

全国计算机等级考试三级教程

C 语言程序设计

(Visual C++6.0 环境)

高峥虹 李浩岩 主编



化学工业出版社

· 北 京 ·

本书是编者依据多年来对计算机等级考试的研究,以及丰富的教学经验,针对三级 C++语言上机考试的题库编写的。具体内容包括 C++语言概述, C++语言的数据类型,运算符和表达式, C++语言程序设计初步,分支程序设计,循环控制,数组,函数,预处理命令,指针,结构体,文件共 12 章。

本书所涉及的题大多来源于三级考试上机题库,把考试的考点和 C++语言知识点紧密地结合在一起,具有完整的 C++语言知识体系,同时针对性很强,非常实用。

本书适用于各计算机等级考试机构作为教材使用,也可供计算机考生自学使用。

图书在版编目(CIP)数据

C 语言程序设计: Visual C++6.0 环境 / 高峥虹, 李浩
岩主编. —北京: 化学工业出版社, 2008.2
全国计算机等级考试三级教程
ISBN 978-7-122-02117-5

I. C… II. ①高…②李… III. C 语言-程序设计-水平考试-自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 017020 号

责任编辑: 张建茹

文字编辑: 谢蓉蓉

责任校对: 蒋 宇

装帧设计: 关 飞

出版发行: 化学工业出版社(北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 北京市彩桥印刷有限责任公司

787mm×1092mm 1/16 印张 10¼ 字数 265 千字 2008 年 3 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888(传真: 010-64519686) 售后服务: 010-64519661

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价: 19.00 元

版权所有 违者必究

前 言

随着中国计算机应用的进一步普及和深入，人们已经达成了一个共识：计算机知识是当代人类文化的重要组成部分；计算机应用能力是跨世纪人才不可缺少的素质。全国计算机等级考试（National Computer Rank Examination, NCRE）是由教育部考试中心主办，用于考查应试人员的计算机应用知识与能力的考试。本考试的证书已经成为许多单位考核、录用工作人员的重要条件。

全国计算机等级考试从 1994 年开考以来，三级上机题库一共分为两类，一类是 C 语言题库（2008 年上机环境改为 Visual C++6.0），另一类是汇编语言题库。三级一共分四个科目，有三级网络技术、三级信息管理、三级数据库、三级 PC 技术。其中只有三级 PC 技术上机考汇编语言，其他科目均考 C 语言，而且这三科使用的是同一个题库。

本书紧扣考试大纲，内容取舍得当，是一本系统的考试辅导教材。

本书编者依据多年来对计算机等级考试的研究，以及丰富的教学经验，编写这本三级 C++ 语言教材。本书的特色在于，针对三级 C++ 语言上机考试的题库，全面讲解考试中用到的 C++ 语言知识。书中所涉及的题大多来源于三级考试上机题库。这样就可以把考试的考点和 C++ 语言知识点紧密的结合在一起，形成一本具有完整知识体系又针对于计算机三级上机考试的 C++ 语言教材。在本书的帮助下，读者将会达到事半功倍的效果，顺利通过考试。

本书适合各计算机等级考试机构作为教材使用，也可作为考生自学使用，同时是一本学习 C 语言的好教材。本书各章课后的习题答案可以在本书的配套教材《全国计算机等级考试三级教程——C 语言上机真题题库与解析（Visual C++6.0 环境）》中找到，随此书赠送的光盘是《计算机等级考试模拟软件》的试用版，光盘中的上机环境同真实考试环境一样，并配有精确的评分系统和详细的解析。读者可以联系 Email: zcre2007@163.com；网址：WWW.zcre.cn 购买正版软件，以获得完整题库。

本书由高峥虹、李浩岩担任主编。其中第 1、10、11 章由高峥虹编写，第 2、3 章由张国伟编写，第 4 章由赵峰编写，第 5 章由李鹏飞编写，第 6、7 章由李浩岩编写，第 8 章由阿庆兴编写，第 9 章由孙靓编写。由高峥虹对全书进行全面审阅。

由于编写时间仓促，难免有疏漏之处，请广大读者批评指正。

编者

2008 年 1 月

目 录

第 1 章 C++语言概述	1
1.1 C++语言概述	1
1.1.1 C++语言的发展过程	1
1.1.2 C++语言的特点	1
1.2 简单的 C++程序介绍	2
1.3 C++的基本语法	3
1.3.1 C++源程序的结构特点	3
1.3.2 书写程序时应遵循的规则	3
1.3.3 C++语言的字符集	4
1.3.4 C++语言词汇	4
1.4 Visual C++ 6.0 开发环境及 C++上机过程	5
1.4.1 Visual C++6.0 简介	5
1.4.2 操作步骤	6
1.4.3 创建 C++源程序文件	7
第 2 章 C++语言的数据类型	11
2.1 数据类型	11
2.1.1 基本数据类型	11
2.1.2 构造数据类型	11
2.1.3 指针类型	11
2.1.4 空类型	11
2.2 常量与变量	12
2.2.1 常量和符号常量	12
2.2.2 变量	13
2.3 整型数据	13
2.3.1 整型常量	13
2.3.2 整型变量	13
2.4 实型数据	14
2.4.1 实型常量	14
2.4.2 实型变量的分类	15
2.5 字符型数据	15
2.5.1 字符常量	15
2.5.2 转义字符	16

2.5.3	字符变量	16
2.5.4	字符串常量	17
2.6	变量赋初值和类型转换	18
2.6.1	变量赋初值	18
2.6.2	变量类型的转换	18
第3章	运算符和表达式	20
3.1	运算符简介	20
3.1.1	运算符的分类	20
3.1.2	运算符的优先级和结合性	21
3.2	算术运算符	21
3.2.1	算术运算符的分类	21
3.2.2	算术运算符的应用	21
3.3	赋值运算符和赋值表达式	21
3.4	类型转换运算符	22
3.5	复合的赋值运算符	23
3.6	自增、自减运算符	23
3.7	位运算符	24
3.7.1	左移运算	24
3.7.2	右移运算	25
	习题	25
第4章	C++语言程序设计初步	27
4.1	C++语句概述	27
4.2	赋值语句	28
4.3	C++语言的输入输出	29
4.3.1	putchar 函数（字符输出函数）和 getchar 函数（键盘输入函数）	29
4.3.2	格式输入与输出	30
4.4	顺序结构程序设计举例	37
	习题	38
第5章	分支程序设计	39
5.1	关系运算符和表达式	39
5.1.1	关系运算符及其优先次序	39
5.1.2	关系表达式	39
5.2	逻辑运算符和表达式	40
5.2.1	逻辑运算符及其优先次序	40
5.2.2	逻辑运算的值	40
5.2.3	逻辑表达式	41
5.3	if 语句	42

5.3.1	if 语句的三种形式	42
5.3.2	在使用 if 语句中还应注意的问题	45
5.4	if 语句的嵌套	47
5.5	条件运算符和条件表达式	48
习题		49
第 6 章	循环控制	50
6.1	概述	50
6.2	while 语句	50
6.3	do-while 语句	51
6.4	for 语句	53
6.5	循环的嵌套	57
6.6	break 和 continue 语句	59
6.6.1	break 语句	59
6.6.2	continue 语句	60
习题		61
第 7 章	数组	62
7.1	一维数组的定义和引用	62
7.1.1	一维数组的定义方式	62
7.1.2	一维数组元素的引用	63
7.1.3	一维数组的初始化	64
7.1.4	一维数组程序举例	65
7.2	二维数组的定义和引用	67
7.2.1	二维数组的定义	67
7.2.2	二维数组元素的引用	68
7.2.3	二维数组的初始化	68
7.2.4	二维数组程序举例	69
7.3	字符数组	70
7.3.1	字符数组的定义	70
7.3.2	字符数组的初始化	70
7.3.3	字符数组的引用	71
7.3.4	字符串和字符串结束标志	71
7.3.5	字符数组的输入输出	72
7.3.6	字符串处理函数	73
习题		79
第 8 章	函数	81
8.1	概述	81
8.2	函数定义的一般形式	82

8.3 函数的参数和函数的值	84
8.3.1 形式参数和实际参数	84
8.3.2 函数的返回值	85
8.4 函数的调用	86
8.4.1 函数调用的一般形式	86
8.4.2 函数调用的方式	86
8.4.3 被调用函数的声明和函数原型	87
8.5 数组作为函数参数	88
8.6 局部变量和全局变量	93
8.6.1 局部变量	93
8.6.2 全局变量	94
习题	97
第9章 预处理命令	99
9.1 概述	99
9.2 宏定义	99
9.3 文件包含	100
第10章 指针	102
10.1 地址指针的基本概念	102
10.2 变量的指针和指向变量的指针变量	103
10.2.1 定义一个指针变量	103
10.2.2 指针变量的引用	104
10.2.3 指针变量作为函数参数	107
10.2.4 指针变量几个问题的进一步说明	111
10.3 指向数组的指针变量	113
10.3.1 指向数组元素的指针	113
10.3.2 通过指针引用数组元素	114
10.3.3 数组名作函数参数	117
10.4 字符串的指针和指向字符串的指针变量	123
10.4.1 字符串的表示形式	123
10.4.2 使用字符串指针变量与字符数组的区别	126
10.5 字符串应用举例	126
习题	127
第11章 结构体	129
11.1 定义一个结构的一般形式	129
11.2 结构类型变量的说明	129
11.3 结构变量成员的表示方法	132
11.4 结构体变量的赋值	132

11.5	结构变量的初始化	133
11.6	结构数组的定义	133
11.7	结构指针变量的说明和使用	136
11.7.1	指向结构变量的指针	136
11.7.2	指向结构数组的指针	138
11.7.3	结构指针变量作函数参数	139
	习题	141
第 12 章	文件	143
12.1	C++文件概述	143
12.2	文件的打开与关闭	144
12.2.1	文件的打开 (fopen 函数)	144
12.2.2	文件关闭函数 (fclose 函数)	146
12.3	文件的读写	146
12.3.1	字符读写函数 fgetc 和 fputc	146
12.3.2	字符串读写函数 fgets 和 fputs	148
12.3.3	格式化读写函数 fscanf 和 fprintf	150
12.4	C++库文件	151
12.5	文件操作程序举例	152
附录		155
附录 1	常用字符 ASCII 代码对照表	155
附录 2	运算符和结合性	155
附录 3	三级考试中涉及的 C++关键字	156
附录 4	常用 C++库函数	156
附录 5	C++常用菜单命令	160
参考文献		162

第 1 章 C++语言概述

1.1 C++语言概述

1.1.1 C++语言的发展过程

语言的发展是一个逐步递进的过程，C++是直接从 C 语言发展过来的，而 C 语言是从 B 语言发展过来的。其中最有趣的是 CPL 中 C 的由来，由于当时这个语言是剑桥大学和伦敦大学合作开发的，在伦敦的人员加入之前，C 表示剑桥，伦敦人员加入之后，C 表示 Combined 组合。还有一种非正式的说法，C 表示 Christopher，因为 Christopher 是 CPL 背后的主要动力。

最初导致 C++诞生的原因是在 Bjarne 博士等人试图去分析 UNIX 的内核的时候，这项工作开始于 1979 年 4 月，当时由于没有合适的工具能够有效的分析由于内核分布而造成的网络流量，以及怎样将内核模块化。同年 10 月，Bjarne 博士完成了一个可以运行的预处理程序，称之为 Cpre，它为 C 加上了类似 Simula 的类机制。在这个过程中，Bjarne 博士开始思考是不是要开发一种新的语言，当时贝尔实验室对这个想法很感兴趣，就让 Bjarne 博士等人组成一个开发小组，专门进行研究。

当时不是叫做 C++，而是 C with class，这是把它当作一种 C 语言的有效扩充。由于当时 C 语言在编程界居于老大的地位，要想发展一种新的语言，最强大的竞争对手就是 C 语言，所以当时有两个问题最受关注：C++要在运行时间、代码紧凑性和数据紧凑性方面能够与 C 语言相媲美，但是还要尽量避免在语言应用领域的限制。在这种情况下，一个很自然的想法就是让 C++从 C 语言继承过来，但是 Bjarne 博士更具有先见之明，他为了避免受到 C 语言的局限性，参考了很多的语言。

目前最流行的 C++语言有以下 2 种：

- ① Visual C++ 6.0;
- ② C++ Builder。

这些 C++语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

教育部考试中心规定：全国计算机等级考试三级使用编程软件——Visual C++ 6.0。

1.1.2 C++语言的特点

- ① C++语言简洁、紧凑，使用方便、灵活。

C++常用关键字有 32 个（见附录 C）和 9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。

注意：在 C++语言中，关键字都是小写的。

- ② 运算符丰富。关键字少了，运算符却很丰富，共有 34 种。C++把括号、赋值、逗号等都作为运算符处理。从而使 C++的运算类型极为丰富，可以实现其他高级语言难以实现的运算。关于运算符请查阅本书附录。

③ 数据结构类型丰富。除了 C++ 系统自己的数据结构，程序员还可以自创类型，所以 C++ 语言使用起来灵活、多样。

④ 语法限制不太严格，程序设计自由度大。

⑤ 具有结构化的控制语句。

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。因此不能简单地把 C++ 语言说明为高级语言，它是一种介于汇编语言和高级语言之间的中级语言。

⑦ 与汇编语言相比，用 C++ 语言写的程序可移植性好。

⑧ 生成目标代码质量高，程序执行效率高。

但是，C++ 语言也有缺点。C++ 语言对程序员要求高，程序员用 C++ 写程序会感到限制少、灵活性大、功能强，较其他高级语言在学习上要困难一些，尤其对于是初学者。

C++ 的特长不在科学计算和管理领域。对操作系统和系统实用程序以及需要对硬件进行操作的场合，用 C++ 语言会明显地优越于其他高级语言，有些大型应用软件也是用 C++ 语言编写的。C++ 语言除了用于教学外，还有广泛的应用领域，因此更有生命力。这种人们为了编写 UNIX 操作系统而创造的高级语言强调灵活性，使程序员有较大的自由度，适应宽广的应用面。

C++ 语言对程序员较其他高级语言要高。程序员编写程序会感到限制很少、灵活性大、功能强，理论上来说可以编写任何类型的程序。

1.2 简单的 C++ 程序介绍

为了说明 C++ 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，表现了 C++ 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到组成一个 C++ 源程序的基本部分和书写格式。

【例 1-1】

```
#include<stdio.h>
main()
{
    printf("I love china.\n");
}
```

① main 是主函数的函数名，表示这是一个主函数。

② 每一个 C++ 源程序都必须有，且只能有一个主函数 (main 函数)。

③ 函数调用语句，printf 函数的功能是把要输出的内容送到显示器去显示。

④ printf 函数是一个由系统定义的标准函数，可在程序中直接调用。

【例 1-2】

```
#include<stdio.h>                                /* include 称为文件包含命令*/
main()
{
    double x,y,sum;                                /*定义三个实数变量，以备后面程序使用*/
    printf("input number:\n");                    /*显示提示信息*/
    scanf("%lf%lf",&x,&y);                          /*从键盘获得一个实数 x*/
}
```

```
sum=x+y;                                /*求 x 和 y 的和,并把它赋给变量 sum*/  
printf("x= %lf  y=%lf  sum=%lf\n",x,y,sum); /*显示程序运算结果*/  
}                                         /* main 函数结束*/
```

① C++程序是由函数构成的。一个C++源程序至少包含一个main函数。也可以调用其他函数,本例中的scanf()和printf()就是C++的系统函数,具有输入输出功能。

② 需要说明的是,C++语言规定对scanf和printf这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令#include<stdio.h>。

同样,在例1.1中使用了printf函数,也省略了包含命令。

③ 在例题中的主函数体中又分为两部分,一部分为说明部分,另一部分为执行部分。说明是指变量的类型说明。例题1.1中未使用任何变量,因此无说明部分。C语言规定,源程序中所有用到的变量都必须先说明,后使用,否则将会出错。这一点是编译型高级程序设计语言的一个特点,本例中使用了两个变量x,y,sum,用来表示输入的自变量用类型说明符double来说明。说明部分后的4行为执行部分或称为执行语句部分,用以完成程序的功能。执行部分的第一行是输出语句,调用printf函数在显示器上输出提示字符串,请用户输入自变量x的值。第二行为输入语句,调用scanf函数,接受键盘上输入的数并存入变量x,y中。第三行是使用赋值表达式对变量sum进行赋值。第四行是用printf函数输出变量sum的值,程序结束。

④ C++语言的格式非常的自由,一行内可以写几个语句,一个语句可以分写在多行上。C++程序是没有行号的,没有严格规定书写格式。

⑤ 每个语句的最后必须有一个分号。分号是C++语言的必要组成部分,是这种语言的特点。因为C++语句是以分号为结束而不是以回车为结束语,所以C++语句可以一行写多句,也可多行写一句。

运行本程序时,首先在显示器屏幕上给出提示串input number,这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数,如56,按下回车键,接着在屏幕上给出计算结果。

1.3 C++的基本语法

1.3.1 C++源程序的结构特点

- 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文件或源程序的最前面。
- ⑤ 每一个说明、每一个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间隔。若已有明显的间隔符,也可不再加空格来间隔。

1.3.2 书写程序时应遵循的规则

从书写清晰、便于阅读、理解、维护的角度出发,在书写程序时应遵循以下规则:

- ① 一个说明或一个语句占一行。

② 用 {} 括起来的部分, 通常表示了程序的某一层结构。{} 一般与该结构语句的第一个字母对齐, 并单独占一行。

③ 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。以便看起来更加清晰, 增加程序的可读性。

在编程时应力求遵循这些规则, 以养成良好的编程风格。

1.3.3 C++语言的字符集

字符是组成语言的最基本的元素。C++语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量, 字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

① 请注意 C++语言严格区分大小写字母, 在 C++语言中大写 ABC 和小写 abc 具有不同的含义。

② 字母: 小写字母 a~z 共 26 个; 大写字母 A~Z 共 26 个。

③ 数字: 0~9 共 10 个。

④ 空白符。空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时, 只起间隔作用, 编译程序对它们忽略不计。因此在程序中使用空白符与否, 对程序的编译不发生影响, 但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

⑤ 标点和特殊字符。

1.3.4 C++语言词汇

在 C 语言中使用的词汇分为 6 类: 标识符, 关键字, 运算符, 分隔符, 常量, 注释符等。

(1) 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外, 其余都由用户自定义。C++规定, 标识符只能是字母 (A~Z, a~z)、数字 (0~9)、下划线 (_) 组成的字符串, 并且其第一个字符必须是字母或下划线。

以下标识符是合法的:

a, x, x3, BOOK_1, sum5

以下标识符是非法:

6us 以数字开头

x*Pi 出现非法字符*

-ttx 以减号开头

hyuijn-1 出现非法字符-(减号)

在使用标识符时还必须注意以下几点。

① 标准 C++不限制标识符的长度, 但它受各种版本的 C++语言编译系统限制, 同时也受到具体机器的限制。例如在某版本 C++中规定标识符前 8 位有效, 当两个标识符前 8 位相同时, 则被认为是同一个标识符。

② 在标识符中, 大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。

③ 标识符虽然可由程序员随意定义, 但标识符是用于标识某个量的符号。因此, 命名应尽量有相应的意义, 以便于阅读理解, 做到“顾名思义”。

(2) 关键字

关键字是由 C++语言规定的具有特定意义的字符串, 通常也称为保留字。用户定义的标识符不应与关键字相同。C++语言的关键字分为以下几类。

① 类型说明符。用于定义、说明变量、函数或其他数据结构的类型。如前面例题中用到的 `int`, `double` 等。

② 语句定义符。用于表示一个语句的功能, 如 `if else` 就是条件语句的语句定义符。

③ 预处理命令。用于表示一个预处理命令, 如前面各例中用到的 `include`。

(3) 运算符

C++语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式, 表示各种运算功能。运算符由一个或多个字符组成。

(4) 分隔符

在C++语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中, 分隔各个变量。空格多用于语句各单词之间, 作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔, 否则将会出现语法错误, 例如把 `int a;` 写成 `inta;` C++编译器会把 `inta` 当成一个标识符处理, 其结果必然出错。

(5) 常量

C++语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在后面章节中将专门给予介绍。

(6) 注释符

C++语言的注释符是以“/*”开头并以“*/”结尾的串。在“/*”和“*/”之间的即为注释。程序编译时, 不对注释做任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来, 使翻译跳过不做处理, 待调试结束后再去掉注释符。

1.4 Visual C++ 6.0 开发环境及 C++上机过程

1.4.1 Visual C++6.0 简介

Visual C++6.0 (以下简称 VC6) 是美国 Microsoft 公司于 1998 年 10 月推出的 Visual Studio 6.0 开发工具包中的一个产品。VC6 为用户提供了一个集成开发环境, 它使得 C++程序的编辑、编译、链接、调试和运行等工作都能够在统一的操作界面下完成, 因此使用起来十分方便。在实际工作中 VC6 主要用于开发 Windows 图形用户界面程序, 然而对于学习 C++语言本身的初学者来说只需要使用 VC6 提供的部分功能来编写标准 C++程序。本章节将对 VC6 的集成开发环境与项目、标准 C++程序的编写、基本的调试功能等内容进行介绍。

使用 Visual C++6.0 集成环境来编辑、编译并运行下面简单 C++程序。

① `#include <stdio.h>`

```
void main()
{
    printf("This is a C++ program.\n");
}
```

② `#include <stdio.h>`

```
int max(int,int);    //声明自定义函数
void main()          //主函数
{
```

```
int a,b,c;
printf("input two number:\n");
scanf("%d%d",&a,&b);
c=max(a,b);           //调用 max 函数，将得到的值赋给 C++
printf("max=%d\n",%c);
}
int max(int x,int y) //定义 max 函数，函数值为整型，形式参数 x, y 为整型
{
    int z;           //max 函数中的声明部分，定义本函数中用到的变量 z 为整型
    if(x>y) z=x;
    else z=y;
    return(z);       //将 z 的值返回，通过 max 带回调用处
}
```

1.4.2 操作步骤

(1) Visual C++6.0 启动后的用户界面

选择菜单“开始/程序/Microsoft Visual Studio 6.0/Microsoft Visual C++ 6.0”,得到 Visual C++ 6.0 启动后的用户界面,如图 1-1 所示。

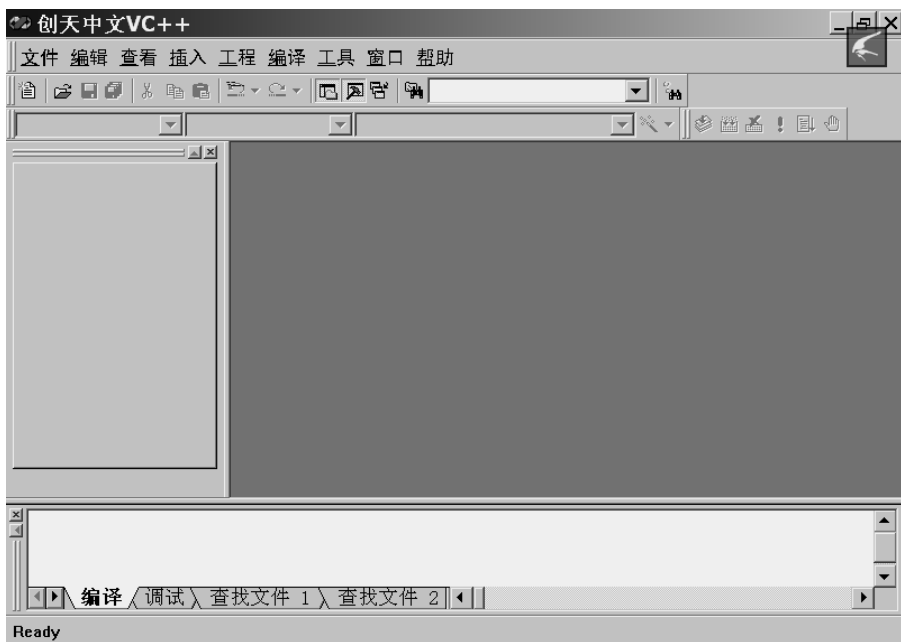


图 1-1 Visual C++ 6.0 用户界面

(2) 创建新工程

① 单击菜单“文件/新建”，显示“新建”对话框，如图 1-2 所示。

② 在此对话框的列表栏中，选择“Win32 Console Application”（Win32 控制台应用程序）。在“工程”栏中输入一个工程名，如实验一。在位置文本框中选择工程所在的路径，如图中所示 H:\C++实验\实验一。单击“确定”按钮。



图 1-2 “新建”对话框

③ 在点击的“Win32 Consol Application-Step 1 of 1”对话框中选择 An empty project 单项。然后单击“Finish”按钮，如图 1-3 所示。

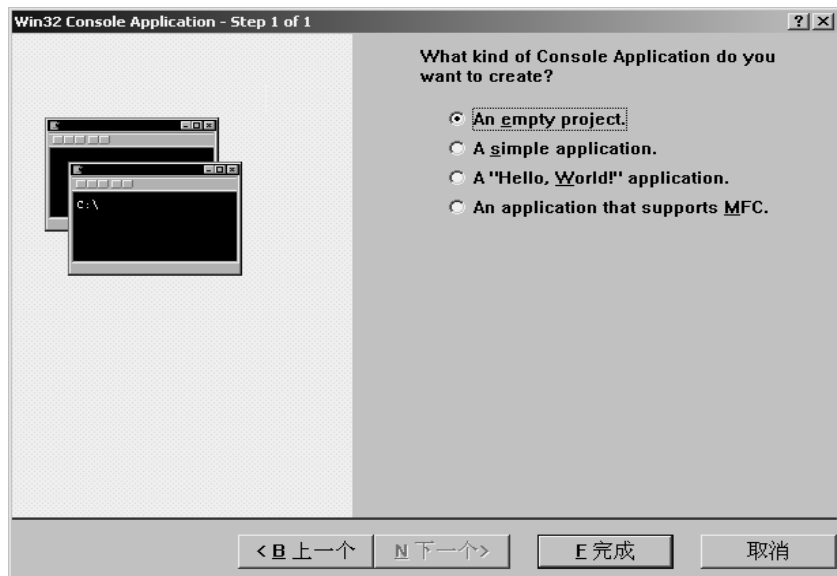


图 1-3 “Win32 Consol Application-Step 1 of 1”对话框

④ 在“新建工程信息”对话框中单击“确认”按钮，完成工程创建过程。

1.4.3 创建 C++源程序文件

① 编辑 C++源程序。编辑 C++源程序时，选择“文件”菜单项，出现一个下拉式菜单，如图 1-4 所示，再选择该菜单中的“新建”选项（热键为 Ctrl+N）。打开“新建”窗口。

② 选择“文件”，在它的下拉列表框中选择“C++ Source File”，如图 1-4 所示，再按“确定”，出现编辑屏幕，如图 1-5 所示。

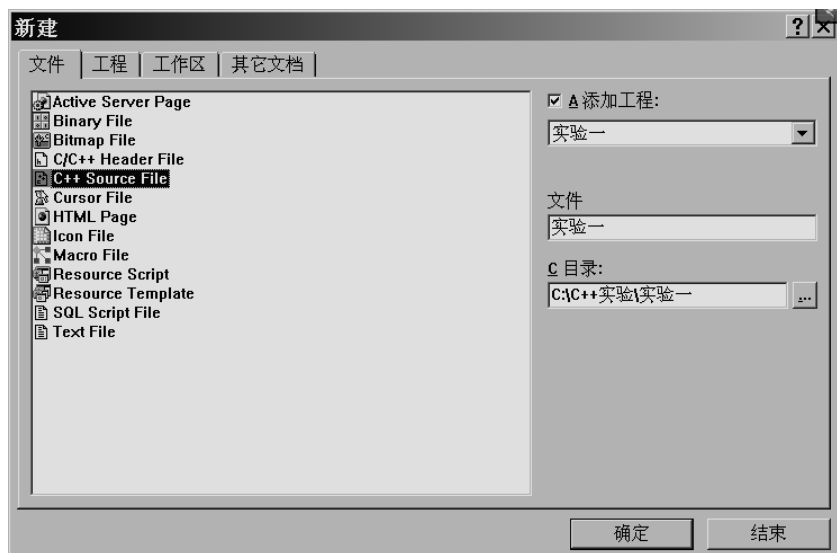


图 1-4 “新建”窗口

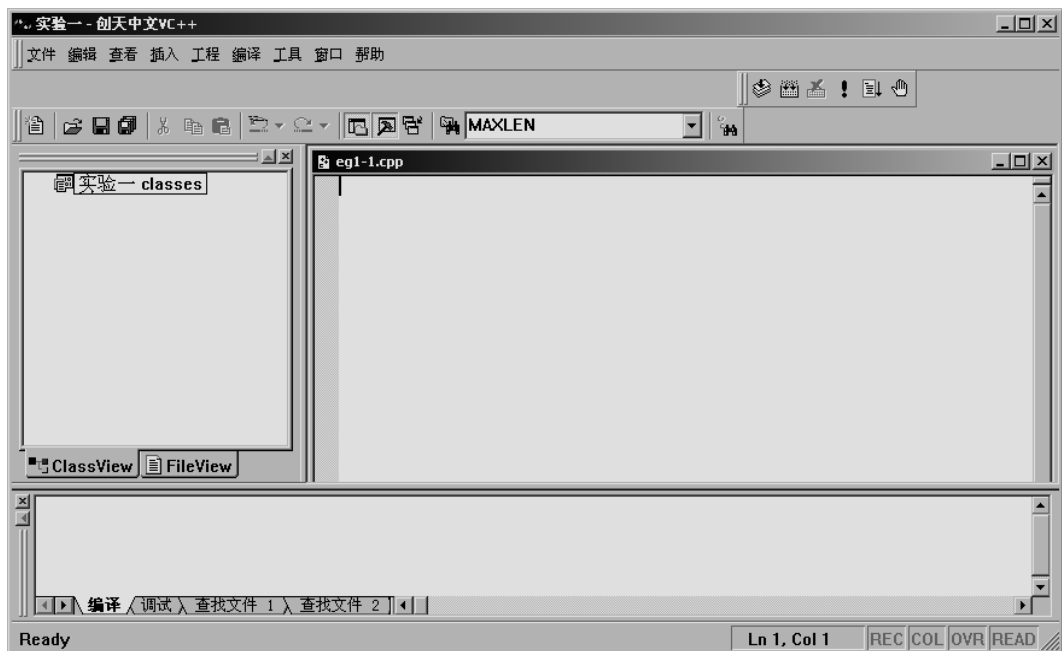


图 1-5 “C++”源程序编辑

③ 在“C++源程序编辑”窗口下编辑 C++源程序。在编辑窗口键入 C++源程序。如图 1-6 所示，键入如下源程序：

④ 连接和运行源程序

- “编译”。出现“编译”的下拉菜单，在该下拉式菜单中选择“编译 eg1-1.cpp”菜单项，这时系统开始对当前的源程序进行编译，在编译过程中，将所发现的错误显示在屏幕下方的“编译”窗口中。所显示的错误信息中指出该错误所在行号和该错误的性质。用户可根据这些错误信息进行修改。上述程序的“编译”窗口，如图 1-7 所示。

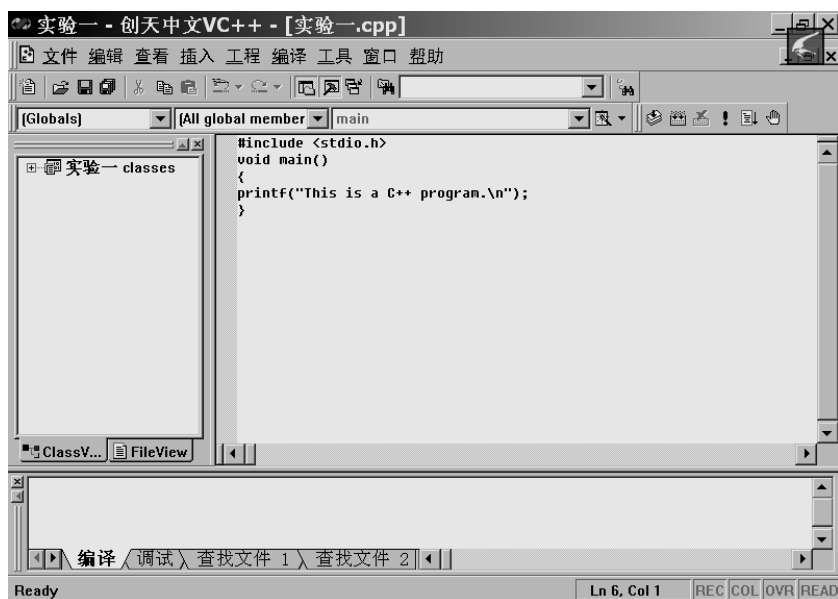


图 1-6 编辑 C++源程序

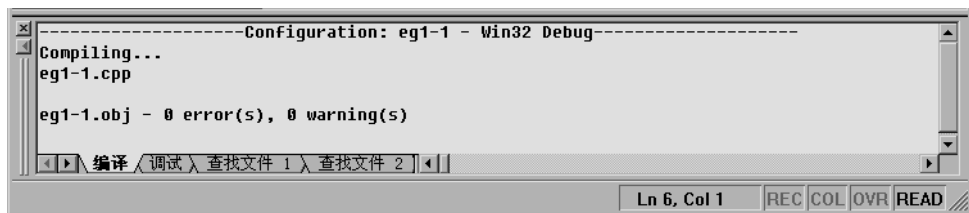


图 1-7 “编译”窗口

• 无错误后，可进行连接生成可执行文件（.exe），这时选择“编译”下拉菜单中的“构件 eg1-1.exe”选项。“编译”窗口出现如图 1-8 所示的信息说明编译连接成功，并生成以源文件名作为名字的可执行文件（eg1-1.exe）。

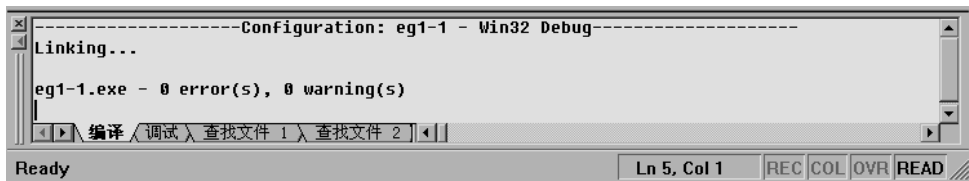


图 1-8 编译连接信息

• 可执行文件运行的方法是选择“编译”菜单项中“执行 eg1-1.exe”选项。这时，运行该可执行文件，并将结果显示在另外一个显示执行文件输出结果的窗口中，如图 1-9 所示。

⑤ 关闭、打开工作区

单击菜单“文件/关闭工作区”，关闭工作区。

单击菜单“文件/打开工作区”，在弹出的对话框中选定“H: \C++实验\实验一\实验一.dsw”，单击“打开”按钮，则可打开工作区，对已建立的工程文件进行修改。