21世纪高等学校计算机科学与技术规划教材

CYUYAN CHENGXUSHEJI

C语言程序设计

郑丽英◎主编

北京邮电大学出版社
www.buptpress.com

21世纪高等学校计算机科学与技术规划教材

C语言程序设计

主　编　郑丽英

副主编　杨景玉　王　松

孟昱煜　王坚生

北京邮电大学出版社

·北京·

内容简介

本书是作者在总结数十年来从事C语言教学和科研的丰富经验的基础上，全面系统地介绍了C语言的使用方法以及提示了在使用中容易出现的问题。本书包含了C语言的运算符、表达式、语句、函数、预处理、指针、结构体以及文件操作等内容。本书在讲述上具有重点突出、难点举例详细介绍、前后连贯等特点。每章有各种类型的练习题。本书语言介绍通俗、概念描述精准，既简单易懂也具备了有深度的内容。

本书可作为高等院校本科和大专学生的教材，也适合科研人员入门学习，亦可作为成人教育、自学考试等的教材和参考书目。

图书在版编目（CIP）数据

C语言程序设计 / 郑丽英主编．—北京：北京邮电大学出版社，2012.1
ISBN 978-7-5635-2903-2

Ⅰ．①C… Ⅱ．①郑… Ⅲ．①C语言-程序设计 Ⅳ．①TP312

中国版本图书馆CIP数据核字（2012）第005634号

书　　名 C语言程序设计
主　　编 郑丽英
策 划 人 马　飞
责任编辑 苏文刚
出版发行 北京邮电大学出版社
社　　址 北京市海淀区西土城路10号（100876）
电话传真 010—82333010　62282185（发行部）　010—82333009　62283578（传真）
网　　址 www.buptpress3.com
电子信箱 ctrd@buptpress.com
经　　销 各地新华书店
印　　刷 北京联兴华印刷厂
开　　本 787 mm×1 092 mm　1/16
印　　张 19
字　　数 459千字
版　　次 2012年1月第1版　2012年1月第1次印刷

ISBN 978-7-5635-2903-2　　定　价：34.50元

前　言

C语言是目前实际使用最广的语言之一，它包含了需要理解的基本机理和主要机制，能满足讨论程序与程序设计问题的需要。学生可以用它完成程序练习，得到有关知识积累和能力锻炼，还能掌握一种实用的工具。另一方面，程序设计是计算机领域的基础课，学了C语言后，学习操作系统等后续课程比较容易。此外，许多语言从C语言借鉴了想法和形式，C语言知识对于进一步了解其他语言，包括未来的新语言都有价值。

本书的主要目标是能让没学过程序设计的初学者能尽快掌握这门语言，同时能学会计算机解决问题的方法，碰到问题的时候可以着手进行解决。另外，本书也给出了隐藏在很多问题后面的本质，让初学者不至于走入误区。

本书始终强调一种观点：程序问题并没有需要记住的标准答案。由于分析问题时的不同考虑，设计过程中的不同决定或选择，人们对同一问题常会得到许多合理而正确的不同程序。这些程序常各有长短，可能各有侧重点，也可能反映了对问题的不同认识。应该特别注意思考和解决问题的方法，包括如何分析问题，逐步把问题弄得更清楚明确；如何寻找可能求解途径，把复杂问题分解为相对简单的步骤；如何在可用的语言结构中做出选择等。这里的每一步都可能产生分支，应该认清各种选择的后果，无论是收获还是损失。

我们希望读者能建立一种认识：程序设计问题并没有什么“标准答案”，要追求的是比较好的“正确”答案，各种书籍中给出的答案（包括本书）不过是作者对问题理解和分析的反映，还有很多可能选择。进一步说，我们还希望读者养成这样的习惯：在看别人的程序时，应分析其中隐含着作者的哪些考虑和选择，哪些是合理的有价值的（或不合理没价值的）？还可能有什么选择？沿其他选择走下去可以得到什么（或失去什么），等等。这种思考习惯必将使人受益无穷。当然，这并不是说书中的示例不重要。恰恰相反，因为程序设计中有许多选择可能，书中更应当给出合理的、好的选择供读者参考。入门书籍还应当说明理由，如果可能，还应当指出采取这些选择带来的问题（缺点、限制等）。

在撰写本书时，我心中有几个努力追求的目标，列在这里供读者和同行参考：

1. 没有程序设计经验的初学者，希望本书适合作为入门课教材。

2. 以程序设计为基本线索，同时对C语言做深入介绍，强调如何认识程序、写程序和用C语言解决实际问题。不少实例给出了在不同考虑下可能形成的多种解法，以帮助读者理解程序设计的真谛。

3. 强调好的程序设计风格，通过函数抽象建立起清晰结构的重要性。尽可能早地引进函数概念，从库函数、简单函数定义等。强调程序的结构性、可读性、易修改性。书中还根据进展和遇到的问题分析了一些不良程序设计习惯及其危害。

本书力图坚持这种正确途径，讨论了如何写出更可靠、不易包藏隐含错误的C语言程序的各方面问题，通过实例说明了应该如何写和不应该如何写。在此基础上也介绍了一些实用的C语言程序设计技术。本书希望强调的是如何写正确、清晰、简洁、高效的C语言

程序。

4. 详细介绍和解释了C语言的各种结构和机制，因为其中不少问题反应了相关领域的普遍性知识和情况。这里还尽可能提供一些背景理由，以帮助读者理解问题实质。对语言和程序设计的一般性问题及计算机科学中的一般问题也有适度介绍。

本书主要包含以下内容：

第1章，C程序设计语言概述。介绍程序与程序语言的概念，C语言的发展及其特点。用一个小例子介绍C程序形式，其加工和执行。最后介绍程序设计与开发过程。

第2章，使用计算机解决问题的过程。介绍了计算机是如何通过运算来解决问题的，如何从自然语言过渡到流程图，再最终到C语言的程序。

第3章，数据类型及表达式。讨论程序语言的许多最基本概念，包括：字符集、标识符和关键字、数据与类型、数据表示、运算符、表达式与计算过程等。

第4章，顺序结构。介绍了如何编写简单的C语言程序。

第5章，分支结构。主要是if系列语句和switch语句。

第6章，循环结构。主要包括各种循环的构成方式，while语句，do-while语句等。

第7章，数组。主要有数组的概念、使用等。

第8章，函数。介绍函数的概念、参数的概念、参数传递的方式及使用中注意的事项等。

第9章，指针。指针概念和指针变量的使用，C语言中指针与数组的关系，多维数组作为参数的通用函数。而后讨论了动态存储管理，类型定义，指向函数的指针等一系列问题。最后讲述了typedef如何把复杂的定义简单化。

第10章，编译预处理。包括宏定义、文件包含、条件编译。

第11章，结构及其他数据机制。介绍结构（struct）、联合（union）、枚举（enum）等数据定义机制的意义及在程序中的使用。同时也包括了位操作运算符，也介绍了链表的概念。

第12章，文件。讨论了文件概念，与输入输出有关的各种问题，标准库的输入输出功能，以及输入输出的程序设计问题。

最后有几个附录和一个比较详细的索引。

本书以标准C语言为背景，有关内容不依赖于具体C语言系统。读者可用任何符合ANSI C标准的C语言系统作为编程环境。虽然软件厂商不断推出新版本的程序开发环境，但从学习基本程序设计的角度看，新开发环境比早先版本并没有实质性改善，通常是更复杂、占用更多计算机资源，初学者入门更困难。本书的实例程序都在Visual C++6.0下调试通过，在其他环境中也能运行，但是像Turbo C这样比较老的环境可能和本书的输出结果不同。

本书第3、4、5、9章由杨景玉编写，第7、8、11、12章由王松编写，第1、2、6章及附录4由孟昱煜编写，第10章及附录1、2、3由王坚生编写。本书由郑丽英统稿定稿。

由于时间仓促，作者水平有限，书中难免有错误和疏漏之处，敬请读者批评指正。

编　者

目 录

第1章　C程序设计语言概述

1.1 冯·诺依曼结构

说到计算机的发展，就不能不提到德国科学家冯·诺依曼（见图1-1）。从20世纪初，物理学和电子学科学家们就在争论制造可以进行数值计算的机器应该采用什么样的结构。人们被十进制这个人类习惯的计数方法所困扰。所以，那时以研制模拟计算机的呼声更为响亮和有力。20世纪30年代中期，德国科学家冯·诺依曼大胆的提出，抛弃十进制，采用二进制作为数字计算机的数制基础。同时，他还说预先编制计算程序，然后由计算机来按照人们事前制定的计算顺序来执行数值计算工作。

图1-1　冯·诺依曼

冯·诺依曼理论的要点是：数字计算机的数制采用二进制；计算机应该按照程序顺序执行。

人们把冯·诺依曼的这个理论称为冯·诺依曼体系结构（见图1-2）。从ENIAC到当前最先进的计算机都采用的是冯·诺依曼体系结构。所以冯·诺依曼是当之无愧的数字计算机之父。

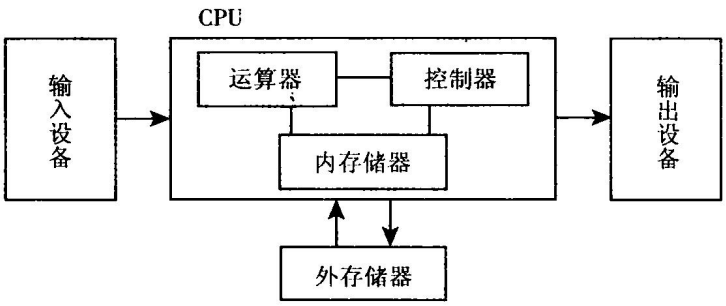

图1-2　冯·诺依曼体系结构

根据冯·诺依曼体系结构构成的计算机，必须具有如下功能：把需要的程序和数据送至计算机中。必须具有长期记忆程序、数据、中间结果及最终运算结果的能力。能够完成各种算术、逻辑运算和数据传送等数据加工处理的能力。能够根据需要控制程序走向，并能根据指令控制机器的各部件协调操作。能够按照要求将处理结果输出给用户。

为了完成上述的功能，计算机必须具备五大基本组成部件，包括：输入数据和程序的输入设备、记忆程序和数据的存储器、完成数据加工处理的运算器、控制程序执行的控制器、输出处理结果的输出设备（见图1-3）。

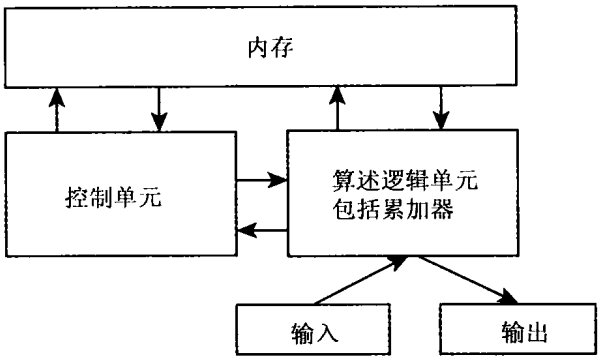

图 1-3　冯・诺依曼计算机基本组成部分

冯・诺依曼的思想具体如下：

1）采用存储程序方式，指令和数据不加区别混合存储在同一个存储器中，数据和程序在内存中是没有区别的，它们都是内存中的数据。指令和数据都可以送到运算器进行运算，即由指令组成的程序是可以修改的。“指令”指的是让计算机进行某种操作的命令，比如让计算机完成加法，那么执行“ADD 指令”，“数据”指的是需要加工处理的数，比如加法的 2 个加数。

2）存储器是按地址访问的线性编址的一维结构，每个单元的位数是固定的（见图 1-4）。

地址	0000	0001	0002	0003	0004	…
存储器单元						…

图 1-4　存储器的编址方式

3）指令由操作码和地址组成。操作码指明本指令的操作类型，地址码指明操作数和地址。操作数本身无数据类型的标志，它的数据类型由操作码确定。比如“加法第一个加数第二个加数”。

4）通过执行指令直接发出控制信号控制计算机的操作。指令在存储器中按其执行顺序存放，由指令计数器指明要执行的指令所在的单元地址。指令计数器只有一个，一般按顺序递增，但执行顺序可按运算结果或当时的外界条件而改变。

5）以运算器为中心，I/O 设备与存储器间的数据传送都要经过运算器。

6）数据以二进制表示。

从本质上讲，冯・诺依曼体系结构的本征属性就是两个一维性，即一维的计算模型和一维的存储模型。冯・诺依曼机存储器是一维的线性排列的单元，按顺序排列的地址访问。

根据冯・诺亦曼结构，我们要完成 10＋20 必须采用下述方式：

1）将混合有 2 个数据 10，20 和控制代码的程序存放到存储器，这样的话，数据 10，20 及控制代码这三个进行加法的要素都在存储器中特定的位置，也就是说地址是确定的。

2）通过数据总线将 10 和 20 放入 CPU 的寄存器。

3）执行加法操作，CPU 根据控制总线传递的信号，进行加法操作。

4）将 10＋20 的结果存放到特定位置。

这样，用助记符或者说是汇编语言表示，那么就是如下的代码：

MOV AX 10 ；将 10 这个数放在名字为 AX 的寄存器中。

MOV BX 20 ；将 20 这个数放在名字为 BX 的寄存器中。

ADD AX BX ；进行加法操作，运算的结果还存放在 AX 中。

很明显，按这种方式使用计算机是十分不方便的。

要使计算机做为一个强大的工具，那么我们必须得按照计算机能理解的语言来给计算机布置任务，上述的汇编语言就是和计算机交流的一种语言之一，布置的任务一般是比较精准的，所有这些指令的集合我们一般叫做“程序”。但是，这种语言除了计算机方面的专家外，其他人很难掌握。

随着时间的发展，人们发明了比汇编语言更容易使用的语言，一般称为“高级语言”，“高级语言”更接近自然语言，在给计算机布置任务的时候也更加方便，也就是说，高级语言的程序更加容易编写，同时也更加容易阅读。

C 语言是最常用的高级语言之一。

1.2 C 语言的发展历史

世界上第一个高级语言是“ALGOL 语言”，也叫算法语言。第二个高级语言是“FORTRAN 语言”，也叫公式翻译语言。以后陆续出现了很多种高级语言，在 20 世纪 90 年代之前使用比较广泛的有 BASIC 语言、COBOL 语言、FOXBASE 语言、PASCAL 语言和 C 语言。

C 语言的发展历史可以简单描述如下：

C 语言的原型 ALGOL 60 语言（也称为 A 语言）。

1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL（Combined Programming Language）语言。

1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，于是产生了 BCPL 语言。

1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改，并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言煮干，提炼出它的精华。并且他用 B 语言写了第一个 UNIX 操作系统。

而在 1972 年，B 语言也给人“煮”了一下，美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。

为了使 UNIX 操作系统推广，1977 年 Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。

1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的《*The C Programming Language*》一书。通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的 C 语言标准，后来由美国国家标准化协会（American National Standards Institute，

ANSI）在此基础上制定了一个C语言标准，于1983年发表。通常称之为ANSI C。

1987年，随着微型计算机的日益普及，出现了许多C语言版本。由于没有统一的标准，使得这些C语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准化协会为C语言制定了一套ANSI标准——87 ANSI C，成为被广泛接受的C语言标准。

1990年，国际标准化组织（International Standard Organization，ISO）接受了87 ANSI C为ISO C的标准（ISO9899－1990）。1994年，ISO修订了C语言的标准。

在ANSI C标准确立之后，C语言的规范在很长一段时间内都没有大的变动。1995年WG14小组对C语言进行了一些修改，成为后来的1999年发布的ISO/IEC 9899：1999标准，通常被称为C99规范，但是被认可的程度不及87 ANSI C。

目前在微机上使用的C编译程序有：Turbo C、Watcom C、GNU C以及微软的Visual C++等。各个编译器一般都以“ANSI C”为标准，但是具体实现上有轻微的差别，本书将使用微软的Visual C++ 6.0编译器编译示例程序。对于一个初学者，Microsoft Visual C++ 6.0是一个很好好的软件。界面友好，功能强大，调试也很方便。

1.3 C语言的特点

C语言从正式被发表到目前为止，已经40多年了，计算机技术的发展一日千里，速度非常快，由于新的需要，各种新的编程语言也被设计和实现出来了。好像C语言已经老态龙钟，不能适应新的要求了，但是，由TIOBE进行的编程语言使用排名中，C语言一直占据前几名的位置，比如，在2010年12月的调查中，C语言依然排名第二。由此可以看出，C语言是一门有着多么强大生命力的语言。究其原因，主要是C语言有着以下特点：

（1）C语言通常称为中级计算机语言

中级语言是相对于高级语言而言，并没有贬义，不意味着它功能差、难以使用、或者比Pascal或者Java那样的高级语言原始，也不意味着它与汇编语言或者机器语言相似，会给使用者带来类似的麻烦。C语言之所以被称为中级语言，是因为它把高级语言的易于使用的特点和汇编语言强大的功能这两个优点结合起来了。

（2）C语言的另一个重要特点是它仅有32个关键字

这些关键字相当于C语言所有的可以使用的命令。和IBM PC的BASIC语言相比，BASIC包含的关键字达159个之多（随版本不同而有所差异）。

（3）C语言是结构化语言

结构化语言的显著特征是代码和数据的分离。这种语言能够把执行某个特殊任务的指令和数据从程序的其余部分分离出去、隐藏起来。让任务可以分割的一个方法是调用使用局部（临时）变量的子程序。通过使用局部变量，我们能够写出对程序其他部分没有副作用的子程序。这使得编写共享代码段的程序变得十分简单。如果开发了一些分离很好的函数，在引用时我们仅需要知道函数做什么，不必知道它如何做。

结构化语言比非结构化语言更易于程序设计，用结构化语言编写的程序的清晰性使得它们更易于维护。这已是人们普遍接受的观点了。C语言的主要结构成分是函数C的独立子程序。

（4）运算符丰富

C 的运算符包含的范围很广泛，共有 34 种运算符。C 把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

（5）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，使用起来比 PASCAL 更为灵活、多样。

（6）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做检查，由程序编写者自己保证程序的正确。对变量的类型使用比较灵活，例如整型数据与字符型数据可以通用。一般的高级语言语法检查比较严，能检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度，因此，放宽了语法检查。程序员应当仔细检查程序，保证其正确，而不要过分依赖 C 编译程序去查错。“限制”与“灵活”是一对矛盾。限制严格，就失去灵活性。而强调灵活，就必然放松限制。一个不熟练的编程人员，编一个正确的 C 程序可能会比编一个其他高级语言程序难一些。也就是说，对用 C 语言的人，要求对程序设计更熟练一些。

（7）C 语言的功能

C 语言能进行位（bit）操作，能实现汇编语言的绝大部分功能，可以直接对硬件进行操作。因此 C 既具有高级语言的功能，又具有低级语言的许多功能，可用来写系统软件。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。

（8）C 语言具有良好的可移植性

目前，C 语言的编译器在大型机、小型机、微型机、单片机等各种不同类型的计算机上都有实现。

1.4 C 语言的应用场合

最初，C 语言被用于系统程序设计。一个“系统程序”是一大类程序的一部分，这一大类构成了计算机操作系统及实用程序。通常被称为系统程序的有：

（1）操作系统

（2）翻译程序

（3）编辑程序

（4）汇编程序

（5）编译程序

（6）数据库管理程序

（7）直接和硬件交互的程序，比如单片机的程序或者嵌入式系统的程序

1.5 C语言和C++的关系

1983年，贝尔实验室的 Bjarne Strou strup 博士推出了C++（C Plus Plus）语言。C++语言是一种面向对象的程序设计语言。C++目前流行的最新版本是 Borland C++，Symantec C++、GNU C++和 Microsoft Visual C++等。

C++语言是为了面向对象程序设计而生的，虽然它语法上兼容C语言，即所谓的C语言的“超集”，但是所提倡的编程方式和C有本质的区别，C语言的编程方式一般是结构化程序设计。

1.6 第一个C程序

为了让C语言的程序能够被计算机识别并理解，我们需要一个程序，这个程序叫做“编译程序”，同时，为了以更加方便和友好地方式输入C语言的程序，我们还需要一个工具，这些工具的集合叫做“集成调试环境”（Integrated Development Environment 简称IDE），我们选用的IDE是微软的 Microsoft Visual C++ 6.0。关于 Microsoft Visual C++ 6.0的安装和详细使用方法，参照附录1。

下面我们先来看第一个C程序。

1）首先打开 Microsoft Visual C++6.0，然后选择菜单“文件”→“新建”→“工程”，出现如图1-5所示的对话框。

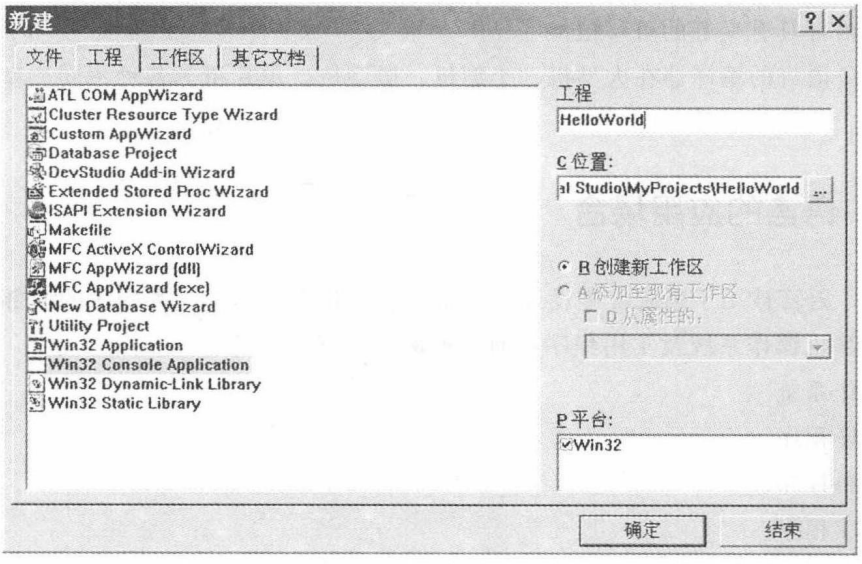

图1-5 新建项目

选择“工程”，然后在右侧选择“Win32 Console Application”（控制台应用程序），在“工程”下面的输入框中输入第一个程序的名称“HelloWorld”（斜体以后表示需要我们输入的内容）。单击“确定”按钮。

随后出现如下对话框。

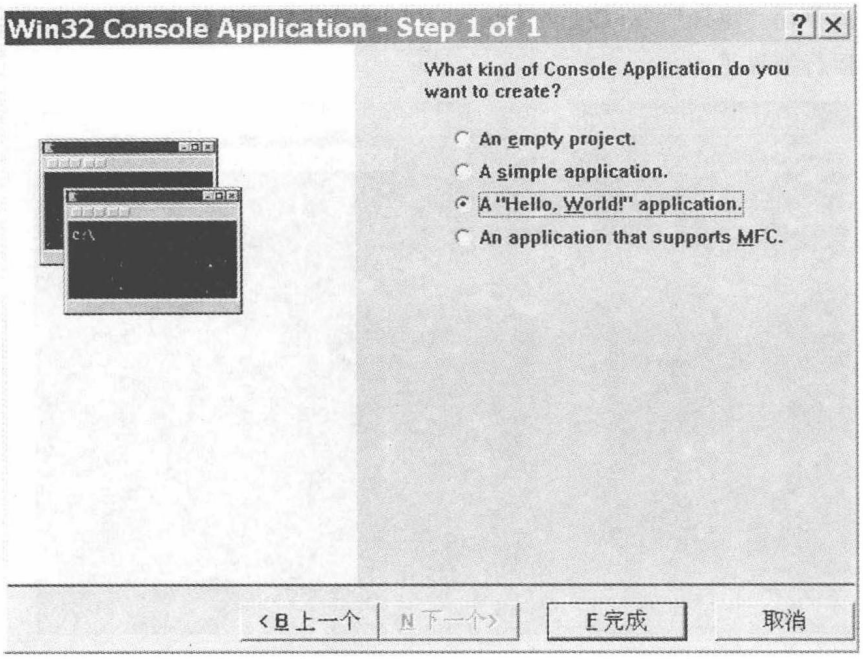

图 1-6　确认项目类型

2）然后单击“完成”按钮。

请注意，不要修改任何选型。

在随后出现的对话框中点击“确定”。出现如图 1-7 所示窗口，如和下图不同，请先选择菜单“查看”→“工作区”，然后点击工程管理器（WorkSpace）下侧的 Fileview，再展开 Source Files，双击“HelloWorld. cpp”。

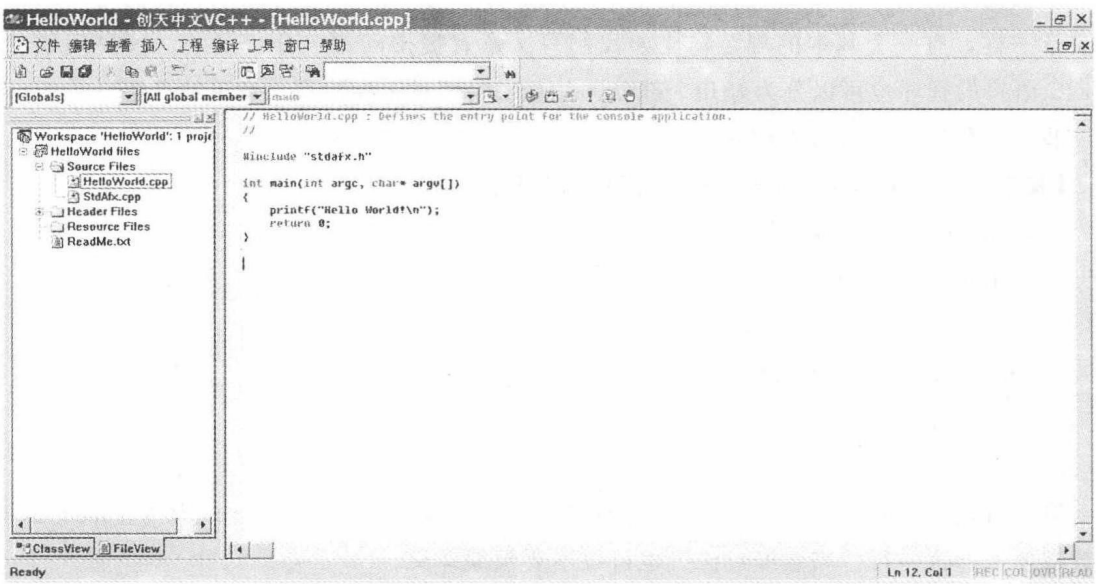

图 1-7　Microsoft Visual C++ 6.0 初始界面

这样，一个简单的C语言程序就在工程向导的帮助下完成了，我们可以开始运行这个程序了。选择菜单“编译”“→执行 HelloWorld.exe”，在出现的对话框中选择“是”，可以看到程序的运行结果了。

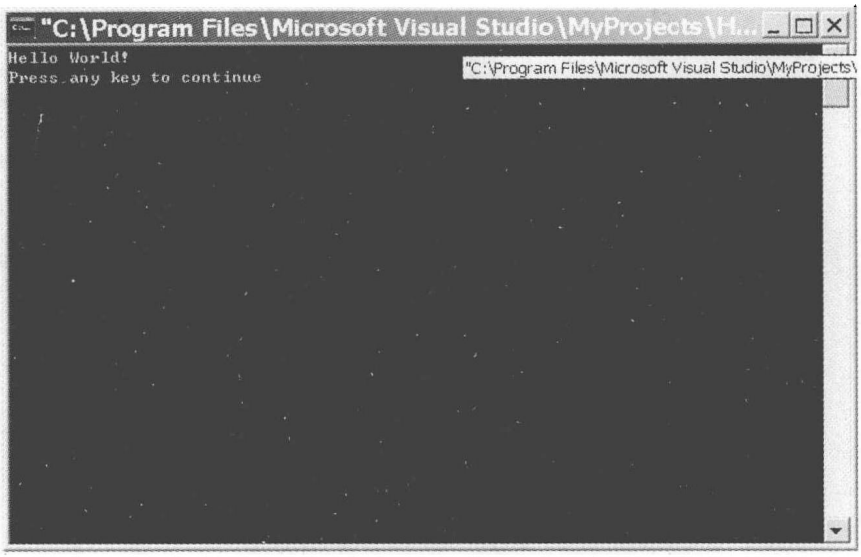

图 1-8　第一个程序的运行结果

可以看出，我们的第一个程序十分简单，就是在屏幕上输出“Hello World!”这个简单的句子，虽然这个程序十分简单，但是这是通过我们编程而让计算机按照我们的要求显示的句子，我们将通过逐步的学习程序设计，能让计算机进行复杂的计算，让计算机显示精美的图片和动画，更进一步我们可以通过设计游戏程序让计算机变成一台精巧多变的游戏机，或者极大地提高我们的工作效率。

下面简单说明一下C语言程序的基本组成部分。写一个C语言程序和我们写一篇中文的文章一样，有一个基本的规范，比如我们写文章有提出问题、分析问题、解决问题的过程，C语言的程序也可以分为好几个部分，并且限制相对于我们写文章更加严格。

我们给程序加上行号以便解释。

【程序 1_1.c】（将下面6行输入计算机并保存为 1_1.c）

```
1 #include " stdafx.h"
2 int main (int argc, char * argv [])
3 {
4 printf (" Hello World! \n");
5 return 0;
6 }
```

第1行是 Microsoft Visual C++的预编译头，如果是要使用 Microsoft Visual C++，一般都要加上这一行。当然经过设置也可以不要预编译头。

第2行叫做主函数（main function），意思是说这是一个函数，完成某个特定的功能。这里函数的概念和数学里面的相似，也是给定一些输入，然后产生一些输出。主函数的意思

是说，虽然程序里面可以有很多个函数，但是必须从主函数开始执行这个程序。

第 3 行和第 6 行合起来构成了一对花括号，表示了主函数开始和结束的地方。

第 4 行调用了一个函数，这个函数的作用是向屏幕输出某些内容，在这里，我们输出了“Hello World!”。这个函数是提供好了的，我们可以直接使用。

第 5 行用来表示这个函数的输出，这个函数的输出是一个简单的 0，而且不管输入是什么，输出都是 0。

下面，我们通过另外一个例子，来进一步了解 C 语言程序的结构。

【程序 1_2.c】

```
1  #include<stdafx.h>
2  int add (int a, int b) /*此函数的功能是返回 a+b 之和 */
3  {
4     return a+b;
5  }
6  main ()
7  {
8     int c;
9     c=add (3, 5);
10     printf ("%d\n", c);
11     }
```

这个程序比刚才的第一个程序复杂，多了一个 add 函数。其中，我们可以看出一些 C 程序的规则。

1）C 语言的程序可以由多个函数组成，但是函数之中不允许有其他的函数。比如从第 2 行到第 5 行，我们定义了一个新的函数，名字叫做 add。

2）函数可以有参数，作为函数的“自变量”。比如第 2 行中的 a 和 b。

3）函数可以有结果，作为函数的“因变量”。比如第 4 行中的返回结果。

4）为了让程序更容易被理解，可以加入注释，注释以“/*”开始，以“*/”结束，中间的部分是注释。第 2 行中的“/*此函数的功能是返回 a+b 之和*/”就是注释。

在以后的学习中，我们将通过对 C 语言的学习，了解到更多关于书写 C 语言程序的规则。

1.7 程序书写规范

为了让整个程序看起来整洁优雅，我们要按照一定的样式来书写程序，这些样式不是必须遵守的，但是如果使用了这些建议的规则，那么整个程序会变得容易阅读，容易理解程序的目的和含义，容易发现和修改其中存在的错误。

（1）对齐

程序一般是有层次结构的，所以我们一般按照一定的层次来书写程序，同时，如果一个