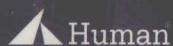


一群中国优秀程序员编程的亲身体会
一盏进入程序设计大门的指路灯
一套全新的多媒体教材
祝贺您的选择

洪恩软件



编程之道

C/C++ 程序设计入门

按初学者特点安排知识点的顺序

语言浅显，实例穿插于讲解之中

在VC环境下学习C和C++

通过章后练习，提高实践能力

多媒体教学光盘

全程语音讲解，演示抽象概念

北京金洪恩电脑有限公司
天津电子出版社

编程之道 系列

C/C++ 程序设计入门

北京金洪恩电脑有限公司 著

天津电子出版社

内 容 提 要

欢迎加入学习 C 和 C++ 程序设计的队伍。C 和 C++ 语言是现今非常流行的程序设计语言，深受广大程序员的青睐。本教材从简单的样例入手，边学边练，重点讲述解决问题的思路；精心设计每一章节的样例，由浅入深、讲解透彻；每章备有相关练习题，让读者能及时巩固。

本教材前九章介绍 C 语言，为后面的程序设计奠定坚实的基础，这部分内容包括：C 语言基础、逻辑运算、控制语句、数组、函数、编译预处理、指针、结构体。后十章介绍 C++ 语言，为今后的实际应用添砖加瓦，这部分内容包括：C++ 语言的新特性、类与对象、静态成员、友元、运算符重载、继承、多态性和虚函数、流、模板。

此外，本教材所有的例程都在 Visual C++ 6.0 下调试通过，读者在学习本教材的过程中，还可以学习如何在 Visual C++ 6.0 中编写控制台程序。

本教材适合程序设计的初学者、对程序设计有初步了解的人士、或想了解 C 和 C++ 语言的读者阅读。

本教材配有多媒体教学光盘，将理论学习中难于理解的内容用生动形象的多媒体形式展现在读者面前，使学习变得更加轻松。

版权所有 翻印必究

教 材 名： C/C++ 程序设计入门

教材编著： 北京金洪恩电脑有限公司

CD 著作者： 北京金洪恩电脑有限公司

出 版 社： 天津电子出版社

开 本： 787 × 1092 1/16

印 张： 26 印张

千 字： 457.6 千字

版 次： 2003 年 8 月第 1 次印刷

本 版 号： ISBN 7-900338-56-X/G · 104

定 价： 48.00 元（1CD）

《C/C++ 程序设计入门》制作群

策 划： 牛承光

稿 件： 宋 红 赵 方

制 作： 牛承光 徐莉蓓

C D 制作： 牛承光 辛 建 徐莉蓓

责任编辑： 刘晓莉

编 校： 张晓宜

美术设计： 郭大卫 王 涛 郑小康

终 审： 于志坚

为普及计算机技术作贡献

原清华大学校长 张孝文 书赠

前言

如果选择了本教材，那么你一定只是想成为一名优秀的程序员或者一位计算机专家，最起码也是希望将来能借此谋生。我们推出《编程之道》系列教材的目的，就是要让你这样的有志之士把愿望变成现实。下面的这些话请各位想入“道”之人阅读，高人自可跳过。

一、《编程之道》系列教材

“编程之道，其道远兮”。了解一些计算机的人，大概都知道这个行业的技术发展日新月异，“恐怕自己多睡了几个懒觉，就要被这个行业的新技术所抛弃”。难道只有那些聪明绝顶的人才能在这个赚钱的行业里淘金吗？其实只要你掌握了其中之“道”，自然就可以大胆地去淘金了。

那么“道”在何处？让我们静下心来，澄清以下观点：

1. 编程只是实现工具

读一些计算机历史的文章，我们可以清楚地知道，计算机是数学家们的发明，他们并非个个都会编程。现今也会看到这样有趣的现象，不少大师级的计算机技术研究者并非谙熟编程。而各种媒体上的炒作和现实中优厚的待遇往往把编程神秘化了。

计算机技术包括了管理信息、多媒体、计算机网络、人工智能、辅助设计等等方面。编程只是这些具体技术在理论研究或者实践中表达算法的过程。编程的人不一定对计算机技术的了解就一定很高深，但要成为一名编程高手，就必须扎实地掌握丰富的计算机技术。

因此首先要明白，编程只是实现想法的工具，而解决问题的方法和思想更重要。

2. 学习编程，莫忘基础

如果你想成为一名优秀的程序员，建议除了学习编程语言、开发工具之外，莫忘扎实地学习如下一些课程：汇编语言、算法和数据结构、计算机体系结构、操作系统原理；还可以补充学习计算机网络、数据库原理等课程。掌握了这些基础知识，才能让你的“编程之道”更宽广、更平坦。当然这里只是列举了几门最基本的课程。

3. 且莫追风、静心修道

不要被那些流行的新技术、新名词所迷惑，不然就要犯狗熊掰玉米的毛病了。

.NET、XML 等等技术固然诱人，能在短时间内让人找到一种满足感，可是如果自己基础不扎实，自然也无法深入其中，酷似云中漫步。

这些“包装精美”的洋货掩盖了许多底层的原理，要想真正学技术还是走下云端，脚踏实地地把 C、C++ 以及前面提到的基础知识先学好，尽管这是一个艰苦的过程（不要梦想

21 天怎样……除非你生来就是搞计算机的)。有了这些基础,学起这些时髦的东西也就是个把星期的事情了。

4. 编程不难!道在基础

比起学英语,学习编程的难度实在太小,毕竟一门编程语言的“单词”只有区区几十个,语法也不多。耗费时间的是学习那些基础的计算机知识,但“道”在其中,自然要学。

《编程之道》系列教材则是为修炼“编程技术之道”的众生设计的经文,希望你在学习这一系列教材时,可以同时学习前面提到的基础课程,或者提前学习它们。

《编程之道》系列教材分为两大类:入门级、提高级。每一级中都涉及一系列应用广泛的编程语言和开发工具。

入门级是为那些没有编程基础或稍有一点基础的人设计的,从最基本的语法到基本应用,照顾的方面较多。提高级是为有编程基础的人设计的,旨在提高某专业方向上的编程技能及开发工具的深入应用,涉及的内容有:数据库、网络、Internet、多媒体等方面。

二、《C/C++ 程序设计入门》教材的内容

《C/C++ 程序设计入门》是《编程之道》系列教材中的首本,其内容都是针对 C 和 C++ 语言初学者的特点编写而成的,因此在语言上尽可能做到言简意赅、清晰流畅。在内容上也做了适当的取舍,舍去了 C 语言中关于共同体、位操作、汇编语句、文件的操作等部分;文件的操作只在 C++ 部分讲到;C++ 中关于异常处理的部分也没有涉及。如果你想学习更全面的 C 和 C++ 的知识可以参阅其它大全类的书籍。

这本教材的具体内容这里就不再赘述,参看目录即可一目了然。

三、图书中的一些约定

黑体文字,正文中的有些词汇或者语句是用黑体印刷的,这些为了让读者能更清楚地看到这些内容而设计的。

程序代码中的粗体字,这是为了强调这条语句或者表达式是这一部分重点讲述的内容。

输出结果中的粗体字,这些字表示是从键盘输入的数据。

四、使用本教材的要求

对于初学者,建议按顺序从头开始学习,不要为 C++ 的美名所惑,从中间开始哦!

当你觉得教材中的一些知识点不易理解时,可以参考配套多媒体光盘中的讲解。

本教材中的范例程序在光盘的 S&K\code 目录下,每一章的范例程序放在一个目录中;其中 C++ 部分中每个范例程序是一个目录,双击每个例程中的.dsw 文件即可打开范例的项目文件(前提是安装好了 Visual C++ 6.0)。

在教材中我们还设计了一些练习题，希望读者能够上机练习这些题目。选择、填空题的答案附在教材的附录部分；判断、改错题请读者上机自己检查；问答题和程序设计题的参考答案放在光盘的 **S&K\key** 目录中，建议打开 **S&K\index.htm** 文件查看，也可以通过光盘主界面上的“源程序及习题解答”按钮打开查看。建议读者先自己尝试做一下程序设计题，然后再对照光盘上的参考答案。

“练从难处练，用从易处用”。我们还是建议读者亲自动手敲代码，“眼过千遍，不如手过一遍”，在学编程的时候同样是有道理的。

学习本教材的过程中，你还可以学习到 **Visual C++ 6.0** 的使用，教材中的范例程序均在 **Visual C++ 6.0** 下调试通过。另外，你也可以选择其它的 C 和 C++ 编译器，如 **Dev C++**（可以到 **www.bloodshed.net** 下载）。

最后，感谢各位读者使用本教材，也希望读者指出本教材中的瑕疵、错漏之处，以便我们在下一版中修正。

我们的技术服务热线：(010) 62634069，E-mail: pcbook@goldhuman.com。

北京金洪恩电脑有限公司

董事长 池宇峰

目 录

第 1 章 C 语言概述及 Visual C++ 6.0 控制台程序

1. 什么是 C 程序	1
2. 如何让我们的 C 程序运行起来!	3
3. 这个程序是怎么被执行的?	7
4. 注意 C 中字母的大小有别	8

第 2 章 基本数据类型、运算符和表达式

一、基本数据类型	10
1. C 语言提供的基本数据类型有哪些	10
2. 常量和变量	10
3. C 语言的基本数据类型及其表示	13
二、运算符和表达式	20
1. 给变量赋初值	20
2. 运算符和表达式	21
3. 数据类型的转换	27

第 3 章 数据的输入输出

一、格式输出、输入函数	31
1. 格式输出函数 printf	31
2. 格式输入函数 scanf	36
二、字符输入输出函数	38
1. 字符输出函数 putchar	38
2. 字符输入函数 getchar	39
三、应用举例	39

第 4 章 逻辑运算和控制语句

一、程序的三种基本结构	45
1. 顺序结构	45
2. 选择结构	45
3. 循环结构	47
4. 程序的流程图	48
二、选择控制语句	49
1. 关系运算符和关系表达式	49
2. 逻辑运算符和逻辑表达式	50
3. 条件运算符和条件表达式	52
4. 条件选择语句 if	53
5. 分支结构 switch	59
三、循环控制语句	63
1. for 循环语句	63
2. while 循环语句	66

3. do while 循环语句.....	68
4. 多重循环语句.....	69
5. break 语句和 continue 语句.....	70
第 5 章 数组	
一、一维数组.....	76
1. 一维数组的定义.....	76
2. 一维数组元素的赋值和初始化.....	77
3. 一维数组的程序举例.....	80
二、二维数组.....	85
1. 二维数组的定义.....	86
2. 二维数组元素的赋值和初始化.....	87
3. 二维数组程序举例.....	89
三、字符数组.....	90
1. 字符数组的定义.....	91
2. 字符数组的初始化.....	91
四、字符串和字符串处理.....	92
1. 字符串的定义.....	92
2. 字符串的初始化.....	92
3. 字符串的输入输出.....	93
4. 字符串处理函数.....	96
第 6 章 函数	
一、函数的定义与调用.....	109
1. 什么是函数.....	109
2. 函数的定义.....	111
3. 函数的声明.....	117
4. 函数的调用.....	117
二、参数传递.....	122
三、递归函数.....	124
四、变量的作用域与生存期.....	126
1. 变量的作用域.....	126
2. 变量的生存期和存储类型.....	129
第 7 章 编译预处理	
一、宏定义.....	135
1. 不带参数的宏定义.....	135
2. 带参数的宏定义.....	137
二、文件包含.....	138
三、多文件程序.....	139
第 8 章 指针	
一、指针.....	142

1. 什么是指针	142
2. 指针的定义	144
3. 指针的运算	149
二、指针与数组	152
1. 指针与数组的关系	152
2. 应用举例	157
3. 指针数组	161
三、指针、数组与函数	165
1. 指针作函数的参数	165
2. 数组与函数	168
3. 用变量的地址作为函数的实参	170
4. 返回指针的函数	171
四、利用指针动态分配内存	172
 第 9 章 结构体和枚举类型	
一、结构体类型	179
1. 什么是结构体	179
2. 结构体的定义	180
3. 定义结构体变量	181
4. 结构体变量的初始化	184
5. 结构体数组	187
6. 结构体指针	189
7. 结构体与函数	193
二、结构体在链表中的应用	198
1. 什么是链表	198
2. 如何建立链表	200
3. 链表的输出	204
4. 统计链表结点的个数	206
5. 链表结点的查找	206
6. 链表的插入	207
7. 删除一个结点	212
三、枚举类型	214
1. 枚举类型的定义	214
2. 枚举变量的声明	215
3. 枚举类型变量的赋值和使用	215
4. 枚举类型的应用举例	217
 第 10 章 C++语言基础部分	
一、编写第一个 C++程序	223
1. Hello.cpp 你的第一个 C++程序	223
2. 运行程序	224
3. Hello.cpp 程序代码分析	225

4. 在 Visual C++ 6.0 下实现基于控制台的 C++ 程序	226
二、C++ 的输入和输出	228
1. 输出流对象 cout 使用介绍	228
2. 输入流对象 cin 使用介绍	231
三、const 常量定义和使用	232
第 11 章 C++ 中函数的新特性	
一、内联函数	235
二、函数原型	239
三、重载函数	239
四、缺省参数	240
五、引用	242
1. 独立引用	243
2. 引用作为参数传递	244
3. 引用作为返回类型	247
4. 引用与指针的区别	248
第 12 章 C++ 中的两个基础概念：类与对象	
一、类和对象	251
1. 新的数据类型“类”的引入	251
2. 类和对象	252
3. 类的定义和实现	253
4. 类的使用	255
5. 类的定义及实现示例程序演示	256
6. 提示：在 Visual C++ 6.0 的程序中下添加一个类	256
二、存取访问控制	261
1. 关键字 public	261
2. 关键字 private	261
3. 关键字 protected	264
三、类与结构体	264
四、应用举例	265
第 13 章 进一步走近“类”	
一、new 和 delete	270
1. new	271
2. delete	272
二、构造函数和析构函数	273
1. 构造函数	273
2. 析构函数	281
三、再谈 new、delete 和 malloc、free	283
四、this 指针	285
五、应用举例	287

第 14 章 静态成员和友元

一、静态成员	296
1. 静态数据成员	296
2. 静态成员函数	300
二、友元	302
1. 友元函数	302
2. 友元类	304
三、应用举例	306

第 15 章 运算符重载

一、明确目标：为什么要进行运算符重载	309
二、以成员函数实现运算符重载	310
1. 重载一元运算符：递增运算符 “++”	311
2. 二元运算符重载：重载加运算符 “+”	319
3. 重载赋值运算符 “=”	321
三、用友元函数重载运算符	324
1. 用友元函数重载加法运算符 “+”	324
2. 重载输出运算符 “<<”	327
四、应用举例	328

第 16 章 继承

一、什么是继承	336
二、实现继承	338
1. 单一继承	338
2. 多重继承	343
3. 多级继承	346
三、构造函数、析构函数与继承	347
1. 何时执行构造函数和析构函数	347
2. 将参数传送给基类的构造函数	349
四、继承与友元及静态成员的关系	353
1. 友元与继承	353
2. 静态成员与继承	355
五、应用举例	356

第 17 章 虚函数与多态性

一、虚函数	362
1. 为什么要引入虚函数	362
2. 虚函数与函数重载的区别	368
3. 虚函数与继承的关系	368
4. 虚函数的层次性	368
5. 析构函数与虚函数	370
二、纯虚函数	371

第 18 章 流, 文件

一、格式化输出	374
1. I/O (输入/输出) 控制符介绍	374
2. 格式化输出举例	375
二、文件流	378
1. 文件的打开和关闭	378
2. 文件读写操作举例	379

第 19 章 模板

一、函数模板	384
二、类模板	387

附录 A Turbo C 2.0 的操作

1. 设置环境目录	393
2. 编写源程序	394
3. 保存和打开 C 程序	394
4. 编译和执行 C 程序	395
5. 获取帮助	395

附录 B ASC II 表

附录 C 常用库函数

一、数值函数	397
1. 算术函数	397
2. 三角函数	397
二、字符函数和字符串函数	398
1. 字符函数	398
2. 字符串函数	399

附录 D 练习答案

第 1 章 C 语言概述及 Visual C++ 6.0 控制台程序

本章要点:

- C 语言的框架
 - Main()函数、注释语句
 - 用 Visual C++ 6.0 编写、编译、执行一个 C 程序
-

欢迎你加入学习 C 和 C++ 程序设计的队伍。

C 和 C++ 语言是现今非常流行的程序设计语言。为何它们如此流行、深受程序员的青睐？自然有它们的好处，而这些好处只有学习了它们，再比较其它的程序设计语言之后才能体会到。

你或许听说 C 语言、C++ 语言比较难学，在这里请你不要害怕，学习它们远比学英文容易得多。学习语言本身并不会花费你多少脑筋，重要的是解决问题的思路。

本书前九章介绍 C 语言，后十章介绍 C++ 语言。你可以把学习 C 语言视作“造砖”，把学习 C++ 语言视作“建房子”。“砖”造得好，建的“房子”才够结实。

闲话少叙，下面我们就开始“造砖”的历程。

1. 什么是 C 程序

让我们首先看一个用 C 语言编写的程序。

程序清单 C01_01.c

```
/* 我的第一个 C 语言程序*/  
#include <stdio.h>  
main()  
{  
    printf("Hello, 我要学习 C 语言.\n");  
}
```

这段程序的作用就是在屏幕上显示一句话：

Hello, 我要学习 C 语言。

看上去 C 语言挺简单，就是一段文字，很简单的几句话，就可以在屏幕上显示一句话了。这样书写好的 C 程序就叫做“源程序”，它是一个文本文件。

不过，即使它很简单，我们也要搞清楚每句话的意思。

第一句：

```
/* 我的第一个 C 语言程序*/
```

只是起到一个说明的作用，C 语言中用“/* */”来说明一个注释语句。当计算机看到“/*”

就开始忽略后面的内容，直到遇见“*/”。

因此在写程序的时候，如果想写一些说明的话，就用“/*”和“*/”包起来就行了。

第二句：

```
#include <stdio.h>
```

`include` 是“包含”的意思，`stdio.h` 是一个文件的名称，它是“standard input/output”的缩写，意思就是“标准输入输出”，“#”是和 `include` 配合在一起使用的符号。

这句话的意思就是：把文件 `stdio.h` 包含到我们的程序中。为什么要这样呢？因为只有这样，第四句：

```
printf("Hello, 我要学习 C 语言.\n");
```

才能执行。

现在你搞不明白这句话的意思也没关系，第 7 章我们还会讲的。你现在只需要知道在编程的时候，先把这一句写在最前面就行了。

第三句：

```
main()
```

这个 `main()` 叫做“main 函数”，样子和数学中的函数 $f(x)$ 、 $g(x)$ 很像，只是用一个单词做函数名，括号中省略了自变量（在程序中就叫做“参数”了）。

`main` 函数是每个 C、C++ 程序所必须有的，因为每个 C、C++ 程序都要从这个函数开始执行。`main` 函数是 C、C++ 程序的入口。`main` 函数的完整写法是：

```
main()
{
    /* 程序语句 */
}
```

所以在上面程序的 `main()` 下面可以看到一对“{ }”，“{”和“}”之间就是具体要做什么的语句，这些语句叫做函数体。

第四句：

```
printf("Hello, 我要学习 C 语言.\n");
```

相信不用解释你也明白这句话的意思了。

`printf()` 也是一个函数，它是定义在 `stdio.h` 文件中的，所以前面要把 `stdio.h` 文件包含上。`printf` 是“print format”的缩写，就是“按一定格式显示”的意思。

引号中的内容就是告诉 `printf` 函数要显示的内容。不过在显示的内容中并没有看到“\n”，原来在 C 中“\n”表示换一行的意思。

语句的最后是一个分号 (;)，它表示一条语句结束。不过注意用“#”开头的语句最后不加分号。

经过上面这一番解释，你差不多认识 C 程序的样子了吧！

下面的一张图描绘了一个基本的 C 程序的框架。

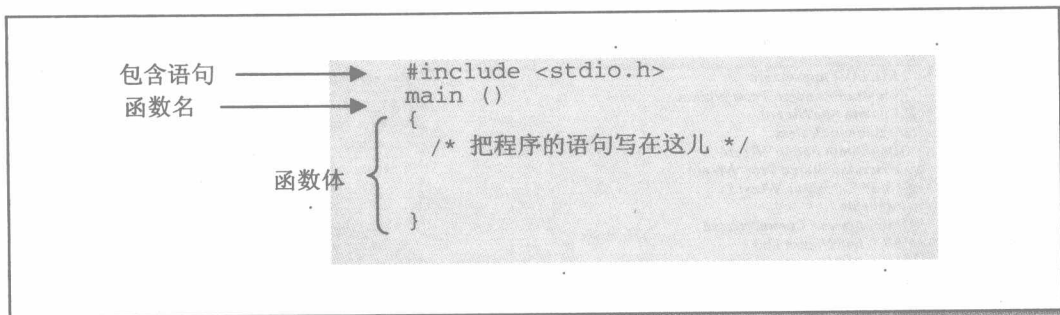


图 1.1 基本的 C 程序的框架

提示：以后你在写程序的时候，可以先把这个框架输入好，然后再开始写其它的部分。当然框架中的汉字就不要输入了。

2. 如何让我们的 C 程序运行起来！

现在你一定很关心，怎么让这段 C 程序运行起来呢？好，下面我们就介绍一下如何用著名的 Visual C++ 6.0 让这段程序运行起来。

接下来你可以跟随下面的图示操作了。

- 1) 启动 Visual C++ 6.0，图 1.2 就是 Visual C++ 6.0 的界面。

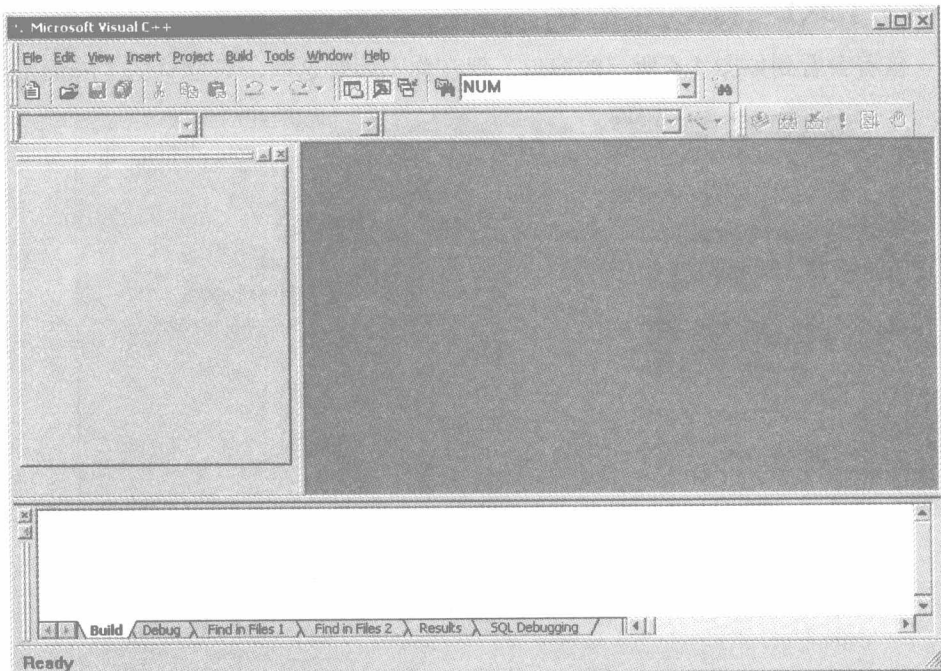


图 1.2 Visual C++ 6.0 的界面

- 2) 打开【File】|【New】，出现如图 1.3 所示的对话框。

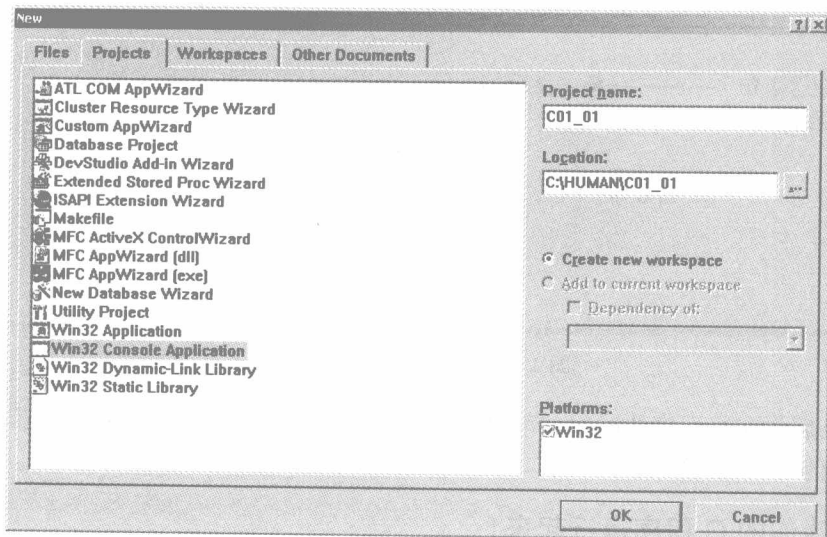


图 1.3 New 对话框

在这个窗口中建立一个新的 C 项目，从【Projects】中选择【Win32 Console Application】，在右边的【Project name】栏中输入程序的名字，比如：C01_01（当然你可以另取名字）。

下面的【Location】是当前程序放在那个目录中，单击右边的按钮可以进行更改。

然后单击【OK】。

3) 接着会看到如图 1.4 所示的窗口。现在你不用管这些，只要单击【Finish】就行了。

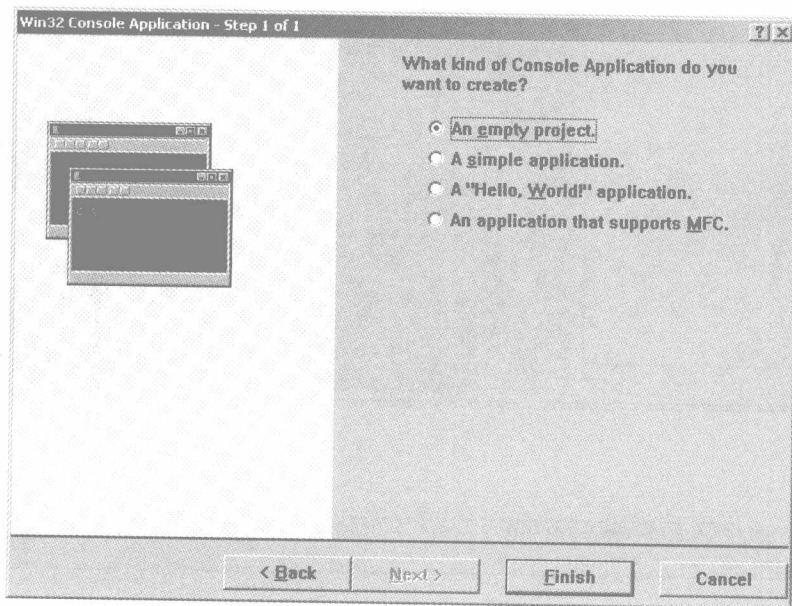


图 1.4 Win32 Console Application 窗口

出现提示窗口后，单击【Finish】。这样就建立了一个项目，现在你先不用管这个项目是什么意思，学一段时间之后你就自然而然地明白了。