



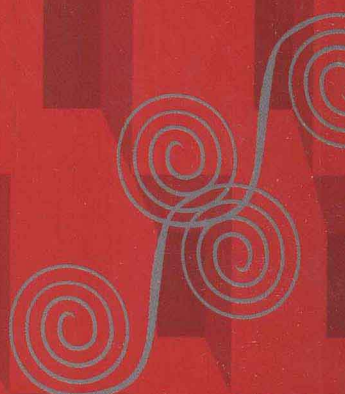
信毅教材大系

# C++ 编程

——面向问题的设计方法

• 李刚 主编

C++ Programming  
— A Problem-Based Approach



复旦大学出版社

 信毅教材大系

# C++ 编程

## ——面向问题的设计方法

• 李刚 主编

C++ Programming  
— A Problem-Based Approach



 复旦大学出版社

图书在版编目(CIP)数据

C++编程——面向问题的设计方法/李刚主编. —上海:复旦大学出版社,2013.11  
信教教材大系  
ISBN 978-7-309-10100-3

I. C… II. 李… III. C语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆CIP数据核字(2013)第230387号

C++编程——面向问题的设计方法

李刚 主编

责任编辑/张志军

复旦大学出版社有限公司出版发行

上海市国权路579号 邮编:200433

网址:fupnet@fudanpress.com <http://www.fudanpress.com>

门市零售:86-21-65642857 团体订购:86-21-65118853

外埠邮购:86-21-65109143

常熟市华顺印刷有限公司

开本 787×1092 1/16 印张 41.75 字数 892 千

2013年11月第1版第1次印刷

ISBN 978-7-309-10100-3/T·493

定价:76.00元

---

如有印装质量问题,请向复旦大学出版社有限公司发行部调换。

版权所有 侵权必究

世界高等教育的起源可以追溯到 1088 年意大利建立的博洛尼亚大学,它运用社会化组织成批量培养社会所需要的人才,改变了知识、技能主要在师徒间、个体间传授的教育方式,满足了大家获取知识的需要,史称“博洛尼亚传统”。

19 世纪初期,德国的教育家洪堡提出“教学与研究相统一”和“学术自由”的原则,并指出大学的主要职能是追求真理,学术研究在大学应当具有第一位的重要性,即“洪堡理念”,强调大学对学术研究人才的培养。

在洪堡理念广为传播和接受之际,德国都柏林天主教大学校长纽曼发表了“大学的理想”的著名演说,旗帜鲜明地指出“从本质上讲,大学是教育的场所”,“我们不能借口履行大学的使命职责,而把它引向不属于它本身的目标。”强调培养人才是大学的唯一职能。纽曼关于“大学的理想”的演说让人们重新审视和思考大学为何而设、为谁而设的问题。

19 世纪后期到 20 世纪初,美国威斯康辛大学查尔斯·范海斯校长提出“大学必须为社会发展服务”的办学理念,更加关注大学与社会需求的结合,从而使大学走出了象牙塔。

2011 年 4 月 24 日,胡锦涛总书记在清华大学百年校庆庆典上,指出高等教育是优秀文化传承的重要载体和思想文化创新的重要源泉,强调要充分发挥大学文化育人和文化遗产创新的职能。

总而言之,随着社会的进步与变革,高等教育不断发展,大学的功能不断扩展,但始终都在围绕着人才培养这一大学的根本使命,致力于不断提高人才培养的质量和水平。

对大学而言,优秀人才的培养,离不开一些必要的物质条件保障,但更重要的是高效的执行体系。高效的执行体系应该体现在三个方面:一是科学合理的学科专业结构,二是能洞悉学科前沿的优秀的师资队伍,三是作为知识载体和传播媒介的优秀教材。教材是体现教学内容与教学方法的知识载体,是进行教学的基本工具,也

是深化教育教学改革,提高人才培养质量的重要保证。

一本好的教材,要能反映该学科领域的学术水平和科研成果,能引导学生沿着正确的学术方向步入所向往的科学殿堂。因此,加强高校教材建设,对于提高教育质量、稳定教学秩序、实现高等教育人才培养目标起着重要的作用。正是基于这样的考虑,江西财经大学与复旦大学出版社达成共识,准备通过编写出版一套高质量的教材系列,以期进一步锻炼学校教师队伍,提高教师素质和教学水平,最终将学校的学科、师资等优势转化为人 才培养优势,提升人才培养质量。为凸显江财特色,我们取校训“信敏廉毅”中一前一尾两个字,将这个系列的教材命名为“信毅教材大系”。

“信毅教材大系”将分期分批出版问世,江西财经大学教师将积极参与这一具有重大意义的学术事业,精益求精地不断提高写作质量,力争将“信毅教材大系”打造成业内有影响力的高端品牌。“信毅教材大系”的出版,得到了复旦大学出版社的大力支持,没有他们的卓越视野和精心组织,就不可能有这套系列教材的问世。作为“信毅教材大系”的合作方和复旦大学出版社的一位多年的合作者,对他们的敬业精神和远见卓识,我感到由衷的钦佩。

王 乔

2012年9月19日

C++语言的前身是诞生于20世纪70年代初的C语言,其强大的功能和各方面的优点自此逐渐为人们认识,到了80年代,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了广泛的使用。成为当代最优秀的程序设计语言之一。

但是,由于C语言是面向过程的结构化和模块化的程序设计语言,当处理问题的规模和复杂度较大时,它就显得力不从心了。尤其是在开发大型软件的时候,这一缺点更是突出。C++语言在保留了C语言的所有优点的基础上,增加了适用于面向对象的程序设计的类类型。因此,C++既支持面向过程的设计,又支持面向对象的程序设计,是一种混合型的程序设计语言。

就像C++之父Stroustrup博士所说:C++语言有很多特性,但是我们学习时只需要学对自己有用的特性就好了。这也是本书写作时所坚持的原则。更具体更全面的東西,在掌握了核心内容之后,学生只需要学会查手册,就可以了解;经过实践的訓練,就可以掌握。为此,作者在编著中抵制住“要全面”的诱惑,大胆把C++部分的异常处理、流、STL库、算法等内容从本书中去掉,而把重点突出在类设计、对象的操作等面向对象设计的核心技术上面。

本书内容组织如下:全书共分两篇,第一篇为语言基础篇(第1~7章),通过大量实例和图片图文并茂地讲解了C++和C语言的共同部分:结构化程序设计的部分,内容包括常量变量、数据类型、运算符、表达式、数组、函数、指针和结构体等。这一篇可以单独拿出来作为完整的C语言课程教学的内容。这一部分采用循序渐进的方式引入各个知识点,而不是分门别类罗列;而且围绕数据计算和文字处理两个大方面的问题进行组织。第一篇建议授课课时为64课时,含上机课时16个(机时可根据实际需要增减)。

第二篇为深入篇(面向对象篇,第8~12章),通过实例生动讲解面向对象程序设计中的基本特征,是C++不同于C的部分,是C++的面向对象的新特性,内容有:重载、类和对象、对象的运算、

继承和派生、多态性等。这一部分首先介绍了 C++ 引入的一些新特性,如流式输入、函数重载等(第 8 章);然后引入类这种用户自定义类型,在后续的第 9~12 章围绕着类类型讨论如何用类定义变量(创建对象)、如何设计类以使得它可以像内置类型一样、在语义许可的条件下参加各种运算,如赋值、比较、输入输出等(通过运算符重载)。读者应把握这条主线,以防在众多新的知识点中迷失。此外,第 11 章的继承和第 12 章的多态是重要的代码重用技术。使用继承和多态可以有效利用原有代码,也可以在原有代码上添加新的特性,而只要对原来代码做很小的改动(扩展原代码,或曰升级)。这一部分的建议教学课时为 96 课时(含上机课时 32,可按实际情况安排)。

本书的总体架构设计、编写、统稿、定稿均由李刚负责,在整个过程中广泛听取和采纳了课程组同事和相关专业学生的意见建议。在本书的编写过程中复旦大学出版社的编辑、编者所在学院的领导同事、图书采购中心的老师给予了很大的支持,并提出了很多宝贵意见。在此对他们的辛勤工作和热心帮助一并表示由衷的感谢!另外,本人的爱人刘琳红同志在编著的过程中给了本人极大的帮助和支持,可以说:没有她的付出,就没有这本书的面世。以此书向她表示敬意!

另外,本书假设的对象是没有任何的 C++ 或其他语言的编程经验的读者,也可供有经验的读者参考。本书可作为教学课本,也可作为计算机等级考试参考书和培训班的培训教材,但是建议结合考试辅导材料和市面上出版的习题集使用。

由于计算机技术和程序设计技术发展迅速,作者水平有限及本书编辑仓促,书中的疏漏与不当之处在所难免,敬请广大读者和同仁不吝赐教,拨冗指正。

编 者

2013 年 6 月

## 第一篇 基础篇——C++中的 C

|                                  |     |
|----------------------------------|-----|
| <b>第 1 章 认识 C/C++ 语言程序</b> ..... | 003 |
| 1.1 最简单的 C 程序 .....              | 003 |
| 1.2 带简单输出的 C 程序 .....            | 012 |
| 1.3 带有可变内容的输出 .....              | 018 |
| 1.4 C++ 方式的输出 * .....            | 025 |
| 1.5 本章小结 .....                   | 026 |
| <b>第 2 章 控制程序的执行方式</b> .....     | 028 |
| 2.1 根据变量的值选择性输出 .....            | 028 |
| 2.2 通过输入值控制程序的行为 .....           | 036 |
| 2.3 重复输出的自动化实现 .....             | 038 |
| 2.4 格式化输入对程序执行的影响 .....          | 045 |
| 2.5 本章小结 .....                   | 050 |
| <b>第 3 章 函数、数组与指针</b> .....      | 055 |
| 3.1 对重复出现和使用的代码用函数进行封装 .....     | 055 |
| 3.2 用数组存储大量相似数据 .....            | 079 |
| 3.3 使用指针访问内存中的相似数据 .....         | 091 |
| 3.4 本章小结 .....                   | 120 |
| <b>第 4 章 信息的高效查询</b> .....       | 127 |
| 4.1 排序——让随机的信息变得有序 .....         | 127 |
| 4.2 查找——在海量数据中对特定信息的搜索 .....     | 169 |
| 4.3 本章小结 .....                   | 174 |

|                                    |                |
|------------------------------------|----------------|
| <b>第5章 文字信息的处理 .....</b>           | <b>177</b>     |
| 5.1 文本信息的表示与存储 .....               | 177            |
| 5.2 文本信息的处理方法 .....                | 179            |
| 5.3 用于文本信息处理的库函数 .....             | 185            |
| 5.4 转换数值和文字信息 .....                | 200            |
| 5.5 多条文本信息的表示与处理 .....             | 214            |
| 5.6 文字信息的检索 .....                  | 231            |
| 5.7 多条信息的显示 .....                  | 242            |
| 5.8 本章小结 .....                     | 247            |
| <br><b>第6章 复杂记录的表示、处理和存储 .....</b> | <br><b>249</b> |
| 6.1 复杂数据记录对应的C类型 .....             | 249            |
| 6.2 利用数组存储与处理数据 .....              | 260            |
| 6.3 为提高数据插入和删除效率所作的改进 .....        | 261            |
| 6.4 多个数据共享一块内存* .....              | 294            |
| 6.5 本章小结 .....                     | 296            |
| <br><b>第7章 数据的永久存储与加载 .....</b>    | <br><b>299</b> |
| 7.1 改变程序输入的来源 .....                | 300            |
| 7.2 从命令行参数获取程序执行信息 .....           | 309            |
| 7.3 把信息永久保存到文件中 .....              | 311            |
| 7.4 关于文件的其他操作 .....                | 318            |
| 7.5 本章小结 .....                     | 320            |

## 第二篇 深入篇——面向对象编程

|                              |                |
|------------------------------|----------------|
| <b>第8章 函数重载：一名多能 .....</b>   | <b>325</b>     |
| 8.1 C++方式的输入输出 .....         | 325            |
| 8.2 函数的重载 .....              | 334            |
| 8.3 函数模板——类型的参数化 .....       | 348            |
| 8.4 编译器如何选择正确的函数版本 .....     | 361            |
| 8.5 内联函数 .....               | 364            |
| 8.6 本章小结 .....               | 365            |
| <br><b>第9章 抽象：类和对象 .....</b> | <br><b>368</b> |
| 9.1 使用结构体操作记录 .....          | 369            |
| 9.2 使用类操作记录 .....            | 371            |

|               |                          |            |
|---------------|--------------------------|------------|
| 9.3           | 对象的创建、初始化与销毁 .....       | 387        |
| 9.4           | 对象的动态生成与动态销毁 .....       | 419        |
| 9.5           | 本章小结 .....               | 420        |
| <b>第 10 章</b> | <b>运算：操作对象 .....</b>     | <b>427</b> |
| 10.1          | 赋值与拷贝 .....              | 427        |
| 10.2          | 对象的输入输出 .....            | 459        |
| 10.3          | 对象的其他运算 .....            | 463        |
| 10.4          | 运算符重载的一般规则 .....         | 469        |
| 10.5          | 其他友元 .....               | 471        |
| 10.6          | 声明与实现的分离 .....           | 478        |
| 10.7          | 再谈拷贝 .....               | 486        |
| 10.8          | 本章小结 .....               | 495        |
| <b>第 11 章</b> | <b>继承：代码的重用 .....</b>    | <b>503</b> |
| 11.1          | 派生类的定义和使用 .....          | 504        |
| 11.2          | 名字重复的处理 .....            | 519        |
| 11.3          | 对象的创建、初始化与析构 .....       | 525        |
| 11.4          | 多重继承 .....               | 542        |
| 11.5          | 虚拟继承 .....               | 547        |
| 11.6          | 本章小结 .....               | 555        |
| <b>第 12 章</b> | <b>多态：行为的差别 .....</b>    | <b>559</b> |
| 12.1          | 类之间的关系 .....             | 559        |
| 12.2          | 联编与多态 .....              | 566        |
| 12.3          | 虚函数 .....                | 567        |
| 12.4          | 虚函数表 .....               | 580        |
| 12.5          | 纯虚函数与抽象类 .....           | 595        |
| 12.6          | 运行时类型识别 .....            | 599        |
| 12.7          | 本章小结 .....               | 610        |
| <b>附录 I</b>   | <b>C++和C的运算符 .....</b>   | <b>614</b> |
| <b>附录 II</b>  | <b>C和C++的关键字 .....</b>   | <b>618</b> |
| <b>附录 III</b> | <b>DOS 命令参考 .....</b>    | <b>622</b> |
| <b>附录 IV</b>  | <b>string 使用说明 .....</b> | <b>642</b> |
| <b>参考文献</b>   | <b>.....</b>             | <b>657</b> |

# 第一篇 基础篇

## ——C++中的 C

- 第 1 章 认识 C/C++ 语言程序
- 第 2 章 控制程序的执行方式
- 第 3 章 函数、数组与指针
- 第 4 章 信息的高效查询
- 第 5 章 文字信息的处理
- 第 6 章 复杂记录的表示、处理和存储
- 第 7 章 数据的永久存储与加载



# 第 1 章 认识 C/C++ 语言程序

## 学习要点

- C/C++ 语言程序的基本结构
- 如何在 VC++6.0 下编辑、编译、运行程序
- 使用 printf 进行基本的输出
- 转义字符与格式控制串
- 使用 printf 进行可变内容的输出
- 常量、变量以及类型转换

C 语言是 C++ 语言的子集,在学习 C++ 面向对象设计之前,首先学习它的面向对象的设计部分(跟 C 语言相同的部分),更易于入手。相对于 C++ 的庞杂的面向对象的特性,C 语言非常的简单、精炼而且强大,既像一首美妙的诗,又像一把锋利的手术刀,精炼,短小,功能强大。

## 1.1 最简单的 C 程序

首先让我们先看一个最简单的 C 程序,前面的行号是为了讲述方便加上的,编程时实际不用输入(后例皆同)。

**例 1-1** 一个最简单的 C/C++ 语言程序,它什么都不做。

```
0001 / *****/
0002 /* filename:    expl_01.c                      */
0003 / *****/
0004 int main()                      /* 主函数的定义,一个程序只有一个 */
0005 {                               /* 函数体开始 */
0006     return 0;                   //return 表示函数返回。0 是返回值
0007 }                               //函数体结束。也即函数定义语句块结束
```

### 一、程序的编辑、编译和运行

在 Visual C++ 6.0 中,执行菜单命令“文件”→“新建”,弹出“新建”对话框。在此



对话框左上,选择“工程”选项卡(VC 中的“工程”其实就是“程序”的别名),然后选择“Win32 Console Application”(Win32 控制台程序);在右边“工程名称”下面的编辑框中输入工程(Project)的名字“exp1\_01”,在下方的“位置”栏,选择保存该工程的文件夹。其他选项保持不变。然后点击[确定]按钮,如图1-1所示。

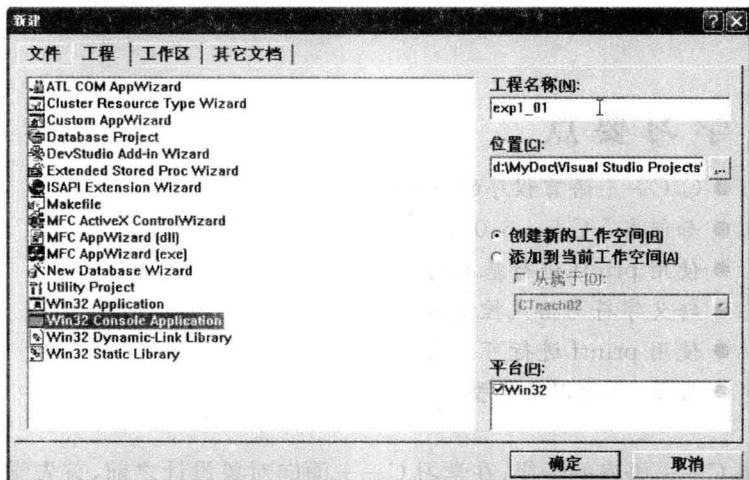


图 1-1 新建工程 exp1\_01

然后在新弹出的对话框中选择“一个空工程”,点击[完成]按钮,如图1-2所示。在新弹出的窗口点击[确定],一个空的 exp1\_01 工程就创建好了。此工程中还没有文件。可以点击 VC 左边栏工作区的 FileView(文件视图)下 exp1\_01 files 前面的“+”号展开工程所有的文件列表,如图1-3所示。

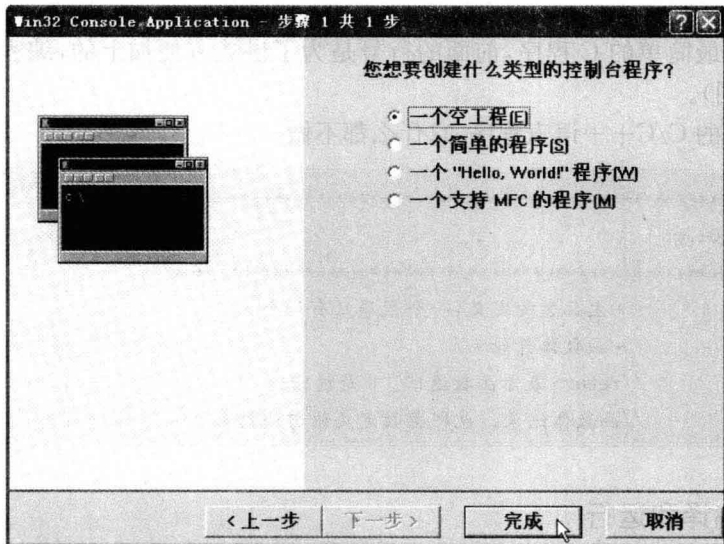


图 1-2 创建空的 exp1\_01 工程



图 1-3 工程的文件视图

因为目前工程 exp1\_01 是一个什么都没有的空工程,需要添加源代码文件。在 VC 中执行菜单命令“工程”→“增加到工程”→“新建”,为工程添加源代码,如图 1-4(a)所示。

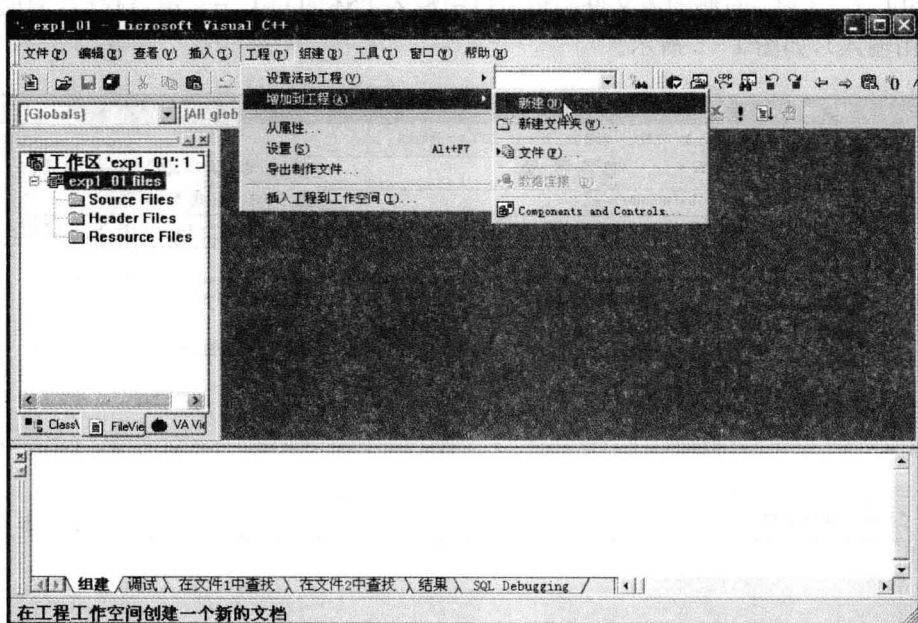


图 1-4(a) 为工程 exp1\_01 添加新建的文件

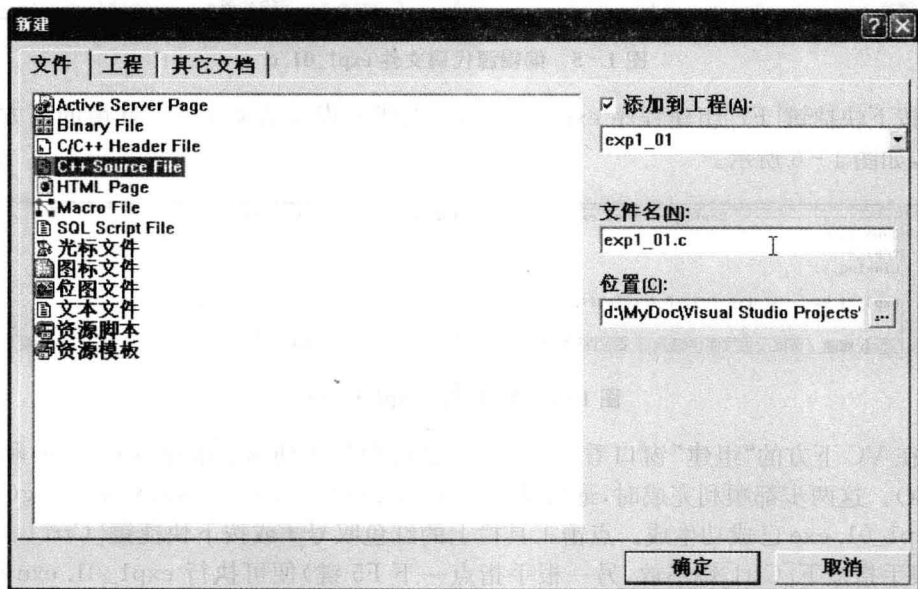


图 1-4(b) 为工程 exp1\_01 增加一个源文件 exp1\_01.c

在弹出的“新建”对话框中,选择“文件”选项卡,然后在下面的列表框中选择要添加的文件类型“C++ Source File”(C++源文件,添加 C 的源文件也要选这一项)。在右



边的“文件名”编辑框中输入要创建的源文件名“exp1\_01.c”(这个扩展名也可以写成cpp而不用c),其他设置不变。如图1-4(b)所示。然后点击[确定]按钮。之后在Visual C++(VC)右侧的源文件编辑窗口中输入上述的C/C++语言代码(前面的数字不用输入),如图1-5所示。

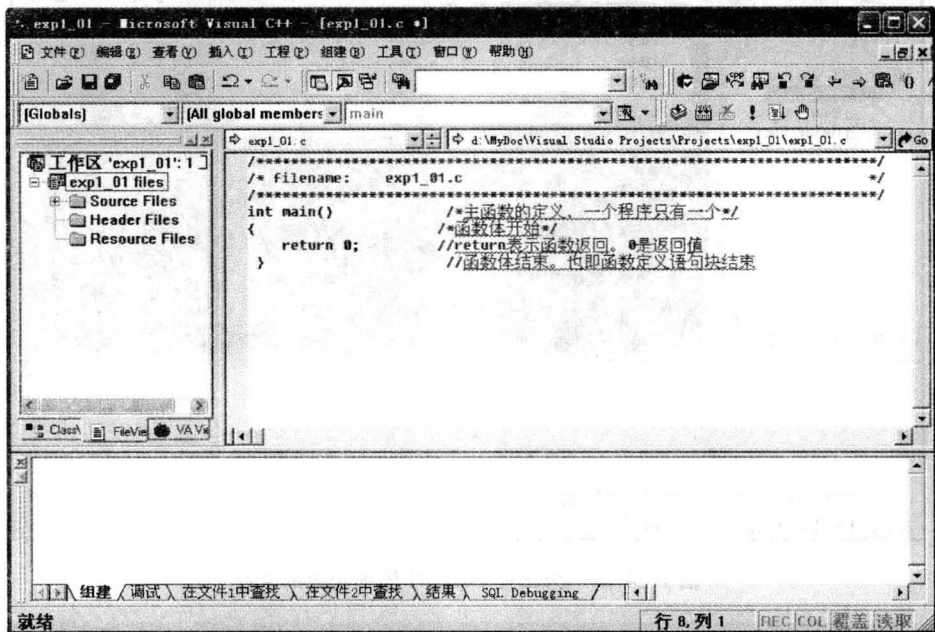


图 1-5 编辑源代码文件 exp1\_01.c

按下快捷键[F7]组建程序 exp1\_01.exe,组建过程及结果在 VC 环境的下方窗口显示,如图1-6所示。



图 1-6 组建程序 exp1\_01.exe

在 VC 下方的“组建”窗口看见,整个组建过程分为两步:编译(Compile)和连接(Link)。这两步都顺利完成时,最后显示 exp1\_01.exe - 0 error(s), 0 warning(s),表示 exp1\_01.exe 已成功生成。点击工具栏上的红色叹号!或按下快捷键[Ctrl]+[F5](一根手指按下[Ctrl]键不放,另一根手指点一下 F5 键)便可执行 exp1\_01.exe 程序。图1-7所示是执行情况。

图中的“Press any key to continue”是 VC++ 自动添加的,是为了防止应用程序执行完后立刻返回,开发者什么都看不见。

另外,也可以采取命令行的方式来执行 exp1\_01.exe 文件。方法是:首先找到

expl\_01.exe 所在的文件夹(工程 expl\_01 所在的文件夹中的“Debug”文件夹下,本例是“D:\MyDoc\Visual Studio Projects\Projects\expl\_01\Debug”),然后在 Windows 的“开始”→“运行”命令框中输入“cmd”,如图 1-8 所示。点击确定,进入命令行模式。

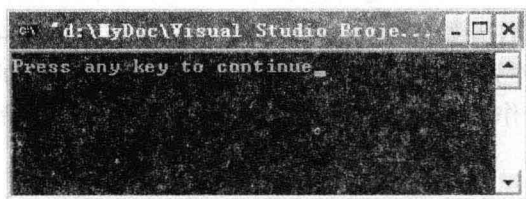


图 1-7 VC++中执行 expl\_01.exe 程序

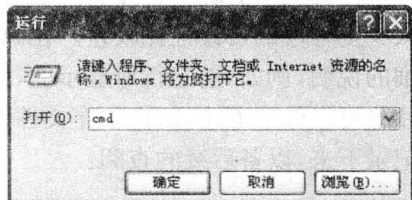


图 1-8 输入 cmd 命令进入命令行模式

在命令行模式中首先用 D: 转到工程所在的盘(本例是 D 盘,读者应换成自己工程所在的盘符)。然后用 CD 命令进入该 exe 文件所在的路径“D:\MyDoc\Visual Studio Projects\Projects\expl\_01\Debug”(在命令行输入命令: cd "D:\MyDoc\Visual Studio Projects\Projects\expl\_01\Debug")。用 dir \*.exe 命令列出该目录下的可执行文件,执行其中的 expl\_01.exe 文件(直接输入 expl\_01.exe 并回车)。以上命令执行顺序及结果如图 1-9 所示。读者看到,在命令行执行 expl\_01 这个程序的时候,既没有输入,也没有输出。程序运行完之后立刻返回到命令行控制台,不会输出“Press any key to continue”,因为程序中并没有输出这些文字的命令。

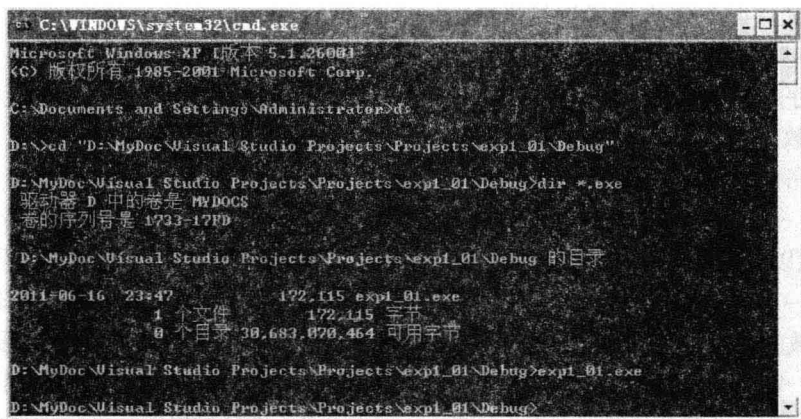


图 1-9 命令行方式执行 expl\_01.exe

到这里第一个 C/C++ 语言程序已经顺利组建并且运行起来了。

## 二、程序的基本构成要素

### 1. 注释

在程序中以“/\*”开始且以“\*/”结束的部分,以及“/\*\*”及其以后的字符序列称为 C/C++ 语言的注释。它的作用是方便源程序的阅读者对程序的理解。在本例中就是