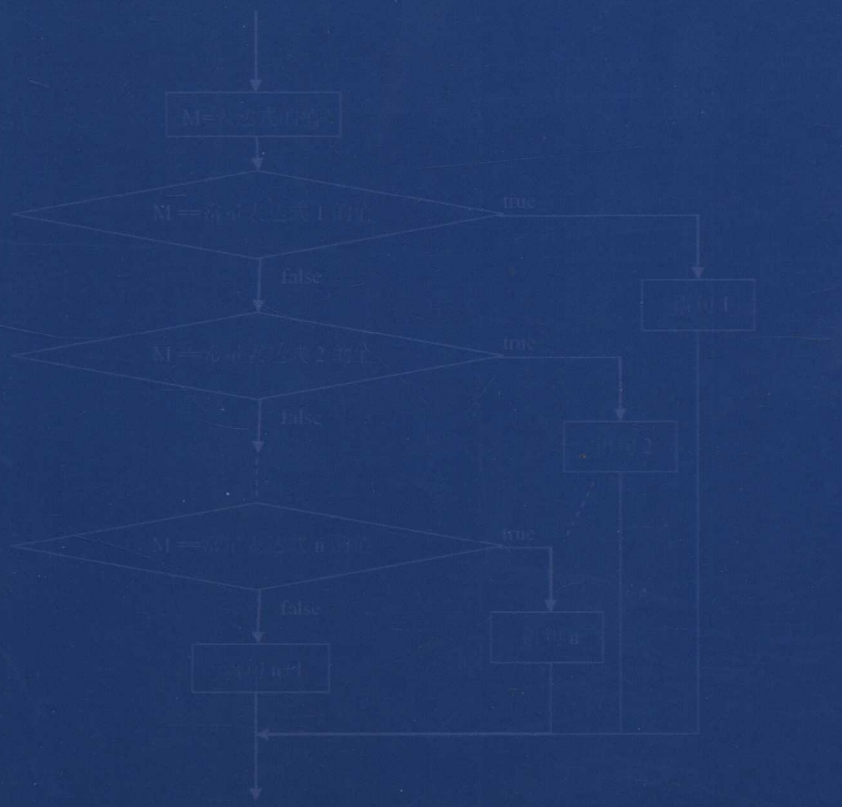
张广庆 杜春涛
付瑞平 张师林 /编著

C++程序设计基础

适用于非计算机专业读者

便于较短时间内掌握C++语言



基本数据类型

控制结构

数组

指针与引用

函数

程序结构

类与对象

继承与派生

多态性

输入输出流

异常处理与MFC基础

中国社会出版社

C++程序设计基础

张广庆　杜春涛
付瑞平　张师林　编著

中国社会出版社

本书参照国家教育部考试中心制定的2004年版《全国计算机等级考试大纲》中关于C++语言的要求编写，主要包括：基本数据类型、控制结构、数组、指针与引用、函数、程序结构、类与对象、继承与派生、多态性、输入输出流和异常处理与MFC基础等内容。

本书内容精炼，结构合理，内容涵盖了C++语言的主要部分，每章后配有较为丰富的练习题，便于读者自学。

图书在版编目(CIP)数据

C++程序设计基础 / 张广庆等编著. —北京：中国社会出版社，2006.1

ISBN 7-5087-0909-8

Ⅰ.C… Ⅱ.张… Ⅲ.C语言—程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字(2005)第153900号

书　　名：C++程序设计基础

编 著 者：张广庆　杜春涛　付瑞平　张师林

责任编辑：尤永弘

出版发行：中国社会出版社　　邮政编码：100032

通联方法：北京市西城区二龙路甲33号新龙大厦

电话：66051698　　电传：66051713　　邮购电话：66060275

经　　销：各地新华书店

印刷装订：北京京海印刷厂

开　　本：787mm×1092mm　1/16

印　　张：23.75

字　　数：598千字

版　　次：2006年1月第1版

印　　次：2006年1月第1次印刷

书　　号：ISBN 7-5087-0909-8/TP·7

定　　价：39.80元

北方工业大学校编教材编委会

前　言

本书根据国家教育部考试中心制定的2004年版《全国计算机等级考试大纲》中关于C++语言的要求和编者多年来C++语言教学经验编写而成。

本书根据非计算机专业对计算机语言需求特点，对C++语言的内容进行了有针对性地取舍，保留了基本数据类型、控制结构、数组、指针与引用、函数、程序结构、类与对象、继承与派生、多态性、输入输出流和异常处理与MFC基础等内容，略去了位运算、结构体、共用体、链表等内容，以便读者能够在较短的时间内较为容易地掌握C++语言的主要内容。全书共分13章。第1章为语言概述，第2章至第8章为C++语言的基础部分，例题程序采用面向过程的方法编写。第9章至第11章介绍了C++语言三大特性，即封装、继承与派生和多态性。例题程序采用面向对象的程序设计方法编写。第12章、第13章分别介绍了输入输出流和异常处理与MFC基础，以便读者进一步学习可视化设计。书中所有例题程序均在Microsoft Visual C++ 6.0集成开发环境中调试通过。

本书第1章、第9章、第10章、第11章由张广庆编写；第5章、第6章、第7章、第8章由杜春涛编写；第2章、第3章、第4章由付瑞平编写；第11章第4节、第12章、第13章由张师林编写。

本书内容精炼，结构合理，内容涵盖了C++语言的主要部分，书中配有大量例题和丰富的练习题，便于读者自学。

由于编者水平所限，书中难免有疏漏谬误，敬请读者提出宝贵意见。

目 录

第1章 绪 论

本章要点

本章简要介绍计算机语言的基本概念，C++语言的发展历史，C++语言的词法规则，程序设计结构的特点和程序的实现方法以及使用 Visual C++ 6.0 对 C++程序进行编辑编译的简单方法。

1.1 计算机语言

一个计算机系统不仅需要强大的硬件系统，而且需要强有力的软件系统支持。软件系统中除了完成特定功能所需的数据之外，主要是使计算机运行的各种指令的集合（程序）。一般来说，我们称编制计算机程序所使用的字符，词法和语法规则系统为**计算机语言**。目前所使用的计算机程序语言可以分为机器语言，汇编语言和高级语言。

机器语言：由计算机系统可以识别的二进制指令，即简单的“0”和“1”组成的语言。计算机可以很好地识别机器语言，但对程序设计者来说，机器语言晦涩难懂，难于记忆。

汇编语言是将机器指令映射为一些可以被人读懂的助计算机符号，如 ADD,SUB 等。可以直接对硬件物理结构进行操作，以前的操作系统等系统软件主要是用汇编语言编写的。汇编语言的语法规则因所操作的硬件的具体结构的不同而有所差异，可移植性和可读性较差，主要为专业人员掌握和使用。

高级语言大都采用接近英语的语法规则编写程序，易学且可读性和可移植性好，适合所有人员掌握使用。高级语言从 20 世纪 50 年代至今已开发了几百种。目前主要的高级语言有 BASIC、FORTTRAN、C/C++等。

汇编语言和高级语言所编写的程序一般称为**源程序**或**源代码**，而机器语言写成的程序一般称为**机器代码**或**目标代码**。计算机系统只能识别机器代码，因此，源程序必须被翻译成机器代码。将高级语言编写的源程序翻译成机器代码的软件称为**编译程序**，而将汇编语言编写的源程序翻译成机器代码的软件称为**汇编程序**。

机器语言和汇编语言又称为低级语言，他们都是面向机器的语言，即用之编写的程序只适用于特定类型的计算机。

高级语言按其程序设计思想又分为面向过程的语言，如 BASIC、FORTTRAN、PASCAL、C 等；和面向对象的语言，如 C++。

高级语言按其翻译成机器代码和执行过程的不同，又可分为**解释性语言**和**编译性语言**，BASIC 是解释性语言，其他为编译性语言。解释性语言采用逐句解释直接执行的方式运作，易学，适合业余人员编程，但运行速度很慢，不适宜编写大型应用软件。编译性语言采用集中编译连接形成**可执行文件**（.EXE 文件）,然后再执行的方式运作，速度极快，但对编程人员素质要求较高，学习有一定难度。

我们现在所使用的应用软件基本上是用高级语言编写的。对于理工科专业 C/C++语言是必须掌握的。C/C++语言不仅具有高级语言的一切优点，还拥有汇编语言的大部分功能，可以直接操作硬件物理结构。因此，当今主要的操作系统软件如著名的 Unix 和

Windows 等均使用 C/C++语言编写，而应用软件更是首选 C/C++语言。尽管现在已有一些新的语言如 JAVA、UML 等出现，并显示出更好的性能，但在今后相当长时期内，C/C++语言仍将是主要编程语言。

1.2 C++的由来

C++语言是由 C 语言发展而来的。1972 年，贝尔实验室的 D.M.Ritchie 开发了 C 语言,但直到 1975 年 C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的可移植 C 语言编译程序，使 C 语言移植到其他机器时所需工作大大简化，C 语言得到了广泛的应用。1983 年，美国国家标准化协会（ANSI）制定了 ANSI C 标准。1987 年又公布了 87 ANSI C。成为目前 C 语言的共同基础。C 语言与早期高级语言相比有以下特点：

（1）语言简洁、紧凑、使用方便、灵活性好。

（2）运算符丰富，共有 34 种运算符可以实现在其他高级语言中难以实现的运算。

（3）数据结构丰富，具有现代化语言的各种数据结构。C 的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据结构（如链表、树、栈等）的运算。尤其是指针类型数据，使用方式比 PASCAL 更为灵活，多样。

（4）具有结构化的控制语句（如 if…else 语句、while 语句、do…while 语句、switch 语句、for 语句）。用函数作为程序模块以实现程序的模块化，是结构化的理想语言。

（5）语法限制不太严格，程序设计自由度大。例如，对数组下标越界不作检查；对变量的类型使用比较灵活，例如，整型量与字符型以及逻辑型数据可以通用。

（6）C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 既具有高级语言的功能，又具有低级语言的功能，C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。有人把 C 称为“高级语言中的低级语言”，也有人称它为“中级语言“。

（7）生成目标代码质量高，程序执行效率高，比汇编程序生成的目标代码效率低 10-20%。

（8）用 C 语言编写的程序可移植性好，相比于汇编语言，基本不作修改就能用于各种操作系统。

和其他早期高级语言一样，C 语言是所谓“面向过程”的语言，这限制了它应用范围，C 语言的局限性主要表现在：

（1）检查机制较弱，使得一些错误不能在编译时发现。

（2）不具备代码重用的语言结构，使得新开发的程序很难利用已开发的程序资源。

（3）当开发较大规模的程序时，程序员难于驾驭其复杂性。

由于 C 语言的上述局限性，C++语言的开发就成为必然。C++语言是以 C 语言为基础而开发的一种支持面向对象编程方法的程序设计语言，它的开发者是 AT&T 贝尔实验室的 Bjarne Stroustrup 博士。C++最初被称为“带类的 C”，1983 年取名为 C++，以后经过不断完善和发展，在 C++4.0 的基础上，1997 年正式发布了 ANSI C++标准。

C++将 C 的几乎全部内容作为它的子集，并对 C 的类型系统进行了改进和扩充，因此，C++比 C 更安全，C++编译器可以检查更多的类型错误。

C++语言开发的另一个主要目标是支持面向对象的程序设计。面向对象的程序设计是一种更好的结构化程序的技术。C++语言引入了类，继承，多态等机制来支持“面向

对象”的程序设计，使 C++语言对更高级的抽象有更好地支持，这种支持使得在计算机上解决问题的方式更类似于人类的思维活动。

C++保持与 C 兼容，这就使诸多 C 代码不经修改就可以为 C++所用，C 的库函数和软件基本上都可以用于 C++环境。更重要的是，C 程序员仅需学习 C++的新特征就可以编写 C++程序。另外，用 C++编写的程序可读性更好，代码结构更合理，可以更直接映射问题空间的结构。

1.3 C++的词法及词法规则

像任何人类语言一样，计算机语言也是由语言的基本单位，即符号（字符集）和单词组成的。

1.3.1 字符集

字符是程序中的一些可以区分的最小符号。C++语言的字符集由下述一些字符构成：

大小写英文字母：A～Z a～Z。

数字字符：0～9。

特殊字符：

空格	!	#	%	^	&	_（下划线）	
+	-	=	~	<	>	/	\
‘	“	;	,	.	()	[]	{}

1.3.2 单词

计算机语言中的**单词**是由若干个字符组成的具有一定意义的最小语法单元。C++共有六种单词：它们是关键字、标识符、常量、操作符（或运算符），标点符号（分隔符），注释符。

1. 关键字

关键字是 C++预定义的单词，它们在程序中有不同的使用目的。关键字是 C++语言保留成分，不允许被重新定义。表 1.1 中给出了 C++全部的关键字。

表 1.1 C++的关键字

auto	bool	break	case	catch	char	clase
const	const_cast	continue	default	delete	do	double
else	dynamic_cast	enum	explicit	extern	false	float
for	friend	goto	if	inline	int	long
new	operator	mutable	private	projected	public	register
return	reinterpret	short	signed	sizeof	static	static_cast
struct	switch	template	this	throw	true	try
typedef	typeid	typename	union	unsigned	virtual	void
volatile	while					

2. 标识符

标识符是程序员定义的单词，它代表程序中的一些实体，以便在程序语句中使用这些实体。常见的标识符有函数名，变量名，常量名，标号名，类型名，对象名，类名等。

在 C++中，标识符由大小写的字母，0 到 9 数字和下划线组成。组成规则是：以字母或下划线开始，其后跟零个或多个字母、数字或下划线。例如：

Result DoubleList reck_of _race1

注意：

（1）标识符的长度（组成标识符的字符个数）是任意的，但特定的编译系统能够识别的标识符的长度是有限的，如 Borlan C++编译器只能识别标识符的前 32 个字符，超过者被忽略。

（2）C++中，字母的大小写是有区别的。例如：

Add　　add　　ADD

是不同的标识符。

（3）在实际应用中，尽量使用有意义的单词作标识符，不要通过字符的大小写形式来区分不同的标识符，以免影响可读性。

（4）定义标识符时，不要采用系统的关键字。如：auto，cout。

（5）定义标识符时，习惯上，将单词的第一个字符大写。

3．常量

常量是在程序中直接使用的以符号表示的数据，包括数字常量、字符常量、字符串常量和布尔（Bool）常量。在第二章中，我们将详细讨论 C++的各种常量。

4．操作符（运算符）

程序设计中的**操作符**表示对操作数进行特定的运算，并得到一个结果值。操作符通常由一到两个字符组成：例如，“+”表示加操作，“==”表示恒等判断。

操作符又被分类为一元（单目）操作符和二元（双目）操作符。C++还有一个三元（三目）操作符。我们将在以后的课程中详细讨论它门的功能。

5．分隔符（标点符号）

分隔符用于分隔各个单词或程序正文，表示一个程序实体的结束和另一个程序实体的开始。它们不表示任何实际的操作，仅用于构造程序。C++的分隔符是：

（1）**空格符**：常用来作为单词与单词之间的分隔符。

（2）**逗号**“,”：用来作为说明多个变量或对象类型时，变量之间的分隔符；或者用来作为函数的多个变量之间的分隔符。逗号还可以用作运算符。

（3）**分号**“;”：仅用作 for 循环语句中 for 关键词后面括号中三个表达式的分隔符。

（4）**冒号**“:”：用来作语句标号与语句之间的分隔符和 switch 语句中关键字 case<整常型表达式>与语句序列 之间的分隔符。

（5）**圆括号**“()”：主要用于函数头的参数表达。

（6）**花括号**“{}”：构成函数体。

6．注释符

注释符用来引出对程序的注解和说明，注释的目的是为了便于阅读程序。在程序编译的词法分析阶段，注释将被从程序中删除。C++中采用两种注释符。

（1）“/*”，和“*/” 在“/*”，和“*/”之间的所有字符都被作为注释处理。这种方法适用于有多行注释信息的情况。如：

```
/*This program writes two messages to the screen，say anything at all。But
make the messages on two separate lines。*/
```

（2）“//“ 从“//“开始，直到它所在的行尾，所有字符都被作为注释处理。这种方法用于单行注释信息。如：

```
// This is a comment。
```

7．空白符

空白符是**空格**，**换行符**，**制表符**和**注释**的总称。程序正文中的空格、制表符（TAB 键产生的字符）、换行符（Enter 键产生的字符）和注释在程序编译时的词法分析阶段将

变为空白。空白用于表示词法记号的开始和结束位置，但除了这一功能之外，其余的空白将被忽略。例如：

```
int i;
int     i;
int
i
;
```

三个语句是等价的。

1.4 C++程序结构的特点

下面我们通过一个具体程序介绍C++程序结构的特点。

例 1-1

程序功能是：从键盘输入两个浮点数，求其和并在屏幕上输出显示。源代码

```
//This is a C++   program.
#include <iostream>
using namespace std;
int main()
{
    double x, y;
    cout<<"Enter two float number:";
    cin>>x>>y;
    double z = x + y;
    cout<<"x + y="<<z<<endl;
    return 0;
}
```

执行该程序，显示如下信息：

```
Enter two float number: 7.2 9.3
```

输入两个浮点数，用空格符作分隔，按回车键，输出如下结果：

```
 x + y=16.5
```

结合上述程序，可以看出C++程序有如下基本组成部分：

1．预处理命令（**预编译指令**）

以“#”开头的命令称为**预处理命令**（**预编译指令**）。程序中：

```
#include <iostream >
```

指示编译器在对程序进行预处理时，将文件 iostream 中的代码嵌入到程序中该指令所在的地方。其中 include 是关键字，预处理命令#include 称为文件包含命令，iostream 是被包含的文件名。该文件声明了程序中出现的输入 cin>> 和输出 cout<<操作的有关信息。通过嵌入该文件，程序就可以直接使用 C++系统的输入输出操作。一般称 iostream 为头文件。预处理命令还有其他的命令，它们都是以#符号开始。需要注意的是预处理命令不属于 C++语句内容。

2．命名空间

程序中

```
using namespace std;
```

语句的含义是使用**命名空间** std。对于初学者只需记住在所有的文件包含命令后都必须有此语句即可。在以后的深入学习时，参考有关文献即可理解。

3．输入和输出

cin 代表一个**标准输入设备**（一般代表键盘设备）。操作符“>>”表示从输入设备上接收数据来更新“>>”右边的对象。cout 代表一个**标准输出设备**（一般代表屏幕）。操作符“<<”表示将“<<”右边的数据输出到“<<”左边的输出设备上。下列语句的含义见语句右边的注释。

```
cout<<"Enter two float number:"; //输出字符串"Enter two float number:"到屏
                                 //作为提示信息
cin>>x>>y;                       //从键盘输入两个浮点数，赋给变量 x，y
cout<<"x + y="<<z<<endl;         //输出字符串"x + y="和变量 z 的值到屏幕
```

4．函数

程序中

```
int  main()
{
    ...
    return 0;
}
```

“int main()”称为函数头，花括号{}括起的部分称为函数体，函数头和函数体构成了一个完整的**函数定义**。

对于初学者只需记住函数名 main 前必须写 int 并用空格与 main 分隔，同时在 main 函数体的右花括号前写上语句：

```
return 0;
```

至于它们的具体含义，我们将在函数一章内讲述。

一个 C++程序至少要有一个函数，在组成一个程序的若干函数中，必须有一个而且只能有一个主函数 main()。一般来说，C++程序的运行从主函数开始，也结束于主函数，其他函数只能通过主函数或被主函数调用的函数进行调用而被执行。

5．语句

函数是由若干条**语句**组成的。没有语句的函数称为**空函数**。语句是由单词组成的，单词之间用空格符分隔，每条语句以分号结尾，语句是程序的基本单元。示范程序的**函数**中共有 6 条语句。

6．变量

变量在使用之前必须声明（定义）。示范程序中：

```
double x, y;
```

声明了两个 double 型变量 x, y。C++中声明变量，意味着给变量分配内存空间。

7．注释

注释是程序员为读者所作的说明，是提高程序可读性的一种手段。注释分为两种：序言注释和注解性注释。前者用于程序开头，说明程序或文件的名称，用途等有关信息，后者用于程序难懂的地方。

```
//This is a C++  program.    //序言注释
```

注意书写 C++程序时的基本原则：

（1）一行写一条语句。短语句可以一行写多个语句。长句可以一条写多行。分行的

原则是不能将一个单词分开。用双引号引用的一个字符串最好不要分开，如果一定要分开，应按照具体编译系统的规定进行。

（2）采用适当的缩格书写方式以增加可读性。如：同一内容的语句行要对齐；同一循环体内的个语句要对齐。

（3）建议花括号各占一行，并与使用花括号的语句对齐。花括号内的语句采用缩格书写方式，一般缩进两个字符。

我们看一个例子。

```
#include<iosteam>
using namespace std;
void
main()
{
    int a,
     b;
    a=5;b
    =7;cout
    <<"a*b="
    <<a*
    b<<endl;
}
```

上面程序的可读性非常差。

按书写基本原则修改后，可读性大大改善了。

```
#include<iosteam>
using namespace std;
void main()
{
    int a, b;
    a=5;
    b=7;
    cout<<"a*b="<<a*b<<endl;
}
```

1.5 C++程序的实现过程

C++程序的实现一般经过编辑、编译、连接和运行四个阶段。

1.5.1 编辑

编辑是将编写好的 C++源程序，通过特定软件所提供的编辑器（如：Visual C++，Borland C++，Turbo C++等），录入到扩展名为.CPP 的磁盘文件中的过程。

1.5.2 编译

编译是通过特定软件所提供的编译程序（编译器）将 C++源程序代码转换成目标代码。编译过程分成两个子过程：

1．预处理过程

如果源程序中有预处理命令，则先执行预处理命令。没有预处理命令，就执行下面

的编译过程。

2. 编译过程

编译过程主要是对源程序进行词法分析和语法分析。在分析过程中如发现有词法和语法错误，编译程序会及时显示在屏幕上报告给用户。如无错误，则形成以.OBJ 为扩展名的**目标代码文件**。

1.5.3 连接

连接过程是对编译过程所生成的目标代码进行连接，最后生成可执行文件的过程。

一般程序都有多个源文件，编译器对每个源文件分别进行编译，从而形成了多个目标代码文件，需要连接起来才能运行。此外源文件生成的目标代码文件还需要系统提供的**库文件**（扩展名.LIB）中的一些代码。

连接工作由编译系统中的连接程序（又称连接器）来完成。连接器对目标代码文件和库中的某些文件代码进行连接处理，生成扩展名为.EXE 的**可执行文件**。

1.5.4 运行

一般的编译系统都有运行功能，通过选择菜单项，即可实现。也可在 MS—DOS 系统下，在 DOS 提示符后，直接输入可执行文件名，并按回车即可执行。上述过程一般要经过反复多次，才能完成。

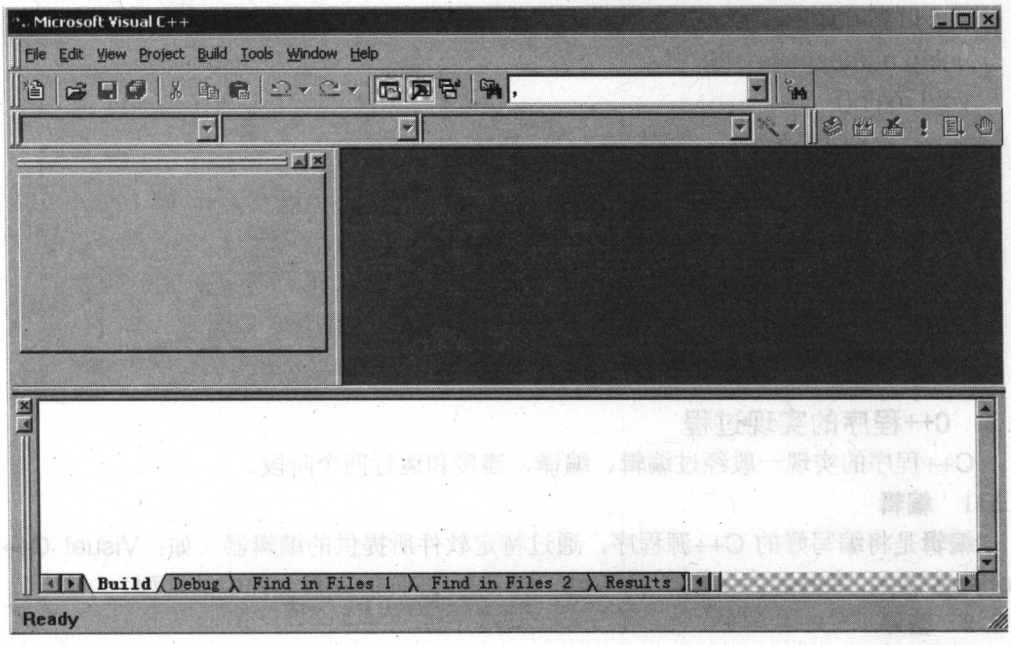
1.6 Visual C++ 6.0 基本用法

本书采用 Visual C++ 6.0 作为编辑、编译和运行平台。下面对这一系统进行介绍。

1.6.1 编辑 C++源程序

1. 启动 Visual C++ 6.0

单击 Visual C++ 6.0 图标，出现“Microsoft Developer Studio”窗口。如下图。



2. 进入编辑窗口

按如下步骤操作：File→New→File→C++ Source File→OK 出现编辑屏幕，依次见下图。