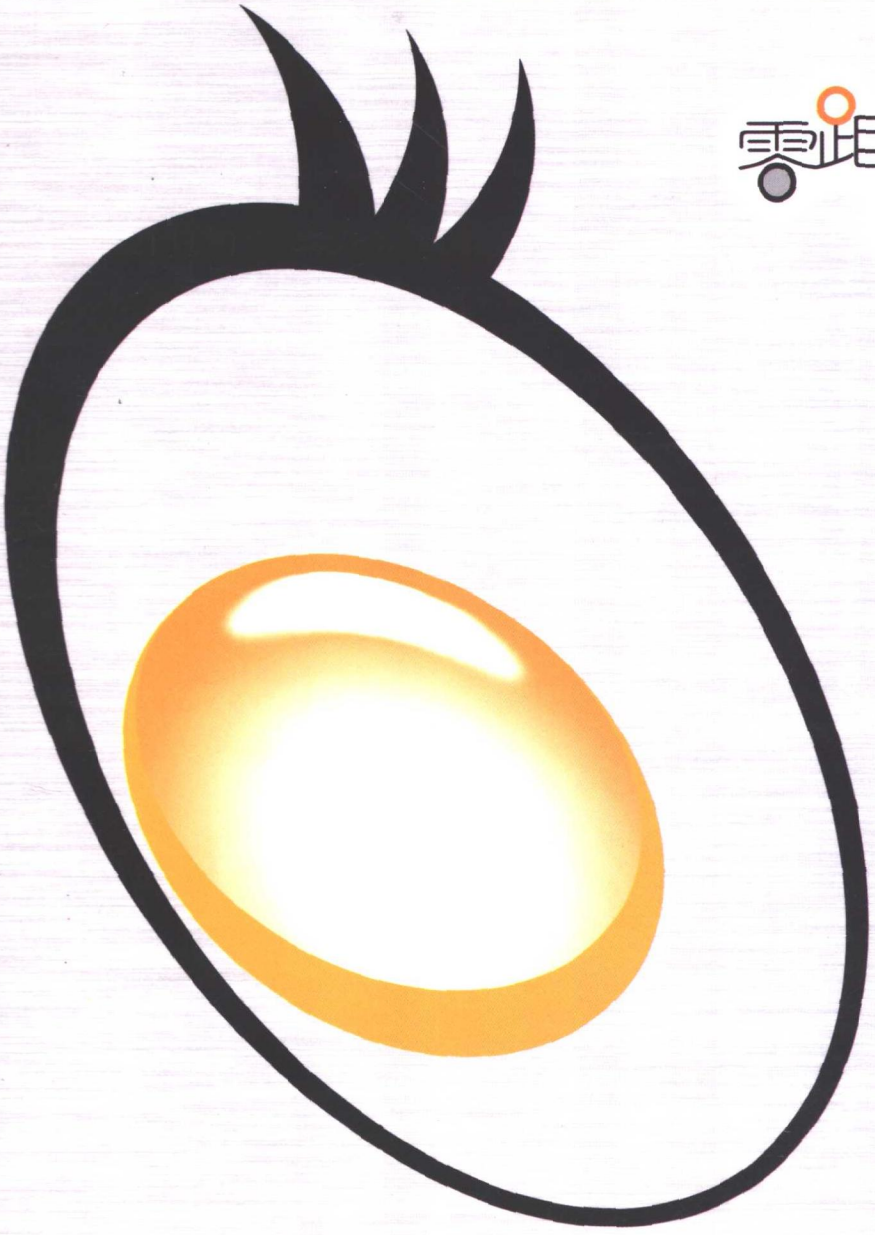
零距离 电脑培训学校
编程系列



C/C++
程序设计教程

陈 策 等编著

机械工业出版社
CHINA MACHINE PRESS

零距离电脑培训学校

编程系列

C/C++程序设计教程

陈　策　等编著

机械工业出版社

本书是系统介绍 C 与 C++语言的基础教程。全书共分 12 课，第 1 课到第 6 课主要讲解 C 语言程序设计的内容，包括 C 语言的发展历史、C 程序的特点与基本构成、C 中的数据类型、运算符和表达式、程序控制语句、函数和预编译指令等。从第 7 课开始，主要讲解 C++语言在面向对象方面扩充的内容和特性，包括面向对象方法学的基本理论、C++程序的基本构成、类与对象、函数与运算符重载、继承与多态等。第 12 课专门对 C 与 C++中的输入与输出功能进行了讲解和对比。每课除了讲解 C 与 C++必备的知识外，还有上机操作部分、常见问题解答和课后作业，以帮助读者更好地理解和掌握本单元的内容。书中的源代码和习题答案均可在 http://www.cmpbook.com 免费下载。

本书内容全面、实例丰富、叙述清晰、结构安排合理，既能作为学习 C 与 C++语言的入门书籍，又能作为深入掌握 C 与 C++的提高书籍。

本书适合电脑培训班学员及 C/C++编程爱好者使用。

图书在版编目（CIP）数据

C/C++程序设计教程/陈策等编著. —北京：机械工业出版社，2004.6
（零距离电脑培训学校编程系列）
ISBN 7-111-14426-0

Ⅰ.C… Ⅱ.陈… Ⅲ. C 语言—程序设计—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2004）第 041627 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策　　划：胡毓坚
责任编辑：车　忱
责任印制：李　妍
北京蓝海印刷有限公司印刷 • 新华书店北京发行所发行
2004 年 7 月第 1 版 • 第 1 次印刷
787mm×1092mm 1/16 • 20.75 印张 • 509 千字
0001-5000 册
定价：31.00 元

出 版 说 明

近几年来,电脑在我国迅速普及，人们的日常生活、娱乐和工作越来越离不开电脑。能够熟练使用电脑也是许多行业对从业者的基本要求。

目前，我国有大量电脑初级用户，他们迫切要求掌握电脑操作的基本方法；还有许多已有一定电脑基础知识的中级用户，希望学会使用与自身工作密切相关的软件。但是在紧张的工作之后，多数人不可能花费太多的时间来系统地学习电脑知识。基于以上背景，我社邀请国内著名计算机职业教育学校的资深老师，为电脑初、中级用户编写了这套“零距离电脑培训学校”丛书。

本套丛书紧紧围绕“短期培训”这个中心，尽量将基础知识与基本技能贯穿于基本操作和应用能力教学之中，书中列举了大量实例，鼓励读者在练中学。丛书强调“不求全、不求精、只求会”，对每一种软件或技术不要求学全，只要学会其中最重要的、与学习者的工作或专业联系最密切的内容就可以。本套丛书通俗易懂、图文并茂，便于读者迅速掌握所学知识。

为了便于读者自学以及培训班授课，我们为部分图书配了电子教案或习题答案，读者可以在我社网站（http://www.cmpbook.com）免费下载。

本套丛书覆盖了电脑应用的大部分领域。今后我们会不断补充新的图书，以满足广大读者的需求。

机械工业出版社

前　言

C 语言自诞生以来，一直受到计算机领域的研究人员、教育工作者和程序设计者的广泛关注。C 语言兼具低级语言和高级语言的特性，非常适合编写系统程序。由它开发出的程序具有代码结构简洁、执行效率高、可移植性好、维护和扩展方便等特点。因此，C 语言曾一度被誉为“真正的程序设计者的语言”。

面向对象方法是目前系统分析和程序设计领域最有效、最实用、最流行的一项技术。C++作为 C 语言的一个超集，将 C 扩展成面向对象的程序设计语言，这进一步使 C 和 C++成为目前占主导地位的编程开发语言。C/C++中的许多概念和元素都是任何一种程序设计语言所必备的。因此，学好 C/C++，不仅可以为大家进入程序员的行列打下良好的基础，而且可以为大家迅速掌握其他编程语言作好准备。

本书从 C 语言的发展历史入手，系统全面地介绍了 C 与 C++的语法元素、编程规则和实用技巧。另外，在介绍 C/C++的同时，还讲解了与它们有关的结构化分析与程序设计以及面向对象的分析与程序设计理论。

相对其他高级程序设计语言而言，C/C++在某些概念上更为难懂，也更容易出错，例如指针、动态内存分配、流等。为此，本书列举了大量实例，以帮助读者消化和吸收所学到的内容。另外，在每一单元的上机实战部分，也给出了一些针对具体应用背景的程序设计实例。所有这些实例都力求典型和易懂，从 www.cmpbook.com 可下载它们的完整代码。

要编写 C/C++程序，离不开相应的编程开发环境。目前，有许多用于 C/C++语言的集成开发环境，VC++就是其中之一。虽然 VC++的集成环境非常复杂，但我们还是选用它作为 C/C++编程开发的练习环境，以便为大家今后的进一步学习和提高打下基础。

本书在编写过程中，张江涛、刘伟、尹建民、张海霞、吴建文、智雨青、刘旭、赵磊、徐日强、范翠丽、俞雷、薛年喜、郑艳华、王建平、李士良等也参加了部分内容的编写及素材整理工作，在此一并表示感谢。

由于作者水平有限，加上时间仓促，书中错误和不当之处在所难免，希望广大读者及专家批评指正，可以将意见通过网站www.bykw.com反馈给我们，也可以与作者直接联系：pyschce@sina.com。

编　者

目　录

第 1 课 C语言概述

学习目的：

- 了解程序设计语言的发展历程
- 掌握 C 语言的特点及 C 程序的基本构成
- 了解 C 程序的编辑、编译、连接和执行过程
- 熟悉 Visual C++ 6.0 集成开发环境

学习重点：

- C 语言的特点
- C 程序的基本构成

1.1 课前导读

近些年来，随着计算机技术的飞速发展，软件开发领域涌现出了许多程序设计语言。据统计，自 1954 年第一个高级程序设计语言 FORTRAN 产生以来，共出现了 400 多种语言。这些语言有些经过进化和发展成为更加流行的新语言，有些由于固有的缺陷而逐渐退出了历史舞台，有些由于只应用于某个特定的领域而不为人们所熟知。C 语言作为第三代编程语言的代表，自产生以来，一直受到人们的重视和关注。C 语言流行很广，影响巨大。目前几乎所有大学的计算机课程中有关程序设计部分讲解的都是 C 语言，仅从这一点来看，C 语言会继续流行下去。

C 语言兼具高级语言和低级语言的特性，非常适合编写系统程序，曾被誉为是真正的程序设计者的语言。尽管近些年来，随着面向对象技术的发展，C 作为一种结构化程序设计语言已逐渐由面向对象的 C++所替代。然而 C 语言是构成 C++语言的基础，要想学好 C++语言，就必须首先学习和掌握 C 语言。另外目前许多操作系统和编译系统都是用 C 语言设计的，采用 C 语言进行程序设计可以更好地与这些系统结合。例如 Windows API 函数的描述和调用就是采用与 C 语言相似的规则。因此，对于每个有志于进入程序设计领域的人来说，C 语言是其首选。

1.2 课堂教学

1.2.1 程序设计语言的发展

任何一种计算机系统都包括硬件（hardware）和软件（software）两大部分。硬件只是

提供了计算的可能性，还必须有支持和管理计算机的软件，系统才能实现计算（computing）。从这个意义上讲，硬件是计算机的物质基础，而软件则是计算机的灵魂。

注意：

计算机中的计算并不仅仅指数值计算，所有计算机中运行的指令都可以称为计算。例如，逻辑推理、事务处理、数据管理等。

软件的一个核心部分就是计算机编程语言。在计算机发展初期，所谓的软件，就是程序。软件的设计和开发几乎是由单个的程序员所完成的。随着计算机技术的发展，传统的软件设计、开发和生产技术已不能适应信息时代的发展要求。于是计算机科学工作者将工程学的基本原理和方法引进到软件设计和生产中，形成了一门新兴的学科，即软件工程学。此时软件的内含以及外延都发生了重大的变化，软件开发已不单纯地指程序设计，它囊括了系统分析和设计、文档编写、程序设计以及文档和程序版本管理等诸多方面。尽管如此，程序设计和编码仍将是形成最终软件产品的必经之路，而程序设计和编码则一刻也无法脱离计算机编程语言。

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的发展历程。

1. 机器语言

计算机执行的指令实际上是 1 和 0 二进制数的集合，它们代表着计算机内部产生的电子信号。在计算机应用的最初十几年中，程序员主要使用计算机能够识别的二进制数来编写程序，这也是计算机惟一可以读懂的语言，一般称作机器语言。这种语言虽然十分简单，机器可以读懂，但对于程序员来说却很不方便。完成一个简单的计算公式也要编写几十条指令，编程工作枯燥而繁琐，程序冗长而难读，调试、修改和维护都是难题。另外采用机器语言编写的程序往往依赖于特定的硬件系统，可移值性很差。因此，早期的计算机应用不广，编程的专业性极强。机器语言通常被称为第一代计算机语言。

2. 汇编语言

为了减轻机器语言编程的困难，计算机专家们开始尝试用人们比较习惯的符号来代替机器的二进制串指令，例如用“ADD”来代替“001”表示加法操作，用“MOV”来代替“010”表示数据移动等等。这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。

使用汇编语言编程比使用机器语言编程要容易，另外由于汇编语言指令与机器语言指令基本上一条对一条或一条对几条，所以汇编系统的程序开发也不太复杂。因此，汇编语言编程很快取代了机器语言编程。

汇编语言和机器语言都属于低级语言，其语言结构都以面向机器的指令序列形式为主，与人的习惯语言方式距离较远。采用汇编语言编写的程序同样十分依赖于机器，移植性不好。而且程序代码冗长，不易编写大规模程序，可读性和可维护性都较差。

3. 高级语言

从最初与计算机交流的痛苦经历中，人们意识到，应该设计一种这样的语言，这种语言接近数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过努力，最早由 IBM 公司的 John Backus 领导的研究小组于 1954 年提出了一个高级语言规范“IBM Mathematical Formula Translation System”（FORTRAN）。三年后 FORTRAN 语言 1.0 版本及其编译系统正式对外公布。

提示:

由汇编语言和高级语言编写的程序，均需要通过相应的翻译系统翻译成机器语言。有些语言的翻译过程是对源程序逐行解释的，而有些语言一次翻译所有的源代码。逐行解释的程序设计语言需要通过解释程序完成源代码的翻译，而一次翻译所有的源代码的程序设计语言则通过编译程序完成翻译。

FORTRAN 语言的诞生是计算机技术发展的一个新的里程碑。在此后的 40 多年时间里，共产生了几百种程序设计语言，形成了一种百花齐放的局面。其中，FORTRAN、ALGOL、COBOL、Pascal、BASIC、Ada、Lisp 等都是影响较大的高级程序语言。

科学技术的发展永远不会停滞不前，高级程序设计语言也是如此。从 FORTRAN 语言开始，高级程序设计语言主要在应用领域的扩展和软件设计方法的改进两个方面向前发展，这两个方向既相对独立又互相影响。在应用领域的扩展方面，COBOL 语言的产生起了非常重要的作用。COBAL 语言的使用使计算机应用领域从单纯的科学计算向商业和事务处理迈出了关键性的一步，现在的数据库技术也起源于此。在软件设计思想和设计方法发展这条主线上，ALGOL 语言的设计思想对后来的程序设计语言产生了很大的影响。目前许多高级程序设计语言都直接或间接的起源于 ALGOL 语言。

随着软件的种类、规模和复杂性的增加，软件的可靠性和可维护性问题日益突出。20 世纪 60 年代在软件开发领域出现了一个名词——软件危机，该词非常形象地描述了那时软件开发和生产当中所面临的难题。为了解决软件危机问题，人们提出以系统工程的观点来对待软件设计和开发，从而产生了一些成熟的软件设计方法和程序设计语言。1969 年，结构化的程序设计方法被提出，1970 年，第一个结构化的程序设计语言——Pascal 语言出现。20 世纪 80 年代初，面向对象的程序设计（OOP）思想产生，其中 C++语言就是面向对象程序设计语言的最典型的代表。

1.2.2 C 语言的产生

C 语言是由贝尔（Bell）实验室的 Dennis Ritchie 在 1972 年开发出来的，它是一种结构化的程序设计语言。C 语言源于 Ken Thompson 的 B 语言，而 B 语言又源自 BCPL（Basic Combined Programming Language）语言，该语言由 Martin Richards 所开发。C 语言最早的前身实际上就是对程序设计思想产生重要影响的 ALGOL 语言，它于 20 世纪 60 年代早期由国际委员会设计和开发。此后，由伦敦和剑桥大学合作开发的 CPL（Combined Programming Language）语言对 ALGOL 进行了改进，使其能够直接作较低层次的操作。然而，跟 ALGOL 语言一样，CPL 语言有太多的特性，很难理解而且实现起来很困难，因此 BPCL 语言随之

产生。C 语言的发展历史可以归纳为表 1-1。

表 1-1 C 语言发展历史

语 言	时 间	开 发 者
ALGOL	20 世纪 60 年代早期	国际委员会
CPL	1963 年	伦敦和剑桥大学
BCPL	1967 年	剑桥大学的 Martin Richards
B	1970 年	贝尔实验室的 Ken Thompson
C	1972 年	贝尔实验室的 Dennis Ritchie

C 语言最初是在 UNIX 操作系统上开发的，此后出现了许多 C 语言编译和集成工具，这些工具在源程序水平上大都能够很好地兼容。然而，由于没有统一的标准，这些工具之间也有不一致的地方。为了改变这种状况，美国国家标准协会（ANSI）于 1983 年制定了 ANSI C 标准。此后，所有的 C 语言工具都遵从这个标准，只要编写符合 ANSI 标准的 C 语言源代码，就可以保证在这些工具中的通用性。

1.2.3 C 语言的特点

C 语言产生的背景以及 C 语言设计者的初衷决定了 C 语言具有许多不同于其他程序设计语言的特点。多年来，C 语言经久不衰并不断进化和发展，完全得益于它的以下特点：

（1）C 语言兼具低级语言和高级语言的特性。C 语言具有允许直接对位、字节和字进行操作的特点；C 语言所提供的指针数据类型使其具有寻址内存区域的能力。而且，C 语言具有与汇编语言的透明接口，在 C 语言中可以方便地插入汇编语言从而使程序达到最高的效率。C 语言的这种低级语言的特点使其非常适合写系统程序，Dennis Ritchie 也正是基于此而开发 C 语言的。

C 语言像 Pascal、BASIC 等高级语言一样支持数据类型的概念。在 C 语言中定义了 5 种基本的数据类型，同时还允许程序员灵活地定义其他数据类型，这极大地扩展了 C 语言的灵活性。

（2）C 语言是一种编译语言。几乎所有的 C 语言开发工具都将 C 程序通过编译系统翻译成机器代码，从这个角度上讲 C 语言是一种编译语言。编译语言与解释语言的不同点在于编译语言只需通过一次翻译即可永久使用生成的目标代码，而解释语言每次执行时均需要通过解释程序一行行地解释。编译过程虽然占用一定的时间，但运行时的效率却比解释语言要高得多。

（3）C 语言具有最佳的代码规模和简洁的关键字集。C 语言比任何其他高级语言都具有更少的语法规则，它正是从复杂的 ALGOL 语言开始一步步地简化而来的。C 语言精炼的语法结构使得 C 语言编译器可以做得非常小，而且质量高。

ANSI C 标准所推荐的关键字只有 32 个。其他语言所具有的输入/输出功能、字符处理和算术操作均不包含在基本的 ANSI 标准中，C 语言的这些功能均通过函数库来提供。

（4）C 语言具有丰富的函数库和扩展功能。几乎所有的 C 编译器都提供了许多功能强大的函数库。这些函数库对图像、文字处理、数据库支持、屏幕窗口化、数据输入等功能

的支持都是非常有用的。程序设计者也可以设计自己的函数库，从而进一步扩展 C 语言的特性。

（5）弱类型检查和边界检查。与其他高级语言相比，C 语言的类型检查机制非常弱。程序员可以在不同的数据类型之间相互转换而不会发生编译错误，这既是 C 语言的一个优点，也是它的一个缺点。弱类型检查意味着在编译源代码时无法有效地检查错误，从而导致在运行时发生不希望的数据处理，程序员必须自己来保证类型转换的正确性。弱类型检查的优点是它赋予 C 语言极大的操作数据的灵活性，程序员完全可以从中受益，例如将一个字符（char）直接赋给整型（int）变量从而得到其 ASCII 码的数值表示。

C 语言不对数组以及内存的边界进行检查，这使得在运行中许多隐秘的、暂时的错误不能够被发现。运行中数组的超界以及内存的非法使用所产生的问题由于不能立即被发现，而导致错误进一步向下扩展，给程序的排错带来困难。

（6）模块化的结构和自顶向下的设计实现。C 语言是一种结构化的程序设计语言，它具有当今高级语言所希望具有的所有的控制结构。结构化程序设计语言的特点可以让用户很自然地采用自顶向下的规划和设计，从而产生可靠的、容易理解和维护的程序。

C 程序的模块化结构不仅体现在源代码的设计实现上，还体现在 C 语言编译器中。几乎所有的 C 语言编译器都支持分块编译，这样就可以只对开发过程中改变过的程序进行编译，这一特性在开发大程序时具有非常重要的意义。

（7）较好的可移植性。可移植性（portability）就是指运行在某一计算机或操作系统中的程序可以在另一台计算机或操作系统中很好地运行。用 C 编写的程序在现代计算机领域内几乎是移植性最好的。这在小型和微型计算机世界中显得尤其突出。

1.2.4 C 程序的基本构成

为了说明 C 语言程序的基本结构，下面给出一个简单的 C 程序例子。

实例 1-1　C 程序构成实例：

```
/*example.c，作者：陈策，时间：2003 年 9 月 7 日*/
#include <stdio.h>

int AddTwo(int a); /*自定义函数声明*/
void main() /*主函数*/
{
    int b; /*变量定义*/
    b=AddTwo(4); /*调用自定义函数，结果返回给 b*/
    printf("Hello!The result is %d\n",b); /*调用屏幕输出函数*/
}
int AddTwo(int a) /*定义 AddTwo 函数*/
{
    int b; /*定义函数内部使用的变量*/
    b=a+2; /*将 a 值加 2 后赋给 b*/
    return b; /*将 p 值返回调用处*/
```

```
}
```

运行这段程序后，在屏幕上将显示一行信息：

```
Hello!The result is 6
```

程序的第一行为注释，编译时不起作用。在 C 语言中，注释由/*和*/括起来，注释的内容可长可短，用来对程序加以说明。注释并不影响编译后的目标文件，对程序的执行不起任何作用。在程序中加注释是一个良好的习惯，这样的程序更容易读懂，于己于人都十分有益。

程序的第二行为#include <stdio.h>，其中#include 是 C 编译器的一个编译指令，stdio.h 是 C 标准库的一个头文件，<>指示编译系统在系统设定的目录中搜索头文件。这行的作用是指示 C 编译器将文件 stdio.h 的内容插入到程序中#include 指令所在的这一行的后面，这使得程序可以使用在文件 stdio.h 中定义的标准输入和输出函数。只有这样，语句：

```
printf("Hello!The result is  %d\n",b);
```

才能顺利通过编译。

之后是一个空行，在 C 语言中空行没有任何意义，只是为了程序代码的直观。

第二个代码行声明了一个自定义函数 AddTwo()，该函数的定义在程序的尾部。

程序中以 main 开始的部分定义了一个函数，该函数规定了该程序的功能。main 是函数名，在函数名之后紧跟一对圆括号。所有的 C 程序都必须有一个名为 main 的主函数，程序执行的开始点是 main 函数中的第一条语句。每条语句均以分号（;）表示结束。

一个 C 函数中的任何成份被括在一对花括号（“{”和“}”）中，左花括号表示“这个函数从这里开始”，右花括号表示“这个函数在这里结束”。花括号括起来的部分称作函数体，而函数名和它后面的一对圆括号称为函数头。函数体由一系列 C 语句组成，这些语句描述这个函数怎样实现它的功能。

主函数 main()中首先定义了一个整型变量，然后调用用户自定义函数 AddTwo()，并将参数 4 送给 AddTwo()函数中的形式参数 a。AddTwo()函数执行后，将参数 4 加 2 后返回。

屏幕上输出的信息是由标准输出函数 printf()完成的。函数 printf 中的第一个参数为一个字符串，用以控制所显示的内容。字符串中的最后一个字符\n 是一个控制字符，称为换行符，用于告诉计算机重新开始新的一行。字符串中的%d 是输出的格式说明，输出时将由函数中的第二个参数替代。

从上例中可以看出，C 程序为函数模块结构，所有的 C 程序都是由一个或多个函数构成，其中只能有一个主函数 main()。程序从主函数开始执行，当执行到调用函数的语句时，程序将控制转移到调用函数中执行，执行结束后，再返回主函数中继续运行，直至程序执行结束。C 程序的函数是由编译系统提供的标准函数（如 printf）和由用户自己定义的函数（如 AddTwo）。虽然从技术上讲，主函数不是 C 语言的一个成分，但它仍被看做是其中的一部分，因此，“main”不能用作变量名。

当编写大型的程序时，通常并不将所有的源程序都集中在一个文件中，而是按照模块划分的原则，将具有相关功能的函数进行分组，形成一个个独立的文件。将函数的定义放

在扩展名为“.c”的文件中，而将函数的声明放在扩展名为“.h”的文件中。扩展名为“.c”的文件称为 C 程序源文件，而“.h”文件则称为头文件。主函数部分可以单独放在一个文件中，也可以和其他函数放在一起。包含主函数的源文件称为主文件。所有的源文件（.c）通过工程文件组合在一起，头文件（.h）用#include 预编译指令插入到某个需要该头文件的源文件中。可以将头文件（.h）组合在工程文件里，也可以不这么做。

提示：

工程文件在不同的 C 语言集成环境中，其格式可能会有所不同，扩展名当然也有差异。甚至有些 C 集成环境中的工程文件还不是文本文件，但这无需担心，因为在这些集成环境中都提供了很好的工程文件制作工具，用户可以利用这些工具方便地生成和维护工程文件。

例 1-2 是将例 1-1 的 example.c 文件中 AddTwo 函数的定义和声明摘出而形成的。AddTwo.h、AddTwo.c 和 example.c 文件是由用户编写的源代码文件，通过某个 C 集成开发环境将这些文件组合成一个工程文件，然后编译和连接，执行的结果与例 1-1 完全相同。

实例 1-2　C 程序的文件构成：

AddTwo.h 头文件代码如下：

```
int AddTwo(int a);
```

AddTwo.c 源文件代码如下：

```
int AddTwo(int a)
{
int b;
      b=a+2;
      return b;
}
```

example2.c 主源文件代码如下：

```
#include <stdio.h>
#include "AddTwo.h"
void main()
{
int b;
b=AddTwo(4);
printf("Hello!The result is  %d\n",b);
}
```

1.2.5　C 程序的编辑、编译、连接和执行

如图 1-1 所示，编写一个 C 程序项目，一般需要经过以下几个步骤：