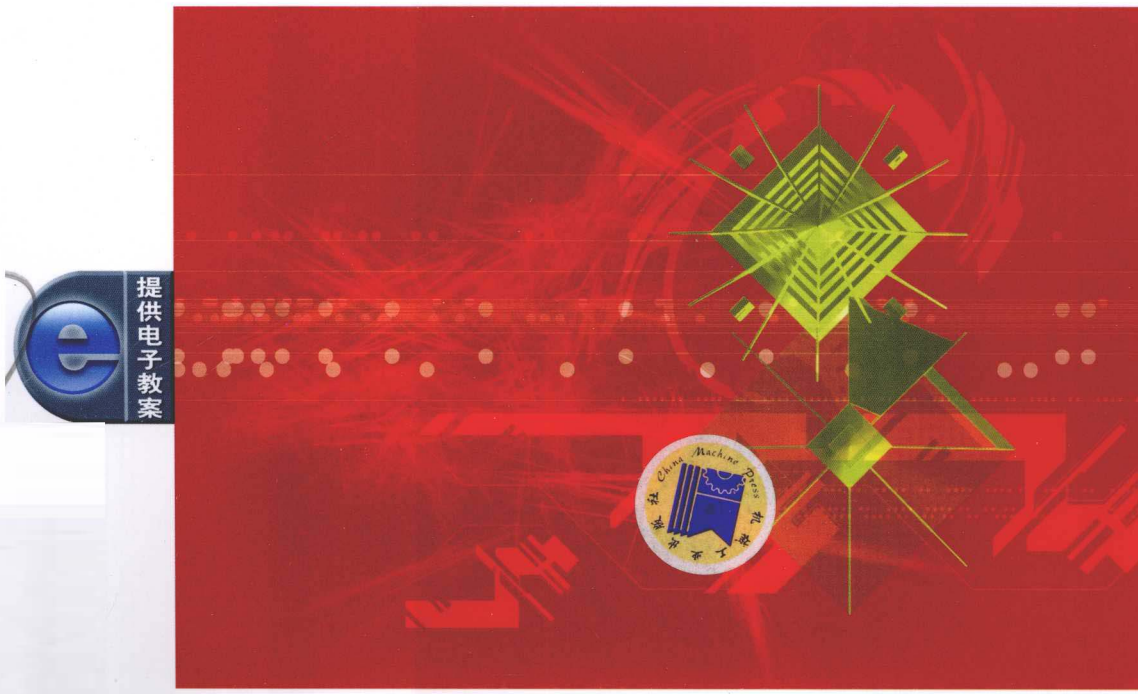
21世纪高等院校应用型规划教材

# C 程序设计实例教程




金林樵　主编
陈伟芳　陈超祥　邱 宁　编著

机械工业出版社
CHINA MACHINE PRESS

21 世纪高等院校应用型规划教材

# C 程序设计实例教程

金林樵　主编
陈伟芳　陈超祥　邱　宁　编著

机 械 工 业 出 版 社

本书从一个核心实例——学生成绩管理系统的开发入手，围绕实例阐述程序设计的基本概念及C语言程序设计的基本思想，有很强的实用性。全书共10章，主要内容包括：程序设计和C语言、C语言程序基础、顺序结构程序设计、程序控制、数组与字符串、模块化程序设计、指针、构造数据类型、位运算和文件等。另外，每章配有大量的例题和习题，便于读者巩固所学知识，掌握程序设计的基本方法与编程技巧。

书中列举了与学生成绩管理系统相关的系统分析及实现代码。为了便于读者更好地理解代码，程序中给出了详细的注释，所有代码均在Visual C++.NET 2005中调试通过，也可将代码在TC、VC++及Free-C等环境中直接使用。

本书可作为高等学校计算机与信息技术及相关专业的基础教材，也可供相关领域的工程技术人员参考或自学。

**图书在版编目（CIP）数据**

C程序设计实例教程/金林樵主编. —北京：机械工业出版社，2010.2
（21世纪高等院校应用型规划教材）
ISBN 978-7-111-29648-5

Ⅰ.C… Ⅱ.金… 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（2010）第016700号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
责任编辑：唐德凯
**责任印制：洪汉军**
**北京市朝阳展望印刷厂印刷**

2010年3月第1版·第1次印刷
184mm×260mm·18印张·438千字
0001-3000册
标准书号：ISBN 978-7-111-29648-5
定价：32.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：（010）88361066
销 售 一 部：（010）68326294
销 售 二 部：（010）88379649
读者服务部：（010）68993821

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com

**封面无防伪标均为盗版**

# 出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21 世纪高等院校计算机教材系列”、“21 世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21 世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列，以及信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。另外，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。在此，衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

# 前　言

随着计算机技术、网络技术的不断发展和日益普及，程序设计已成为现代计算机信息系统和计算机应用系统最基础的技术和核心。C 语言的表达能力强，可移植性好，使用方便、灵活，目前已成为国内外公认的应用面广、基础性好的程序设计入门语言。

本书以学生成绩管理系统的开发作为贯穿始终的一个核心实例。选择这样一个系统的原因是读者比较熟悉有关成绩管理系统项目和功能，同时开发学生成绩管理系统将涉及程序设计课程的全部知识点。通过该项目的实例化驱动教学，可以提高读者的学习兴趣，从而达到学好 C 语言的目的。

C 语言程序设计是实践性很强的课程，在教学过程中应注重实践环节，以提高实际应用能力。书中的实例程序算法清楚，注释较多，可读性强。书中内容通俗易懂，由浅入深，每当提出一个新概念，就给出其清晰的定义和充分的说明，并提供一段完整、恰当的实例程序，且书中所介绍的实例尽可能地围绕学生成绩管理系统展开。有了这些连贯的实例，不仅可使学生更为方便地理解概念，而且可以循序渐进地理解 C 语言中较为复杂的概念。在每章之后均设置了习题和实训，有助于读者对所学知识的巩固，并充分体现本书的实用性和易用性。本书共分为 10 章，包括程序设计和 C 语言、C 语言程序基础、顺序结构程序设计、程序控制、数组与字符串、模块化程序设计、指针、构造数据类型、位运算和文件等内容，还给出了相关的附录。

本书第 1、7、8、9、10 章由金林樵编写，第 2、3 章由陈伟芳编写，第 6 章由陈超祥编写，第 4、5 章由邱宁编写。全书由金林樵统稿与审定。

书中使用的源代码是作者经过几轮教学实践并反复提炼后形成的，所有代码都进行了多次调试，能正常编译与运行。为方便教学，作者将全书的 PPT 及全部程序的源代码放在网站上供用户免费下载，下载地址为 Http://www.cmpedu.com 或 http://60.191.53.249/WebCoursec。

由于作者水平有限，书中难免有不足之处，敬请读者批评指正，提出宝贵意见。

作　者

# 目　录

# 第1章　程序设计和C语言

C语言是一种通用、简洁、灵活、功能强大、应用广泛的程序设计工具。本章从程序设计的基本概念入手，对程序设计及C语言进行简要介绍，使读者一开始就能上手编写简单的C程序，并能上机对程序进行运行和简单的调试。

## 1.1　程序与程序设计

程序——顾名思义，是指为完成某些事务而需执行的步骤。通俗地可以将程序看做是一系列动作（为完成该事务）的执行过程的描述。日常生活中存在着许多“程序”的实例。例如，要到ATM上取一笔钱，那么就要按照如下的“程序”进行。

1）找到要取款的ATM。

2）将取款用的银行卡插入ATM。

3）输入银行卡的密码。

4）若密码正确，则进入下一步；否则返回3）重新输入密码。

5）按“取款”键，并输入取款金额。

6）按“确定”键。

7）如该卡上有足够的余额，则：

① ATM吐出指定数目的钱；② 从ATM中取钱，并保管好。

8）按“退卡”键，将银行卡从ATM上退出。

9）保管好银行卡和钱后离开。

经过这9个步骤后，才完成了从ATM取款的整个过程。这9个步骤必须按顺序依次完成，在没有完成上一步的前提下，就不能进入到下一步，这就是生活中的“程序”。

在上面的“程序”中，1）~3）步是顺序完成的，第4）步要对输入的密码进行验证，这就存在着分支的情况：若输入的密码不正确，就必须返回3）重新输入密码，在超过1次输入不正确的情况下，3）~4）就多次被执行，直到密码输入正确或卡被吞掉，这是一个重复性的动作，构成了循环。同样在第7）步也存在着分支情况：要对输入的取款金额进行比较，只有取款金额不超出该卡余额的情况下，才能实现ATM吐钱的动作；否则ATM将提示“错误”信息。

从上面的例子中可以看出“程序”的一些特征：一个“程序”总有开始和结束，期间要按部就班地完成一系列相关的动作，在达到结束位置时任务就完成了。

### 1.1.1　程序和程序设计的概念

程序是对解决某个问题的方法（算法）步骤的一种描述。对于计算机来说，其程序与生活中的“程序”有些相似，计算机程序是用某种计算机能理解并执行的计算机语言作为

描述语言，对要解决问题中的数据以及处理这些数据的方法和步骤所进行的准确和完整的描述。计算机能自动地按程序中所描述的方法步骤依次执行，完成指定的功能。可以说，程序就是供计算机执行后能完成特定功能的指令序列。人与计算机交流的基本方式就是提供要求它执行的程序。

一个计算机程序主要描述两部分内容：

1）描述问题所涉及的每个对象（数据）及对象之间的关系。

2）描述对这些对象进行运算与控制的处理规则。

其中关于对象及对象之间的关系是数据结构的内容，而处理规则就是问题的求解算法。针对问题所涉及的对象和要完成的处理，设计合理的数据结构可有效地简化算法，数据结构和算法是程序最主要的两个方面。

编制计算机程序的工作被称为程序设计或者编程，这种工作的产品就是程序，而编制计算机程序的人称为程序员。

程序设计的工作是寻求解决实际问题的方法步骤（算法），并将其实现步骤使用某种计算机语言写成计算机可执行的程序。

程序设计的主要步骤包括：

1）确定数据结构。依据所要处理的任务要求，规划输入的数据和输出的结果，确定存放数据的数据结构。由于在 C 语言中数据结构集中体现在数据类型上，因此在进行 C 语言程序设计时，应统筹规划程序中所使用的变量、数组、指针及它们的类型等。

2）设计解决问题的算法。算法是指为解决某一特定问题而采取的有限和确定的步骤。对于同一个问题，每个人确定的算法都不可能完全相同。要成为优秀的程序员，应该学习已有的比较优秀和经典的算法，从中吸取精髓，不断提高。

3）编写程序。在充分论证数据结构和算法以后才能考虑编写程序，编写程序需要结合程序设计方法（面向过程或者面向对象）和程序设计语言（C 语言、C ++ 、Pascal、VB 等）。本书是使用面向过程的 C 语言进行编程的。

4）程序调试与运行。程序员编写的程序称为源程序或源代码，源程序不能直接被计算机执行。源程序要经过编译程序编译，生成目标程序，然后链接其他相关的代码，最后生成可被计算机执行的可执行文件（. EXE 或 . COM 文件）。一个源程序有时要经过多次的修改才能通过编译、连接，因此这一步对初学者来说是比较困难的，需要不断练习才能取得成功。程序编译和连接时，如不能通过，则系统会给出错误提示信息，程序员要根据提示信息修改程序。

在程序设计过程中，上述步骤可能需要不断重复，直到得到正确的结果为止。

### 1.1.2 程序设计语言及其发展

上面从 ATM 上取钱的“程序”是人类的自然语言，目前的计算机还不能直接识别。

为了能与计算机交流，指挥它按人们设定的步骤与方式工作，就需要有与之交流的语言，需要一种语意清晰、人类用起来比较方便、计算机也能处理的描述语言。也就是说，需要有描述程序的合适语言。可供人书写计算机程序的语言就称为程序设计语言，这是一种人造语言，也常被称为编程语言或程序语言。

随着计算机技术的不断发展，为适应新技术的要求，程序设计语言也随之向更高、更好

的方向发展，经历了从机器语言、汇编语言到高级语言的历程。

**1. 机器语言**

机器语言是计算机与生俱来就有的，它随着计算机的诞生而产生，是一种最贴近计算机硬件的语言。机器语言是 CPU 可以识别的由 0 和 1 构成的一组二进制序列指令码，是计算机硬件唯一能接受和执行的计算机语言。每一串二进制代码称为一条指令，一条指令规定计算机执行一个指定的相关动作。一台计算机所能执行的指令集合就称为该计算机的指令系统。

由于机器语言直接面向计算机硬件编程，因此对于人的使用而言，二进制形式的机器语言很不方便，不但书写程序非常困难，工作效率极低，而且程序的正确性也很难保证，出现错误时也很难辨别和改正。例如，要在 8086 指令系统的计算机中，计算算术表达式 a×b + c（假设 a、b、c 的值存放在 1000、1008 和 1020 存储单元中）的值，并将计算结果保存在 d（假设 d 的值存放在 1060 存储单元中）的计算过程中，使用的机器语言指令序列为：

```
1010000000000000000010000      --将单元 1000 的数据(a)装入寄存器 AL
10001010000111100000100000010000   --将单元 1008 的数据(b)装入寄存器 BL
1111011011100011               --完成 a 乘 b 的操作,并将结果装入寄存器 AX
10001010000111100010000000010000   --将单元 1020 的数据(c)装入寄存器 BL
0000000011011000               --完成 AL 加 BL 的操作,并将结果装入寄存器 AL
10100010011000000010000       --将寄存器 AL 里的数据(表达式的值)存入单元 1060
```

由于不同品牌和型号的计算机，其指令系统往往是不同的，所以在一种计算机上能执行的程序，要想在另一种计算机上执行，可能还需要另编程序，由此造成了大量的重复工作。但由于使用的是针对特定型号计算机的语言，故而机器语言的运算效率是所有语言中最高的。

**2. 汇编语言**

从上述示例的机器语言程序中可以看到：用机器语言编写程序，程序员首先要熟记所用计算机的全部指令代码和代码的涵义，手工处理每条指令和各个数据的存储分配和输入输出，还需记住编程过程中每步所使用的存储单元及其所处的状态。这是一种相当烦琐的工作，既难以记忆也难以操作，编写出来的程序全是由 0 和 1 的数字组成，直观性差，难以阅读。不仅难学、难记、难检查，又缺乏通用性，给计算机的推广使用带来很大的障碍。

为了克服机器语言难读、难编、难记和易出错的缺点，人们就用与代码指令实际含义相近的英文缩写词、地址码等符号来取代二进制指令代码，即将机器语言的每一条指令符号化：指令码代之以助记的符号名（英文缩写词，例如用“MOV”表示数据传递，“ADD”表示加法等），地址码代之以符号地址，使得其含义显现在符号上而不再隐藏在编码中，可让人望“词”生义。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，由此汇编语言就诞生了。例如，使用汇编语言完成上述计算算术表达式值的指令序列为：

```
MOV    AL,[1000]      --将单元 1000 的数据(a)装入寄存器 AL
MOV    BL,[1008]      --将单元 1008 的数据(b)装入寄存器 BL
MUL    BL             --完成 a 乘 b 的操作,并将结果装入寄存器 AX
MOV    BL,[1020]      --将单元 1020 的数据(c)装入寄存器 BL
ADD    AL,BL          --完成 AL 加 BL 的操作,并将结果装入寄存器 AL
```

```
MOV    [1060],AL          --将寄存器 AL 里的数据(表达式的值)存入单元 1060
```

从上面两段功能完全相同的程序中可以看出，汇编语言由于采用了助记符号来编写程序，用符号代替了机器指令代码，而且助记符与指令代码一一对应，因此汇编语言比起机器语言在很多方面都有很大的优越性，如编写容易、修改方便、阅读简单、程序较清晰等，在一定程度上简化了记忆，大大缩短了编程过程。这无疑是机器语言朝高级算法语言迈出的一大步。虽然汇编语言能面向机器并较好地发挥机器的特性，得到质量较高的程序，但它像机器语言一样，也是面向机器硬件的操作，因而仍然属于低级语言的“范畴”，不同的计算机需要有不同的指令集，因而在编制复杂程序时，还是比较烦琐、费时，通用性也较差，离高级算法语言还有较大的距离，复杂程序作为整体仍然难以理解。

汇编语言中由于使用了助记符，计算机不能像用机器语言编写的程序一样直接识别和执行，必须通过一个专用的“汇编程序”对其进行加工，逐条翻译成二进制机器指令后，才能变成能被计算机直接识别和处理的二进制代码程序。

**3. 高级语言**

随着计算机技术的发展，程序的复杂程度越来越高，汇编语言已无法满足程序的编写要求，促使人们去寻求一些与人类自然语言相接近且能为计算机所接受的语意确定、规则明确、自然直观和通用易学的计算机语言。这种与自然语言相近并为计算机所接受和执行的计算机语言称为高级语言，它不再依赖于某种特定的计算机，因此编程的重点自然地转移到解题的过程和算法的描述上。1954 年，IBM 公司工程师约翰·巴科斯（J. Backus）带领一个小组，在 IBM 704 计算机上设计出了第一个计算机高级语言——FORTRAN 语言，宣告了程序设计新时代的开始。此后，各种高级语言如雨后春笋般出现，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、Ada、Pascal、C、PROLOG、LOGO、C++ 、VC、VB、Java 等。

例如，在高级语言（例如 C 语言）中，描述前面同样的程序片断只需一行代码即可：

```
d=a*b+c;
```

这表示先要求计算机求出“等于号”右边表达式的值，然后将计算结果存入由 d 代表的存储单元中。这种表示方式与人们所熟悉的数学形式非常相近，因此更容易阅读和理解。

用高级语言编写的程序称为高级语言源程序。高级语言是面向用户的语言，因此计算机不能直接执行高级语言描述的程序，必须翻译成机器语言目标程序才能被计算机执行。高级语言的翻译有两种方式：编译方式和解释方式。

1）编译方式。先通过编译程序（由系统软件提供）把由高级语言编写的源程序翻译为与所用计算机等价的机器语言程序（即目标程序）。此后，只要命令计算机执行这个机器语言目标程序，计算机就能完成程序所规定的任务了。C、Pascal 等高级语言就采用这种翻译方式。

2）解释方式。在运行高级语言源程序时，由解释程序对源程序边翻译边执行。在源程序进入计算机时，由解释程序用边扫描边解释的方式逐句读取逐句翻译，计算机一句句地执行，但翻译过程中不产生目标程序。BASIC 语言则主要采用这种翻译方式。

20 世记 80 年代前的高级语言几乎都是面向过程的。程序的执行像流水线似的一个接着一个，在一个模块被执行完成前，人们不能做别的事，也无法动态地改变程序的执行方向。

这和人们日常处理事物的方式是不一致的，对人而言是希望发生一件事就处理一件事，也就是说，不能面向过程，而应是面向具体的应用，由此计算机界又诞生了一种当前十分流行的高级语言——面向对象（Object）的程序设计语言，如 VB、C ++ 、Object Pascal、Java 等就是其典型的代表。

## 1.2 C 语言简介

C 语言是贝尔实验室 Dennis Ritchie 在 1973 年设计的一种程序设计语言，其目的是用于编写操作系统和系统程序，初期用在 PDP - 11 计算机上写 UNIX 操作系统。20 世纪 70 年代后期作为 UNIX 的标准开发语言，C 语言随着 UNIX 系统的流行而得到越来越广泛的应用，20 世纪 80 年代后期它被搬到包括大型机、工作站等的许多系统上，逐渐成为开发系统程序和复杂软件的一种通用语言。随着计算机的蓬勃发展、处理能力的提高和应用的日益广泛，越来越多的人参与计算机应用系统的开发工作，需要适合开发系统软件和应用软件的语言。C 语言能较好地满足人们的需要，因此在计算机软件开发中得到日益广泛的应用，逐渐成为最常用的系统开发语言之一，被人们用于开发计算机上的各种程序，甚至非常复杂的软件系统。

### 1.2.1 C 语言的发展

C 语言的前身是 ALGOL 语言。用 ALGOL 60 来描述算法很方便，但它离计算机硬件系统太远，不宜用来编写系统程序。

1963 年剑桥大学在 ALGOL 语言基础上增添了处理硬件的能力，并命名为 CPL（Combined Programming Language）语言。CPL 由于规模大，学习和掌握困难，因此没有流行开来。

1967 年剑桥大学的 Matin Richards 对 CPL 语言进行了简化，推出 BCPL（Basic Combind Programming Language）语言。

1970 年美国贝尔实验室的 Ken Thompson 对 BCPL 进行了进一步的简化，突出了硬件处理能力，并取了“BCPL”的第一个字母“B”作为新语言的名称。同时用 B 语言编写了 UNIX 操作系统。

1972 年贝尔实验室的 Brian W. Kernighian 和 Dennis M. Ritchie 对 B 语言进行了完善和扩充，在保留 B 语言强大的硬件处理能力的基础上，扩充了数据类型，恢复了通用性，并取了“BCPL”的第二个字母作为新语言的名称，这就是著名的 C 语言。此后，两人合作，重写了 UNIX 操作系统，C 语言成为 UNIX 的标准开发语言，此后随着 UNIX 系统流行而得到越来越广泛的应用。为表彰他们在程序设计语言方面的卓越贡献，两人分享了 1983 年美国计算机协会（ACM）的图灵奖，成为有史以来少数几个因工程项目得奖的工程师。

1978 年，为了让 C 语言成为任何计算机上都能运行的通用计算机语言，Brian W. Kernighian 和 Dennis M. Ritchie 又合著了《The C Programming Language》一书，通常简称为《K&R》，对 C 语言的语法进行了规范化的描述，从而使 C 语言成为世界上应用最广泛的高级程序设计语言，《K&R》也成为 C 语言方面最权威的教材之一。

随着微型计算机的日益普及，出现了许多不同的 C 语言版本，使得这些 C 语言之间出现了一些不一致的地方。为了统一标准，美国国家标准化协会（ANSI）于 1987 年制定了 C 语言的标准，称为“ANSI C”。通常将“K&R”的标准称为旧标准，将“ANSI C”称为新标准。此后人们对 ANSI C 进行了一些修订和扩充，1999 年通过了 ISO/IEC 9899：1999 标准，通常被称为 C99。目前，几乎所有的开发工具都支持 ANSI C 标准，是 C 语言使用最广泛的一个标准版本，但对 C99 来说各大计算机公司表现出了不同的兴趣，支持率不如 ANSI C。

### 1.2.2 C 程序的特点

C 语言是一种通用编程语言，其主要特色是兼具了高级语言和汇编语言的特点，简洁、丰富、移植性好。使用 C 语言编写程序，既能感觉到使用高级语言的自然，又能体会到利用计算机硬件指令的直接，而程序员却无须考虑机器语言、汇编语言的烦琐。C 语言使用函数将一个大问题化解成若干个小任务，因此函数使 C 程序模块化。C 语言提供了结构式编程所需要的各种现代化的控制结构，方便灵活的控制机制，使其被越来越多的程序员所推崇，成为目前最为流行的高级语言之一。概括一下，其主要特点有：

1）数据类型和运算符十分丰富，程序设计和算法描述更为简单和方便。

2）语法结构十分简单，语句数目相对较少，简单易学。程序相对易编、易读、易查、易改，编制程序的效率高。

3）C 语言兼有高级语言和低级语言的优点。提供了一些比较接近硬件的低级操作函数，因此适合编写需要直接与硬件打交道的程序。这些使 C 语言常被用作汇编语言的“替代物”，大大提高了开发底层程序的效率，所以也称之为中级语言。

4）C 语言是一种结构化程序设计语言，提供了完整的程序控制语句：顺序、选择和循环，很适合结构化的程序设计方法。

5）C 语言是一种模块化程序设计语言，方便易用的函数定义和使用机制可以把复杂问题分解成一个个具有一定独立性的函数，每个函数完成某个特定的功能，如此可以分解程序的复杂性，使之更容易控制和把握。

6）C 语言为开发者提供了大量的库函数（如输入/输出函数等）。这就使语言本身比较简单，可移植性好，大大简化了程序设计工作，促进了 C 语言的广泛传播。

## 1.3 C 程序的基本结构

没有规矩，不成方圆，任何一种语言都具有特定的语法规则及规定的表达方法（如中文、英文），C 程序也不例外。一个程序只有严格按照语言规定的语法和表达方式编写，才能保证编写的程序按程序中设定的功能正确地执行。

### 1.3.1 简单的 C 程序

本书将从下章开始详细讨论各种 C 语言结构及其设计，说明 C 程序设计中出现的各种情况和问题。为让读者先对 C 程序有一个感性的认识，最好的方法是实际看一下用 C 语言编写的程序到底是什么样子的（不要求全部看明白），先来看 3 个简单的 C 程序。

**【例 1-1】** 从终端上输出一串指定信息。

```
/*包含文件。本程序中需要用到C语言系统提供的标准函数时,需要在文件的开始部分
先包含该标准函数的函数声明所在的包含文件,这里是stdio.h*/
#include "stdio.h"
main(  )          /*C程序的主函数*/
{
    printf("\nHello:\nThis is Your first C program\n");     /*输出一串信息*/
}
```

该程序的第1、2行是以“/*”开始，以“*/”结束的注释，第4、6行的尾部也有一段注释。在程序中使用注释的目的是提高程序的可读性，让读者了解语句、语句段或整个函数的功能、设计和实现方法等，一般用于程序的后期维护。C系统忽略程序中的所有注释部分（即对注释部分不作任何处理），因此注释是不会被计算机执行的。

第3行的include命令是编译预处理命令行。其意义是把引号（“”）或尖括号（〈〉）内指定的文件包含到本程序中，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名一般为.h，在C语言中称为头文件。C语言的头文件中包括了各个标准库函数的函数原型，因此凡是在程序中需调用一个标准库函数时，都必须包含该函数原型所在的头文件。在本例中，使用了printf输出函数。scanf和printf是标准输入输出函数，其头文件为stdio.h，故在main函数前用include命令包含了stdio.h文件。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。

第4行是函数的开始部分。该函数的函数名是main，main是C系统使用的专用函数名，称为主函数，它告诉C系统该程序从这里开始执行。紧跟main后面的是一对圆括号，表示该函数是否有参数。本程序中为空，表示函数不使用参数，关于参数的概念将在“模块化程序设计”一章中作详细介绍。

第5和7行是一对匹配的花括号“{”和“}”，是main函数的一部分（C程序中称为函数体），其中的语句用来完成该函数中要实现的任务。在本例中只有一个printf输出函数，表示程序要调用printf函数输出指定的信息。

printf函数是C系统的标准库函数。紧跟printf后面的是一对圆括号，圆括号中用一对双引号括起来的是要传递给printf函数的参数，它是一个字符串，表明要向终端输出的信息。字符串中的“\n”是C程序的换行符（newline），表示其后的字符从下一行的起始位置输出，相当于字处理软件中的回车键。

本例程序的运行结果是向终端（显示器）输出以下两行信息：

```
Hello:
This is Your first C program
```

**【例1-2】** 输入一门课程的平时、期中和期末成绩，按平时20%、期中30%和期末50%的比例计算并输出该课程的实际成绩。

```
#include"stdio.h"
main(  )          /*C程序的主函数*/
{
  int grade1,grade2,grade3;          /*定义存放平时、期中和期末成绩的变量*/
```