21世纪高等学校应用型规划教材

本书荣获中国石油和化学工业优秀出版物奖·教材奖二等奖

C语言程序设计教程

第二版

李丽芬　马　睿　主编　　孙丽云　刘佩贤　副主编

化学工业出版社

21世纪高等学校应用型规划教材

C语言程序设计教程

第二版

李丽芬　马　睿　主　编
孙丽云　刘佩贤　副主编

化学工业出版社
·北京·

本书由浅入深地讲授了C语言程序设计的技术与技巧。全书分为基础知识、项目实战两部分。基础知识部分介绍了C语言的基础语法知识，包括C语言的基本概念、数据类型及其运算、选择结构、循环结构、数组、函数、编译预处理、指针、结构体和共用体、文件10章内容。每章配有程序实例和常见错误分析，有利于读者掌握程序设计的基本技巧。项目实战部分详细展示了项目开发的全过程，从需求分析、算法设计到程序编写和过程调试，以项目实战的形式引导和帮助学生解决实际问题，提高学生解决具体问题的能力。

相关的实验内容、综合实训、学科竞赛训练在与本书配套的《C语言程序设计实验指导与习题解答》中详细阐述。

本书适合作为高等院校C语言程序设计课程的教材，可以满足不同专业、不同学时的教学需要，对电子信息类专业和计算机相关专业可以讲授本书的全部内容，其他专业可以讲授本书的部分内容。本书也适合计算机水平考试培训及各类成人教育教学使用。

图书在版编目（CIP）数据

C语言程序设计教程/李丽芬，马睿主编. —2版. —北京：化学工业出版社，2015.9

21世纪高等学校应用型规划教材

ISBN 978-7-122-24847-3

Ⅰ. ①C… Ⅱ. ①李… ②马… Ⅲ. ①C语言-程序设计-高等学校-教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2015）第180021号

责任编辑：唐旭华 郝英华　　装帧设计：张 辉

责任校对：吴 静

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：三河市延风印装有限公司

787mm×1092mm 1/16 印张20¾ 字数544千字　　2015年10月北京第2版第1次印刷

购书咨询：010-64518888（传真：010-64519686）　售后服务：010-64518899

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：42.00元

本书编写人员

主　　编　李丽芬　马　睿

副 主 编　孙丽云　刘佩贤

编写人员（以姓氏笔画为序）

马　睿　云彩霞　刘佩贤　刘淑艳

孙丽云　李丽芬　张秋菊　莫德举

前　言

C 语言是目前最流行的程序设计语言之一，它既具有高级语言程序设计的特点，又具有汇编语言的功能。同时，C 语言概念简洁、语句紧凑、表达能力强、程序结构性和可读性好，很多院校都将 C 语言作为第一门计算机语言课程开设。

本书详细介绍了 C 语言程序设计中最基本的语法规则和程序设计方法。在编写过程中，力求做到概念准确、简洁，语言通俗易懂；注重前后内容的衔接，知识点安排由浅入深、循序渐进，实例选取贴近实际，可以使初学者快速地掌握 C 语言的基础知识，从而对 C 语言有一个全面、直观、系统的认识。

全书分为基础知识和项目实战两个部分。第 1 部分包括第 1～10 章，根据学习者的认知规律，精选内容，介绍 C 语言的基础语法知识。书中安排了大量例题，难易适中，覆盖面广，既包括理解语法的内容，又有联系实际提高编程能力的例子。实例设置了以下栏目。

（1）问题分析：给出解决问题的思路和算法。

（2）参考代码：给出了关键代码，并进行了详细注释。

（3）运行结果：直观的运行效果，有利于程序结果的验证。

（4）程序说明：对关键技术进行总结，对运行结果进行分析。

每章的常见错误分析部分指出了初学者在学习过程中的一些常见问题，并给出了正确的解决方法，增加了学习的方向性。

第 2 部分包括第 11～13 章，为项目实战部分。这部分通过项目开发全过程的全方位指导，从需求分析、算法设计到程序编写和过程调试，以项目实战的形式引导和帮助学生解决实际问题，提高学生解决具体问题的能力。

本书层次分明，适合作为高等院校 C 语言程序设计课程的教材，可以满足不同专业、不同学时的教学需要。为了克服学时少、内容多的矛盾，建议在教学中注重学生程序设计能力的培养，精讲多练，举一反三，采用案例教学和任务驱动相结合，以提高学生学习的兴趣和主动性，让学生在实践中逐步掌握语法规则。

为了方便教师开展教学工作，本书提供电子课件，如有需要可发邮件至 cipedu@163.com 索取。与本书配套的《C 语言程序设计实验指导与习题解答》也已同步出版。

本书的作者均为承担程序设计、数据结构等课程的骨干教师，项目实践经验丰富，积累了不少的教学素材。其中李丽芬编写第 3、6、11 章，马睿编写第 4、8、12 章，孙丽云编写第 1、9 章，刘佩贤编写第 5、10、13 章，张秋菊编写第 2、7 章。全书由李丽芬、云彩霞负责统稿和校对，刘淑艳进行程序验证，莫德举定稿。

由于水平有限，书中难免存在遗漏和不妥之处，敬请批评指正。

编者

2015 年 7 月

目　录

第 1 部分　基础知识

第 2 部分 项目实战

第1部分　基础知识

第1章 引 言

随着计算机的普及，接触计算机的人越来越多。人们发现利用计算机可以轻而易举地完成很多人类手工很难实现的任务，例如进行大规模的数学计算等。对计算机技术了解不多的人会以为电脑是万能的，它本来就具有这种能力，其实不然，计算机只能机械地执行一些命令，它之所以能完成很多复杂的任务是计算机的使用者告诉它的。人怎么来命令计算机呢？就是通过编程来告诉计算机每步该怎么做。

就像现实生活中人和人交流基本上是用语言，比如汉语、英语等（前提是交流的双方都会这种语言）。如果需要让电脑帮人们做一些事情，就要编写程序来告诉它怎么做，编写程序可以用不同的程序设计语言。C 语言就是一种程序设计语言，是人与计算机交流的一种语言，如果需要计算机来帮助你完成某些工作，则可以用 C 语言来表述你的思想并将它输入到计算机中，让计算机来“运行”它。学编程的过程，就是学习怎样用编程语言说话让计算机听懂的过程，本书介绍的编程语言为 C 语言。

1.1 C 语言的发展

C 语言是国际上广泛流行的计算机高级语言。它既可用来写系统软件，也可用来写应用软件。

C 语言的祖先是 BCPL 语言。1967 年英国剑桥大学的 Mattin Richards 推出了没有类型的 BCPL（Basic Combined Programming Language）语言。1970 年美国 AT&T 贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，设计出了简单且接近硬件的 B 语言（取 BCPL 的第一个字母）。但 B 语言过于简单，功能有限。1972～1973 年间，美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言既保持了 BCPL 和 B 语言精练且接近硬件的优点，又克服了它们过于简单，无数据类型等的缺点，C 语言的新特点主要表现在具有多种数据类型。开发 C 语言的目的在于尽可能降低用它开发的软件对硬件平台的依赖程度，使之具有可移植性。

C 语言与 UNIX 操作系统有着密切的联系，开发 C 语言的最初目的是为了更好地描述 UNIX 操作系统。C 语言的出现，促进了 UNIX 操作系统的开发，同时，随着 UNIX 的日益广泛使用，C 语言也迅速得到推广，C 语言和 UNIX 可以说是一对孪生兄弟，在发展过程中相辅相成。

1.2 C 语言的特点

C 语言是一种通用性很强的结构化程序设计语言，它具有丰富的运算符号和数据类型，语言简单灵活，表达能力强等特点。C 语言的主要特点如下。

① 具有低级语言功能的高级语言：C 语言允许直接访问物理地址，能进行位操作，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 语言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。

② 模块化和结构化语言：C 语言用函数作为程序模块，以实现程序的模块化；C 语言具有结构化的控制语句（如 if...else 语句、while 语句、do...while 语句、switch 语句和 for 语句），语言简洁、紧凑。

③ 可移植性好：C 语言不包含依赖硬件的输入输出机制，使 C 语言本身不依赖于硬件系统，可移植性好。

④ 执行效率高：C 语言生成目标代码质量高，程序执行效率高。

1.3　C 程序结构

C 程序结构由头文件、主函数、系统的库函数和自定义函数组成，因程序功能要求不同，C 程序的组成也有所不同。其中 main 主函数是每个 C 语言程序都必须包含的部分。

1.3.1　C 程序的基本结构

由于读者刚开始接触 C 语言，在这里先不长篇论述 C 程序的全部组成部分，而是介绍 C 程序的基本组成部分。在读者会写简单的 C 程序的基础上，通过后面章节的学习逐步深入了解 C 程序的完整结构。

下面以一个简单的例子说明 C 程序的基本结构。

【例 1-1】 一个仅包含一条输出语句的简单 C 程序。

```
1    #include <stdio.h>
2    int main()
3    {
4        printf("这是我编写的第一个 C 语言程序，yeah!! \n");
5        return 0;
6    }
```

注：C 程序是没有行号的，例 1-1 程序左侧的行号（1，2，3，4，5，6）并非程序的一部分，这里的行号仅是为了对程序进行说明或叙述方便而添加的。

【运行结果】程序运行结果如图 1-1 所示。

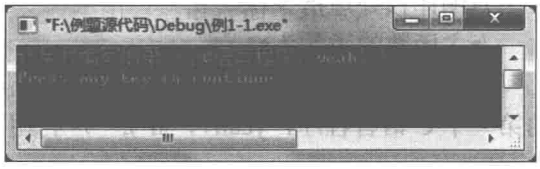

图 1-1　例 1-1 程序运行结果

【程序说明】该运行结果的第 1 行是程序运行后输出的结果，第 2 行是 Visual C++6.0 系统在输出完运行结果后自动输出的一行信息（任何一个 C 程序，只要在 Visual C++6.0 环境下运行，最后都会出现这行信息），告诉用户："如果想继续，请按任意键"。当用户按任意键后，屏幕上不再显示运行结果，而返回程序窗口。

例 1-1 的程序是由头文件和主函数组成的一个简单的 C 语言程序。

第 1 行的作用是通知 C 语言编译系统在对 C 程序进行正式编译之前需做一些预处理工作，程序的第 4 行使用了库函数 printf，编译系统要求程序提供有关此函数的信息（例如对这些输入输出函数的声明和宏的定义、全局变量的定义等），stdio.h 是 C 语言的系统文件，stdio

是“standard input & output（标准输入输出）”的缩写，.h 是文件的扩展名，它说明该文件是一个头文件（head file），这些头文件都是放在程序各文件模块的开头的。

第 2 行 int main()是函数头，其中 main 是函数的名字，表示“主函数”，main 前面的 int 表示函数的返回值是 int 类型（整型）。每一个 C 语言程序都必须有一个 main 函数。

第 3 行～第 6 行由花括号{}括起来的部分是函数体，该程序主函数的函数体由两条语句构成，每条语句后都要加分号，表示语句结束。其中 printf 是 C 编译系统提供的函数库中的输出函数，用来在屏幕上输出内容；“return 0;”的作用是当 main 函数执行结束前将整数 0 作为函数值，返回到调用函数处。

通过对例 1-1 的了解，可以看到 C 程序的结构特点如下。

① C 程序是由函数构成的，函数是 C 程序的基本单位。任何一个 C 源程序都至少包含 main 主函数，也可以包含一个 main 主函数和若干个其他函数。

② 一个函数由两部分组成：函数头和函数体。

函数头即函数的第 1 行，如例 1-1 中的 int main()。函数体即函数头下面的花括号{}内的部分。若一个函数内有多个大括号，则最外层的一对{}为函数体的范围（关于函数的组成部分参见第 6 章函数）。

③ 一个 C 程序总是从 main 函数开始执行的，而不论 main 函数在整个程序中的位置如何（main 函数可以在程序的最前头，也可以放在程序的最后头，也可以放在一些自定义函数中间）。

④ C 程序的每个语句和数据定义的最后必须有一个分号。分号是 C 语句的必要组成部分，必不可少，即使是程序中最后一个语句也应包含分号。

⑤ C 程序书写格式自由，一行内可以写多条语句，一条语句可以分写在多行上。但为了有良好的编程风格，最好将一条语句写在一行。

⑥ 一个好的、有使用价值的源程序都应当加上必要的注释，以增加程序的可读性。C 语言允许用两种注释方式。

• 以“/*”开始，以“*/”结束的块式注释。这种注释可以单独占一行，也可以包含多行。编译系统在发现一个/*后，会开始找注释结束符*/，把二者间的内容作为注释，如例 1-2。

【例 1-2】 对例 1-1 的程序加上注释。

```
#include <stdio.h>          /*编译预处理指令*/
int main()                  /*主函数的函数头*/
{                           /*函数体的开始标记*/
printf("这是我编写的第一个 C 语言程序，yeah!! \n");   /*利用库函数的输出函数在屏幕上输出指定的信息*/
    return 0;               /*main 函数的返回值是 0*/
}                           /*函数的结束标记*/
```

【程序说明】例 1-2 和例 1-1 实现的功能是完全一样的，只不过例 1-2 的可读性更好，即使不是程序的开发者，也容易明白该程序的功能。

其中“/*”和“*/”中间内容为注释内容，注释部分内容不会被编译运行，只起到解释程序语句的作用。

• 以“//”开始的单行注释。这种注释可以单独占一行，也可以出现在一行中其他内容的右侧。此种注释的范围从“//”开始，以换行符结束，即这种注释不能跨行。若注释内容一行内写不下，可以用多个单行注释。如：

```
printf("这是我编写的第一个 C 语言程序，yeah!! \n");   //利用库函数的输出函数
```

//在屏幕上输出指定的信息。

对 printf 函数所在行的程序进行注释时，一行写不下，可以在下一行接着写，但在下一行的开头必须加上“//”。

在 Visual C++编译系统中，注释可以用英文或汉字书写。

1.3.2 C 语言的算法

算法是为解决一个问题而采取的方法和步骤。

【例 1-3】 某老师讲授《C 语言程序设计》课程的某节课，他是这样来完成这节课的：拿出《C 语言程序设计》教材——研读该节课程的内容——根据掌握的该节课程的重点、难点等制作电子课件——将制作好的电子课件拷到 U 盘中——领钥匙——到教室——开始上课。

他的这一系列步骤和完成每一步采用的方法就可以称之为“算法”。在计算机科学中，算法要用计算机算法语言描述，算法代表用计算机解一类问题的精确、有效的方法。算法和程序之间存在密切的关系。

算法具有以下特点。

① 确定性 算法的每一种运算必须有确定的意义，该种运算应执行何种动作应无二义性，目的明确。

② 有穷性 一个算法总是在执行了有穷步的运算后终止，即该算法是可达的。

③ 输入 一个算法有 0 个或多个输入，在算法运算开始之前给出算法所需数据的初值，这些输入取自特定的对象集合。

④ 输出 作为算法运算的结果，一个算法产生一个或多个输出，输出是同输入有某种特定关系的量。

⑤ 有效性 要求算法中有待实现的运算都是有效的，每种运算至少在原理上能由人用纸和笔在有限的时间内完成；如若 x=0，则 y/x 是不能有效执行的。

算法的表示方法很多，通常有以下几种。

（1）用自然语言表示

自然语言表示算法可以用任何语言，比如汉语、英语、俄语等，当然也可以用数学表达式。用自然语言表示通俗易懂，但可能文字冗长，不严格，并且复杂的算法表示很不方便。所以除了简单的问题外，一般不用自然语言描述算法。

（2）用传统流程图表示

传统流程图可用一些图框和流程线来表示各种类型的操作。优点是直观形象，易于理解，缺点是传统流程图不易修改。

传统流程图常用符号如图 1-2 所示。

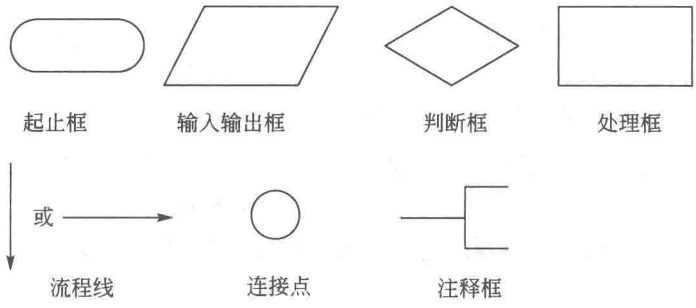

图 1-2 传统流程图的常用符号示例

（3）用 N-S 流程图表示

N-S 流程图是一种新的流程图形式。这种流程图完全去掉了带箭头的流程线，全部算法写在一个矩形框内，在该框内还可以包含其他从属于它的框。N-S 流程图适用于结构化程序设计。

（4）用伪代码表示

用流程图表示算法直观易懂，但不容易修改。伪代码可以克服流程图的这个弱点。

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法的，它不用图形符号，因此书写方便，格式紧凑，好懂，也便于向计算机程序转换。

一般在写程序之前，先列出算法，会使编程思路清晰，虽然有些简单的程序可以直接写出，但建议刚开始学习编程或写较大程序时，最好写出算法，这样助于理顺思路。

本书中的算法都用传统流程图表示。

1.3.3 C 程序的三种基本结构

C 语言程序包含三种基本结构：顺序结构、选择结构（也称分支结构）和循环结构。将这三种基本结构用传统流程图表示如图 1-3 所示。

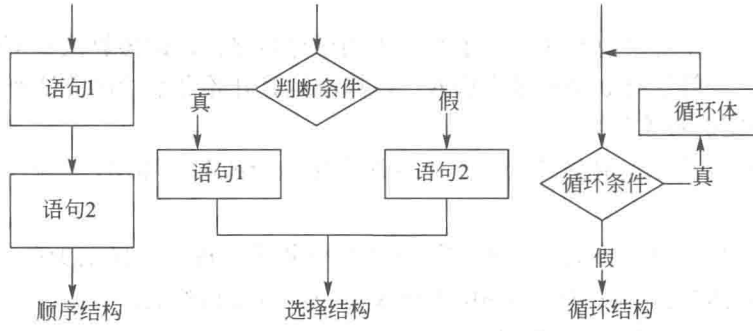

图 1-3 程序的三种基本结构

选择结构和循环结构将分别在第 3 章和第 4 章讲解。

顺序结构是最简单的一种基本结构，即顺序结构的程序一条条顺序执行。在图 1-3 的流程图中表示顺序结构中的程序执行完语句 1 接着执行语句 2。例 1-1 是顺序结构的程序，程序从第 1 行开始运行，依次执行直到程序最后一行。

1.4 C 程序的实现

1.4.1 C 程序的开发步骤

学习 C 语言就是学习编程的过程。程序是计算机的主宰，控制着计算机该去做什么事。所有要计算机做的事情都要编写程序。假如没有程序，那么计算机什么事情都干不了。

编程的第一步是“需求分析”，即要弄清楚到底想让计算机做什么。这个过程很多人都不太重视，但是忽视需求分析的结果就像考试时没有认真审题就开始答题一样，没有认真领会题目的要求，把题解得再漂亮，也得不到分数。“需求分析”在开发大型应用软件的时候，其作用尤为明显。虽然课本上讲授的题目相对较简单，但是大家最好养成好的编程习惯，别把这一步漏掉。

编程的第二步是“设计”，就是弄明白计算机该怎么做这件事。设计的内容包括两方面：

设计算法和设计程序的代码结构，使程序更易于修改、扩充、维护等。

编程的第三步是“编写程序”，即把设计的结果变成一行行代码，输入到程序编辑器中。

编程的第四步是“调试程序”，即编译源代码，变成可执行程序，运行，看是否能得到想要的结果，若不能得到想要的结果，就需要查找问题，修改代码，再重新编译、运行，直到得到正确的结果。

有的读者往往觉得把程序代码写出来就万事大吉了，其实不然，“调试程序”在整个编程过程中也很重要，特别是初学者，通过调试程序，可以掌握一些看书时忽视的问题。初学者将程序源代码写出来运行不了不要灰心丧气。试着慢慢调试程序，可能会发现，有时仅是一个分号或一个括号，导致程序运行出错。

C 语言程序是结构化的程序，是由顺序、选择、循环三种基本结构组成的。这种程序便于编写、阅读、修改和维护。

结构化程序设计强调程序设计风格和程序结构的规范化，提倡清晰的结构。其基本思路是：把一个复杂问题的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。即采取以下方法保证得到结构化的程序：①自顶向下；②逐步细化；③模块化设计；④结构化编码。

在日常生活中，每做一件事情其实也都有算法，只不过因为对做这些事情的步骤非常熟悉不用特别考虑而使人们忽视罢了。

【例 1-4】 某位同学在晚上作了如下计划：若睡觉前作业做完了就上网查资料，上完网再睡觉；作业没做完就继续做作业，直到睡觉。这件事情所对应的算法流程图如图 1-4 所示。

对例 1-4 的分析采用结构化设计方法，自顶向下，逐步细化。首先分析这位同学一共有做作业、上网、睡觉这三项大的活动，再往下细化：作业包括数学、C 语言程序设计、英语三门课程，一门一门来完成，即整体采用顺序结构完成这次作业，流程图如图 1-5 所示。

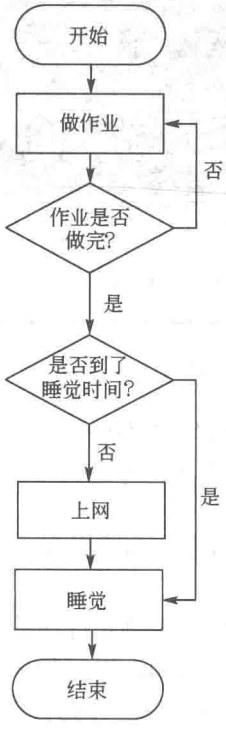

图 1-4 例 1-4 流程图

图 1-5 做作业的流程图

其中数学作业是做课后习题 1.1 和 1.2，一题做完接着做下一题即可，相当于结构化程序中的顺序结构；

C 语言程序设计作业是编写程序计算 1+2+3+4+5+6 的结果，可以有多种计算方法：如法一，顺着算式一个一个数地加，即先算 1+2 再将结果与 3 相加，再依次加上 4,5,6；法二，因为 1+6=2+5=3+4，所以可以用算式(1+6)×3 来求解。在本例中可以有两种途径完成 C 语言程序作业，若想直接计算，选法一，若嫌直接计算麻烦，先找规律再做题，选法二。相当于结构化程序中的选择结构。

英语作业是背诵课文，需要反复阅读课文，直到背下为止，相当于结构化程序中的循环结构。

通过分析，每个子流程图读者就很容易画出来了。

模块化程序设计的思想是一种“分而治之”的思想，把一个大任务分成若干个子任务，每一个子任务就相对简单了。

1.4.2 C 程序的编辑

用 C 语言编写的源程序必须经过编译、连接，得到可执行的二进制文件，然后执行这个可执行文件，最后得到运行结果。这就需要用到 C 编译系统，本书中着重介绍在 Windows 环境下使用的 Visual C++ 6.0。

首先，启动 Visual C++ 6.0，得到如图 1-6 所示的主窗口。

图 1-6 Visual C++ 6.0 主窗口

在 Visual C++主窗口的顶部是主菜单栏，包括 9 个菜单项：文件（File）、编辑（Edit）、查看（View）、插入（Insert）、工程（Project）、组建（Build）、工具（Tools）、窗口（Window）、帮助（Help）。

以上每个菜单项后的括号中是 Visual C++ 6.0 英文版中的英文显示，读者若使用的是英文版本，可以进行对照。

图 1-6 所示的主窗口的左侧是项目工作区窗口，右侧是程序编辑窗口。工作区窗口用来显示所设定的工作区的信息，程序编辑窗口用来输入和编辑源程序。

本节只介绍最简单的 C 程序的编辑情况——程序只由一个源程序文件组成，即单文件程序。

在 Visual C++主窗口的主菜单栏中选择“文件（File）”，然后在其下拉菜单中单击“新建

（New）”，如图 1-7 所示。

在弹出的“新建”对话框（见图 1-8）中，选择此对话框的左上角的“文件”选项卡，选择其中的“C++ Source File”选项，其功能是建立新的 C++源程序文件。

图 1-7　Visual C++ 6.0 的“文件”下拉菜单

在图 1-8 的右半部分的“文件名（File）”文本框中输入准备编辑的源程序文件的名字，如图中输入为“例 1-1.c”，表示要建立的是 C 源程序，这样即将进行输入和编辑的源程序就以“例 1-1.c”为文件名。

“位置（Location）”文本框中输入准备编辑的源程序文件的存储路径，如图中输入为“D:\C 示例”，表示源程序文件“例 1-1.c”将存放在“D:\C 示例”子目录下。

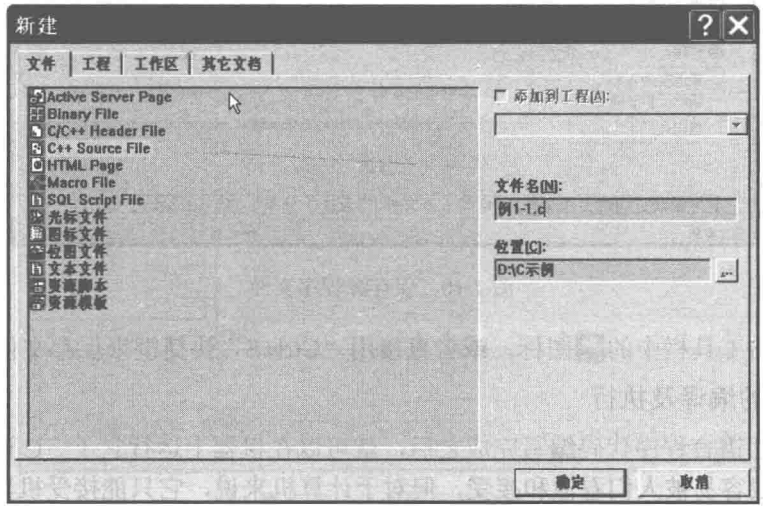

图 1-8　Visual C++ 6.0 的“新建”对话框

单击图 1-8 的“确定”按钮后，回到 Visual C++主窗口，此时光标在程序编辑窗口闪烁，表示程序编辑窗口已激活，可以输入和编辑源程序了。输入例 1-1 的程序，如图 1-9 所示。

在图 1-9 的最下面一行的椭圆形框中，显示了“行 7，列 41”，表示光标的当前位置是第 7 行第 41 列，当光标位置改变时，显示的数字也随之改变。