文件（.c）、编译生成目标文件（.obj）、连接生成可执行文件（.exe）和执行四个步骤。

下面简要介绍在 Visual C++ 6.0 环境下开发 C 程序的流程：

1. Visual C++ 6.0 已成功安装，通过点击开始菜单或双击 Visual C++ 6.0 程序图标运行 Visual C++ 6.0 程序，进入 Visual C++ 6.0 集成环境，如图 1-7 所示：

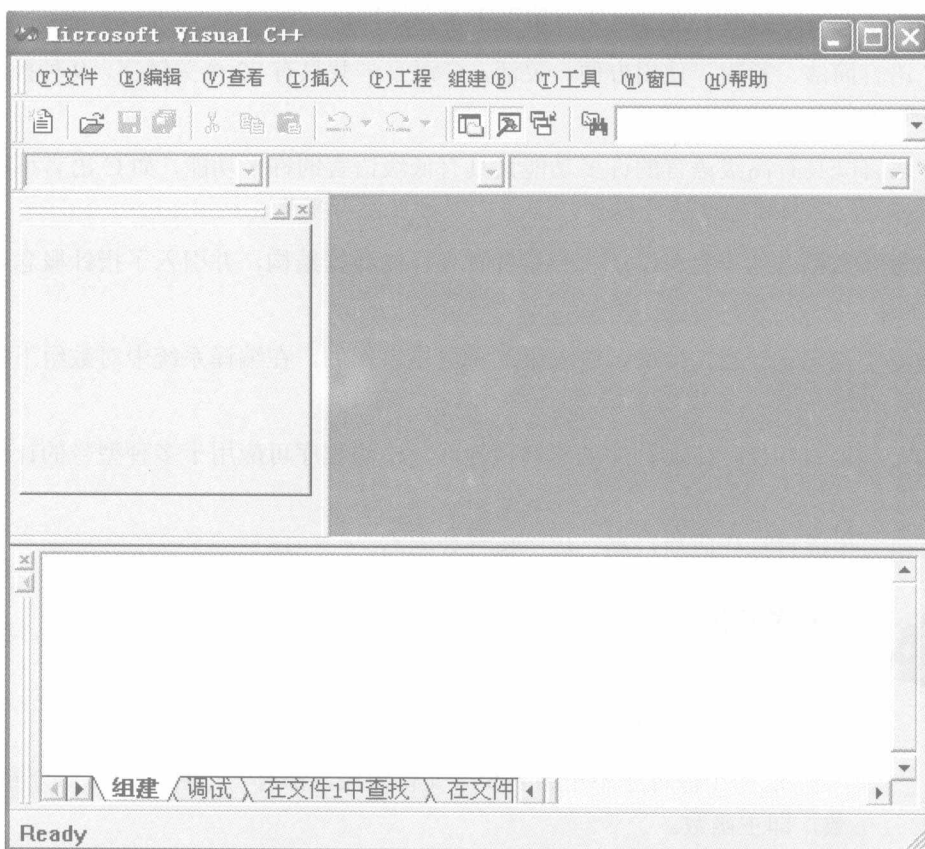


图 1-7 Visual C++ 6.0 集成环境

进入 Visual C++ 6.0 集成环境后，点击【文件】菜单，并在其下拉菜单中选择【新建】命令，调出【新建】对话框。点击新建对话框中的【文件】标签，进入到【新建文件】对话框中。在对话框左侧区域选中【C++ Source File】选项。在对话框右侧【文件名】中填写所要编辑的 C 文件的名称，在【位置】中填写保存新建文件的路径。之后，点击对话框最下端的【确定】按钮，进入 C 文件编辑窗口。

2. 编辑源文件。在编辑状态下，在 C 文件的编辑窗口中根据需要输入或修改源程序。

3. 编译源程序。

选择【组建】菜单，在其下拉菜单中选择【Compile】命令，或是通过工具栏中的第一个编译按钮，对所输入的源程序进行编译，得到扩展名为.obj 的目标程序。在输出框中会显示当前 C 源文件在编译中的错误个数和警告个数。

4. 连接源程序。

选择菜单命令【组建】→【Build】，或通过工具栏中的第二个连接按钮，进行连接操作，之后得到一个扩展名为.exe 的可执行文件。在输出框中会显示连接操作后的可执行文件中的错误个数和警告个数。

5. 执行程序。

选择菜单命令【组建】→【Execute】执行所得到的“.exe”文件。或者通过点击工具栏中的第四个执行按钮.