

庄益瑞 吴权威 编著

C++

全方位学习

编程更容易，应用更便利，学习更有趣

- 以Microsoft Visual C++为工具，运用较多的范例来引导程序设计的技巧，并配合简易的理论解说应用于解题实务上。
- 每章的精心设计，由浅入深的习题，让读者可以循序渐进地体会C++程序设计的精髓。
- 内容由浅入深，适合读者逐步进阶。
- 适合初次学习程序语言或想彻底了解C++语言的读者阅读。
- 可作为高等院校程序语言设计、面向对象程序设计的教科书。
- 按图施工，各显神通，学练结合，保证成功。



内附范例光盘



Step and Step

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

号 830 19/23 京 (京)

出版说明

C++全方位学习

庄益瑞 吴权威 编著

对于学习程序设计的初学者而言,本书采用了通俗易懂的文字,通过渐进式

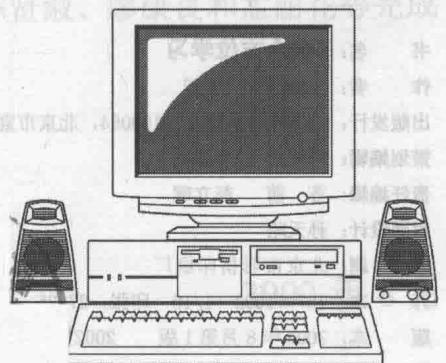
本书非常适合初次学习程序语言或想彻底了解C++的读者阅读,也可作

高等院校教授程序语言、面向对象设计教材。另外,随书所附光盘包含

本书由清华大学股份有限公司出版,中国铁道出版社计算机图书项目

心审定,由戴军、贺军、董亚平、李志伟、陈望淑、唐志和等同志完成了

多的编排工作。



中国铁道出版社

2002·北京

吴权威 庄益瑞

本书附赠光盘,内含本书全部源代码,并附有详细的使用说明。

(京)新登字 063 号

北京市版权局著作合同登记号: 01-2002-2412 号

版 权 声 明

本书中文繁体字版由台湾碁峰资讯股份有限公司出版(2002)。本文中文简体字版经台湾碁峰资讯股份有限公司授权由中国铁道出版社出版(2002)。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

图书在版编目(CIP)数据

C++全方位学习/庄益瑞 吴权威编著. —北京: 中国铁道出版社, 2002. 8

(台湾经典程序设计入门系列)

ISBN 7-113-04816-1

I. C... II. ①庄... ②吴... III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 053856 号

书 名: C++全方位学习

作 者: 庄益瑞 吴权威

出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 郭毅鹏

责任编辑: 苏 茜 彭立辉

封面设计: 孙天昭

印 刷: 北京市彩桥印刷厂

开 本: 787×1092 1/16 印张: 27.25 字数: 652 千

版 本: 2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-04816-1/TP·754

定 价: 46.00 元

中 国 铁 道 出 版 社

京 北 · 5005

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

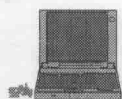
目 录

第 1 章	C 语言的基本概念	1
1-1	程序语言	2
1-1-1	为什么需要程序语言	2
1-1-2	高级语言	3
1-2	C 语言的演化	3
1-3	Microsoft Visual C++ 6.0 的安装	4
1-4	启动程序与画面简介	16
1-4-1	启动 Visual C++	16
1-4-2	Visual C++ 画面概述	17
1-4-3	启动 MSDN Library	19
1-4-4	MSDN Library 画面概述	20
1-5	C 语言程序结构	21
1-5-1	程序设计步骤	21
1-5-2	C 语言程序结构	22
1-6	如何编写 C 语言程序	25
1-6-1	C 语言程序的格式	25
1-6-2	编译程序语言和解释程序语言	25
1-6-3	Visual C++ 程序的编写与编译方法	26
	温故知新	29
	自我突破习题	29
第 2 章	数据类型与表达式	31
2-1	C 语言的数据类型与变量声明	32
2-1-1	常量与变量	32
2-1-2	整数	33
2-1-3	浮点数	34
2-1-4	字符	34
2-2	输入与输出	35
2-3	运算符和表达式	40
2-3-1	数学运算表达式	40
2-3-2	赋值表达式	42
2-3-3	逻辑关系表达式	42
2-3-4	位运算表达式	43
2-3-5	数据类型的转换与强制转换数据类型运算符	47



2-3-6 运算符优先级	48
温故知新	53
自我突破习题	53
第3章 程序流程控制	55
3-1 循环结构	56
3-1-1 for 语句	56
3-1-2 while 语句	64
3-1-3 do...while 语句	68
3-2 条件流程控制	71
3-2-1 if 语句	71
3-2-2 else if 语句	74
3-3 多重分支结构 (switch...case)	77
3-4 中断分支与循环语句	82
3-4-1 中断分支 break 语句	82
3-4-2 continue 语句	85
3-5 条件运算符 (? :)	89
3-6 跳转语句 (goto)	93
温故知新	94
自我突破习题	95
第4章 数组与指针	99
4-1 数组	100
4-1-1 数值数组数据的赋值	100
4-1-2 字符数组数据的赋值	107
4-2 多维数组	111
4-2-1 数值二维数组数据的赋值	112
4-2-2 字符二维数组数据的赋值	117
4-3 指针	119
4-3-1 指针的意义	119
4-3-2 指针的运算	120
4-4 动态分配内存	123
4-4-1 new 配置内存	123
4-4-2 delete 释放内存	127
4-5 指针与数组	129
4-5-1 指针与一维数组	129
4-5-2 指针与多维数组	131
4-5-3 指针与字符数组	133
温故知新	133
自我突破习题	134

第五章 函数与函数库	137
5-1 函数定义	138
5-2 参数传递	140
5-2-1 传值调用	141
5-2-2 传址调用	143
5-3 C++常用函数	146
5-3-1 数学函数	146
5-3-2 随机数函数	151
5-3-3 字符串函数	153
5-3-4 数据转换函数	157
5-3-5 输入输出函数	160
5-4 递归函数 (Recursive Functions)	167
5-5 内联函数	170
温故知新	171
自我突破习题	171
第6章 特殊数据类型与定义	175
6-1 枚举 (enum)	176
6-2 用户自定义类型名称 (typedef)	178
6-3 结构 (struct)	180
6-3-1 结构的定义与存取	180
6-3-2 函数中结构数据的传递	184
6-4 预处理器指令	188
6-4-1 #define 与#undef	188
6-4-2 #include	190
6-4-3 #if...#else...#endif 与#elif	193
6-4-4 #ifdef 与#ifndef	195
6-5 变量的作用范围	196
6-5-1 自动变量 (auto)	196
6-5-2 静态变量 (static)	197
6-5-3 外部变量 (extern)	199
6-5-4 静态外部变量 (static extern)	201
6-5-5 寄存器变量 (register)	202
温故知新	202
自我突破习题	202
第7章 对象与类	205
7-1 面向对象的基本概念	206
7-1-1 什么是对象	206
7-1-2 面向对象设计的优点	208



7-2 加上行为的结构——类	208
7-2-1 类的意义	208
7-2-2 类的设计方法	210
7-2-3 类中成员的种类	214
7-2-4 嵌套类	216
7-3 构造函数与析构函数	217
7-3-1 构造函数的基本概念	217
7-3-2 设定对象的初始值	218
7-3-3 构造函数重载	229
7-3-4 析构函数	230
7-4 对象的声明与操作	230
7-4-1 对象的声明	230
7-4-2 对象的指定	231
7-4-3 动态配置对象	232
温故知新	234
自我突破习题	235
第8章 继承与派生类	239
8-1 派生类	240
8-1-1 继承的概念	240
8-1-2 派生类——公共派生	242
8-1-3 派生类——私有派生	249
8-1-4 派生类——保护派生	250
8-1-5 派生的原则	252
8-1-6 派生类的构造函数与析构函数	253
8-2 多重继承	255
8-2-1 基本概念	255
8-2-2 构造函数的定义	255
8-2-3 成员函数的存取	258
8-3 类的层次	259
8-3-1 单一类层次	259
8-3-2 多重类层次	261
8-4 虚基类	263
8-4-1 使用时机	263
8-4-2 构造函数的存取	265
8-4-3 析构函数的存取	271
8-4-4 虚基类成员的存取	272
温故知新	273
自我突破习题	274

第 9 章 虚函数与友元函数.....	279
9-1 多态	280
9-2 虚函数	281
9-2-1 定义与声明	281
9-2-2 存取关系	284
9-2-3 动态链接	292
9-2-4 纯虚函数	296
9-3 友元函数与友元类	298
9-3-1 友元函数	298
9-3-2 友元类	300
9-4 this 指针	303
温故知新	305
自我突破习题	306
第 10 章 重载	309
10-1 函数重载	310
10-1-1 一般函数重载	310
10-1-2 构造函数重载	311
10-1-3 复制构造函数	315
10-2 运算符重载	319
10-2-1 运算符函数	319
10-2-2 二元运算符的重载 (一)	321
10-2-3 二元运算符的重载 (二)	326
10-2-4 一元运算符的重载	331
10-2-5 等号运算符的重载	334
10-3 类的类型转换	337
10-3-1 利用构造函数转换类型	337
10-3-2 类类型转换成其他类型	339
温故知新	340
自我突破习题	341
第 11 章 模板	347
11-1 什么是模板	348
11-2 函数模板	350
11-2-1 套用一般数据类型的函数模板	350
11-2-2 套用自定义类型的函数模板	353
11-3 类模板	357
11-3-1 定义及使用方法	357
11-3-2 具有固定类型的类模板	359
11-3-3 一个类模板范例——串行队列	362



温故知新	376
自我突破习题	376
第 12 章 输入与输出	379
12-1 流	380
12-2 输出流 cout	382
12-2-1 串联输出	382
12-2-2 字符与字符串输出	382
12-2-3 清除缓冲区 flush()	383
12-2-4 设定输出格式	383
12-3 输入流 cin	389
12-3-1 串联输入	389
12-3-2 单一字符输入 get	389
12-3-3 字符串输入 getline()和 ignore()	391
12-4 文件的输入与输出	393
12-4-1 简单的文件 I/O 处理	394
12-4-2 高级文件处理	397
12-4-3 随机访问文件	409
温故知新	414
自我突破习题	415
附录 A C++关键字	417
附录 B ASCII 码字符对照表	419

chapter 1

C 语言的基本概念

1. 程序语言

2. C 语言的演化

3. Microsoft Visual C++ 6.0 的安装

4. 启动程序与画面简介

5. C 语言程序结构

6. 如何编写 C 语言程序



1-1 程序语言

自从有计算机以来，程序语言就一直存在，它作为人类与计算机沟通的桥梁，默默耕耘了大半个世纪。许多人都赞叹计算机的无所不能与应用的广泛，但是却不知道背后有程序语言这个大功臣，协助程序设计人员实现绚丽的美梦，也成就了无数科技的神话奇迹。

在以前，似乎学习计算机的人，都需要懂一些程序语言，才能把计算机用好。现在，程序语言退居幕后，不需要学习任何计算机程序语言，就可以得心应手地操作计算机，甚至成为专家。说得更清楚些，这种专家是计算机应用专家，在公司的任何部门都可以得到重用；但是，如果身处信息公司或信息相关部门，或者负责公司中的某项信息项目，那么，程序语言就是必修的课程。

现在的高级语言很多，语法和概念大同小异，只要学习过其中一种语言，想要学习另一种语言时，可以有事半功倍的效果。本书将介绍的 C++ 语言，是近代计算机语言较受重视及一直沿用的程序语言，属于中高级语言，但也可以进行低级控制，想要编写应用系统的用户，应该好好学习它。

1-1-1 为什么需要程序语言

观察计算机的进化史可以发现，每一个阶段的硬件发展，其目的都是在改进人类控制计算机的方法，以及应用程序执行的效率。近代美国普林斯顿大学的数学家冯·诺伊曼提出一种新的方法，将控制计算机的程序指令存储在计算机内存中，当计算机要执行时，可从内存中读取指令，执行指令，甚至可以修改程序，使程序设计的方法往前迈进一大步，这就是著名的“存储程序的概念”（stored-program concept）。这个概念对以后的计算机发展影响巨大，使计算机不再只是实验室内的工具，变成每个人都可以应用在日常生活中的必需品。

由于这个程序设计的概念，计算机制造厂商也开始进行系统软件的设计，让硬件在操作系统的指挥下，可以有效率地运行。近代最流行的操作系统，可以说 UNIX 最为普及，其优良的稳定性，可用大量的免费资源，所以从最小的工作站到可以容纳数百位人员同时使用的超级计算机，一般都使用它。另外，在个人计算机方面，以微软（Microsoft）的 MS-DOS 和视窗 Windows 操作系统最为普遍，其所设计的图形用户界面，让一般人可以很容易地操作及设计计算机软件。

说了这么多，到底什么是程序语言呢？简单地说，程序语言是包含一组指令及一套语法规则，使用程序语言所规定的指令，并按照其语法规则组合的，经过编译系统将此语言编译成计算机可以辨认的符号，这种语言称为程序语言。而程序设计人员的目的，就是将设计转换成该程序语言的语法规则，产生计算机程序，来指挥计算机工作。如同学习一般生活上的语言一样，除了有单个字外，还得按照语法来组合单字，才能让人听得懂、看得懂。

为什么要写程序？其实，计算机毕竟不是活的机器，要让计算机能帮我们处理数据，变成有用的信息，必须靠程序设计人员巧妙的思路及创意，把计算机变聪明，甚至变成活的机器。例如：全班学生的成绩登记后，如果只是看看每一个学生的成绩分数，这只能说是一堆数据（data）而已，似乎没有什么特别意义。但是，如果能将成绩予以排序，或者计算出平均分数，以便和其他同学比较，看看自己在全班的排名，看看自己哪一科成绩在班上是比较优秀的，才是有意义的信息（information）。

那么, 我们该选择哪一种程序语言?

在计算机刚发明不久时, 程序设计人员编写程序是用人工方法, 将一个个的指令或程序录入在计算机线路中, 相当费时费力。后来演化成使用代码输入计算机, 称为低级语言, 例如机器语言和汇编语言, 这种方法虽然可以省些时间和精力, 但是人的记忆力是有限的, 低级语言学习起来比较困难, 许多程序的细节都需要清楚地考虑, 一个一个地设计在程序中; 许多复杂的程序, 需要耗费大量人力来输入程序代码, 并且不易调试及维护, 于是高级语言逐渐被开发出来, 用来帮助人们进行计算机软件设计。

1-1-2 高级语言

高级语言的设计理念, 是要仿真人类的语言及逻辑思考的方式, 让程序设计人员可以通过自己熟悉的语言来设计程序, 提高程序的可读性, 以便日后进行调试及维护; 同时, 高级语言所设计的程序有较高的可移植性。在不同的计算机操作系统下, 只要有该计算机系统版本的编译程序, 便可以编译及执行这个程序。因为一般低级计算机程序语言所设计的程序, 其程序指令大部分针对特定系列的微处理器 (microprocessor) 而设计, 所以只能在特定的计算机上运行, 不能转移到其他种类的计算机上编译与执行。例如, Intel 80x86 的汇编程序, 不能在 Macintosh 680x0 计算机上执行。

表 1-1 列出了在高级语言的演化中, 较著名的程序语言:

表 1-1 程序语言的名称及用途

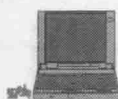
程序语言名称	主 要 用 途
FORTRAN	应用在科学运算、数学、理工等方面
COBOL	应用在数据管理及商业用途
PASCAL	应用在科学计算及教学用途
LISP	人工智能语言
Modula-2、Modula-3	应用在科学计算及一般用途
C、C++	应用在工程及科学用途
BASIC	应用在一般用途及教学用
JAVA	应用于网络及通信用途

虽然高级语言比较容易学习及设计, 但却没有低级语言执行效率高, 因为低级语言可以直接控制计算机最基本的运算方法, 而高级语言所提供的函数库, 主要是提供一般性的使用, 并没有针对某项特别任务而设计, 没有考虑程序的最优化, 也就无法满足要求效率的程序设计; 就如同我们所穿的衣服, 大量制造的成衣穿起来就没有量身订制的衣服舒服。

因此, 设计软件时, 要选择哪种语言设计, 是在系统规划时一项重要的抉择。本书所介绍的程序语言是近来很流行的 C++ 语言, 加上它有面向对象设计的概念, 可以很方便地设计出面向对象程序。至于什么是面向对象? 在本书第 7 章将有详细的介绍。

1-2 C 语言的演化

C 语言是 1972 年由 Dennis Ritchie 和 Ken Thompson 两人在设计 UNIX 操作系统时发展出来的, 由于语法简单, 可以进行高级应用, 也可低级控制计算机的基本功能。由于其功能强大, 因



此成为最近几年来程序设计人员的首选之一。有关 C 语言的版本很多,例如: Turbo C、Borland C、ANSI C 或 Microsoft C 等,除了所附带的函数库有所差异之外,基本的语法都差不多。

比 C 语言还多两个加号的 C++,是由 AT&T 贝尔实验室的 Bjarne Stroustrup 博士设计的,他在 1980 年将面向对象的概念加到 C 语言中,产生了“C with Classes”程序语言。1985 年,Stroustrup 博士在此基础上,加入了更多面向对象的特性,包括运算符重载(operator overloading)、虚函数(virtual function)等,使 C 语言融合了面向对象及传统的程序对象的设计机制,应用面更广,从而完成了 C++语言的设计。于是,各家软件厂商纷纷设计出不同版本的 C++程序语言,如: USL C++(AT&T)、HP C++、Borland C++、Microsoft C++、Visual C++等。本书将使用 Visual C++作为范例设计的语言版本。

1-3 Microsoft Visual C++ 6.0 的安装

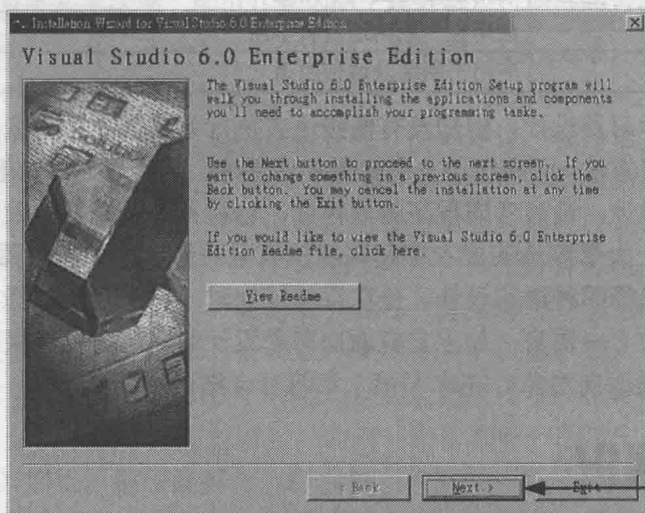
微软公司出版的 Visual C++ 6.0 版是 Visual Studio 6.0 版的成员之一。Visual Studio 6.0 是一套用来开发软件的工具,内容包含五套各有特色的程序语言(Visual Basic、Visual C++、Visual FoxPro、Visual InterDev 和 Visual J++),以及两套辅助的工具(Visual SourceSafe 和 MSDN Library)。因此,安装 Visual C++必须使用 Visual Studio 的第一张光盘来安装。

另外,Visual Studio 6.0 版有两种版本,一个是 Professional 专业版,为计算机专业人员提供了一套功能完整的工具,以便开发解决方案,适合一般 Windows 操作系统或相关软件包的开发者使用;另一个版本是 Enterprise 企业版,用来提供开发大型应用软件,需要管理重要数据的系统,并提供了许多特别的开发与管理工具,适合企业界开发软件使用。

不论您现在手上的 Visual Studio 是哪一种版本,只需要第一张光盘就可以安装 Visual C++,且本书所介绍及使用到的功能,两种版本都会提供。

以下将以企业版为例,安装的步骤如下:

step 1 将 Visual Studio 6.0 版的第一张光盘放入光驱中,系统会自动执行,出现如图 1-1 所示的窗口,直接选择 **Next >** 按钮。



选择 **Next >** 按钮

图 1-1 Visual Studio 6.0 Enterprise Edition 窗口

step 2 出现如图 1-2 所示的 End User License Agreement 窗口后, 选择 I accept the agreement, 再选择 **Next >** 按钮。

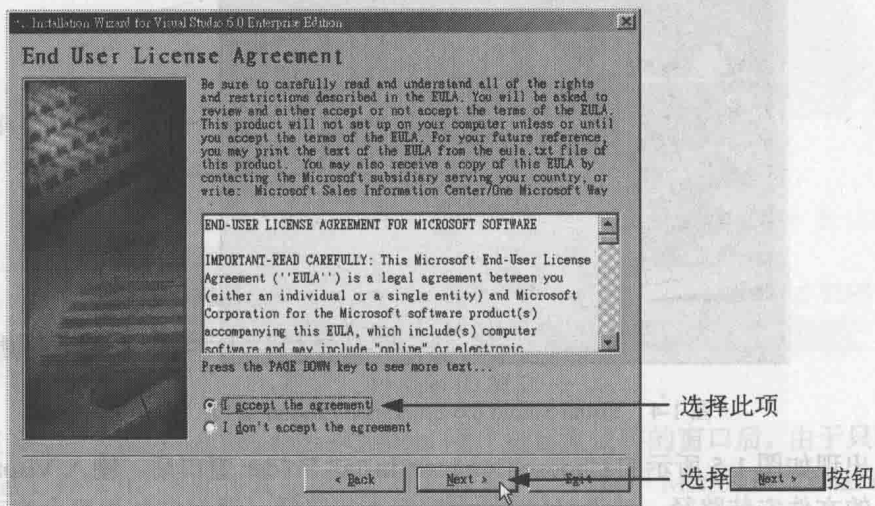


图 1-2 End User License Agreement 窗口

step 3 出现如图 1-3 所示的 Product Number and User ID 窗口后, 输入软件识别码 (如果是 MSDN 中所附带的 Visual Studio 6.0, 则无须输入), 接着输入您的名字及公司名称, 最后选择 **Next >** 按钮。

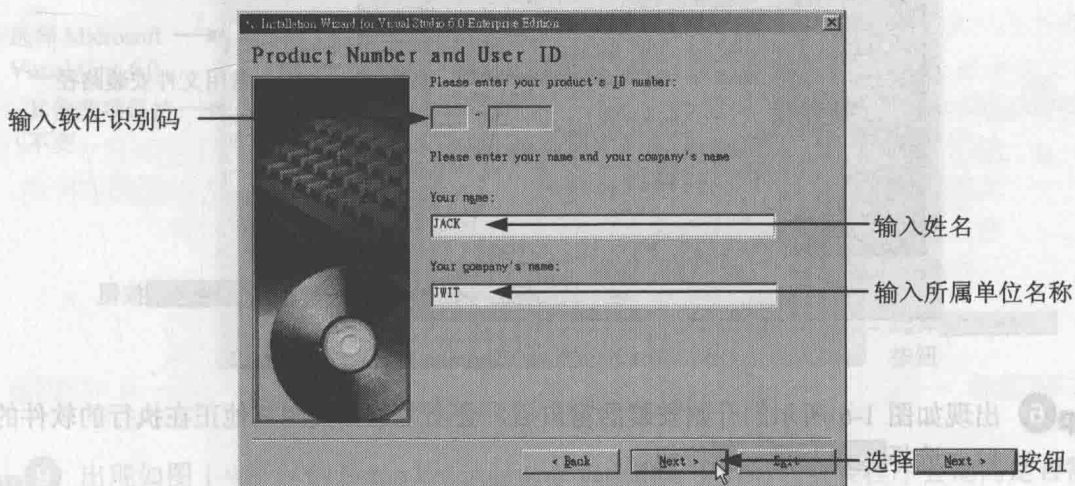


图 1-3 Product Number and User ID 窗口

step 4 出现如图 1-4 所示的 Visual Studio 6.0 Enterprise Edition 窗口后, 因为我们只要安装其中的 Visual C++, 所以选择 Enterprise Setup Options 栏中的 Custom 选项, 再选择 **Next >** 按钮即可。

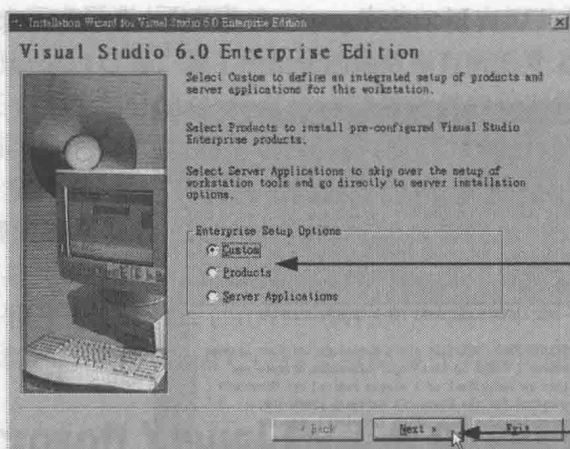


图 1-4 Visual Studio 6.0 Enterprise Edition 窗口

step 5 出现如图 1-5 所示的 Choose Common Install Folder 窗口后，输入 Visual Studio 常用的文件安装路径，或选择 **Browse...** 按钮查找路径（通常安装程序会自动显示默认的路径，不需要修改），接着选择 **Next >** 按钮。

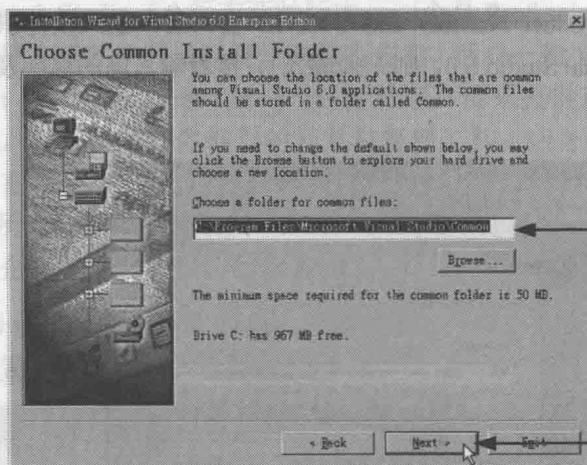


图 1-5 Chose Common Install Folder 窗口

step 6 出现如图 1-6 所示的开始安装的窗口后，会出现请你关闭其他正在执行的软件的窗口，选择 **Continue** 按钮。

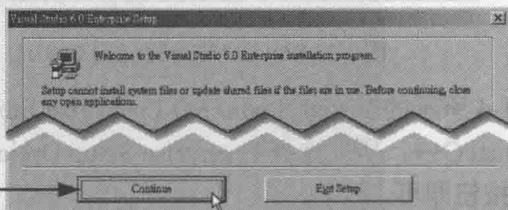


图 1-6 开始安装窗口

step 7 出现如图 1-7 所示的 Product ID 窗口后，选择 **OK** 按钮，开始搜寻已经安装的组件。

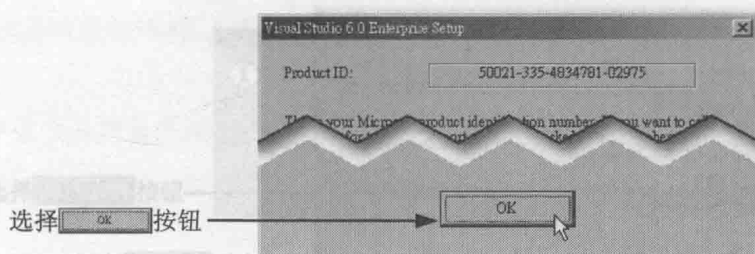


图 1-7 Product ID 窗口

step 8 出现 Visual Studio 6.0 Enterprise-Custom 用户自定义安装的窗口后, 由于只需要安装 Visual C++ 6.0, 因此将其他项目前面的 ☐ 取消, 只留下 Microsoft Visual C++6.0、Active X、Data Access 即可, 然后选择 **Continue** 按钮, 如图 1-8 所示。

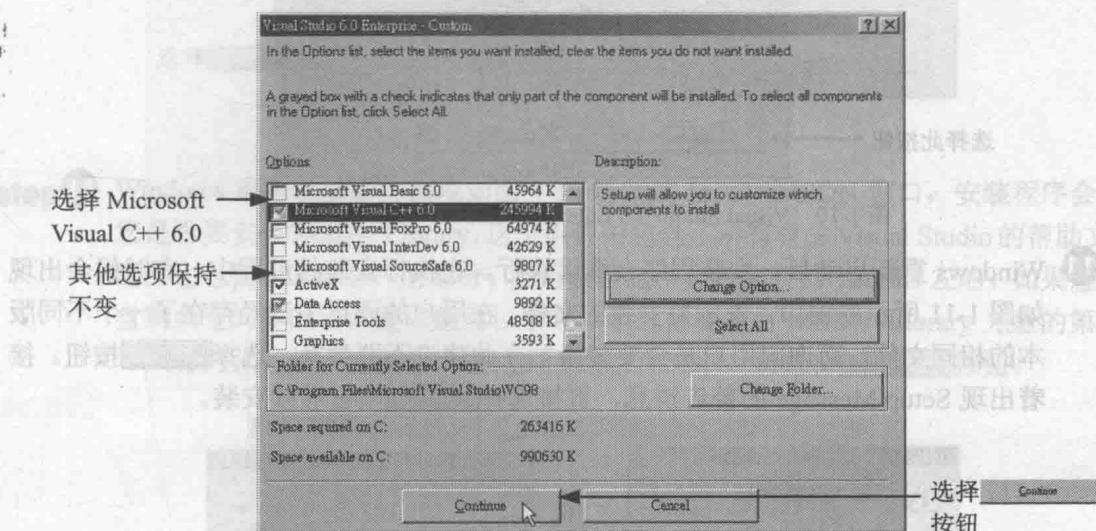
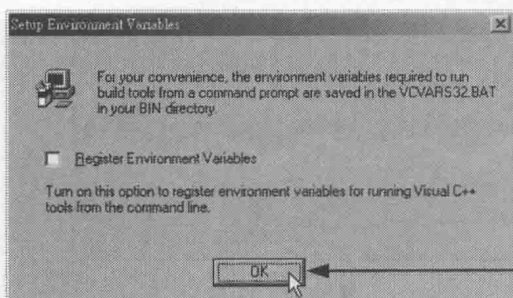
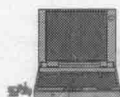


图 1-8 Visual Studio 6.0 Enterprise-Custom 窗口

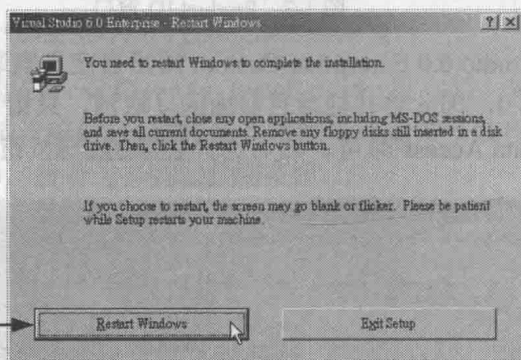
step 9 出现如图 1-9 所示的 Setup Environment Variables 窗口后, 安装程序会询问是否需要保存环境变量设置的批处理文件 VCVARS32.BAT, 以便将来在 DOS 命令行的环境下执行此批处理文件。由于在 DOS 环境下执行的机会不大, 因此不必选择, 直接选择 **OK** 按钮即可。



选择 **OK** 按钮

图 1-9 Setup Environment Variables 窗口

step 10 出现如图 1-10 所示的 Visual Studio 6.0 Enterprise - Restart Windows 窗口后，选择 **Restart Windows** 按钮，系统会自动重新启动 Windows。



选择此按钮

图 1-10 Visual Studio 6.0 Enterprise-Restart Windows 窗口

step 11 Windows 重新启动后，安装程序会继续执行。在执行安装的过程中，有时候会出现如图 1-11 所示的窗口，表示要安装的文件，在用户的硬盘中已经存在了一个不同版本的相同文件，而询问用户是否要替换？在此建议不要替换，选择 **No** 按钮。接着出现 Setup Message 的警告消息，直接选择 **Yes** 按钮继续安装。

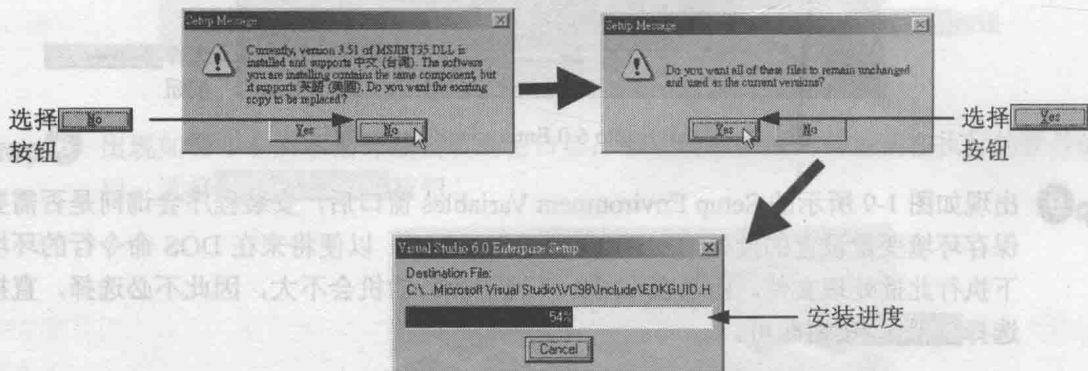


图 1-11 Setup Message 窗口

step 12 出现如图 1-12 所示的 Windows NT Debug Symbols 窗口后，表示系统将会在 Visual C++ 6.0 的程序组中，安装一个 Windows NT system symbols 的安装程序，将来如果需要的话，可以直接执行此程序来安装。直接按下 **确定** 按钮即可。