普通高等教育“十一五”国家级规划教材 计算机系列教材

# 程序设计基础(C语言)

田爱奎 张先伟 张立红 王云 编著

清华大学出版社

普通高等教育“十一五”国家级规划教材　计算机系列教材



田爱奎　张先伟　张立红　王云　编著

# 程序设计基础(C语言)



清华大学出版社
北京

## 内 容 简 介

本书重点介绍在 C 语言环境下编写程序的思路与方法。全书以程序设计的基本思想与方法作为主要结构，主要介绍了程序的基本结构、批量数据的组织方式与处理技巧，引入了分治与递归、动态规划、贪心等常用的算法设计应用案例，注重强调了程序设计的设计方法与实践能力的培养。

本书可作为大专院校教材，亦可供从事计算机相关领域的科研人员参考自学。



**图书在版编目（CIP）数据**

程序设计基础（C 语言）/ 田爱奎等编著. —北京：清华大学出版社，2011.6
（计算机系列教材）
ISBN 978-7-302-25986-2

Ⅰ. ①程… Ⅱ. ①田… Ⅲ. ①C 语言-程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 126207 号

**责任编辑**：魏江江
**责任校对**：焦丽丽
**责任印制**：杨　艳
**出版发行**：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
**社　　总　　机**：010-62770175　　**邮　　购**：010-62786544
**投稿与读者服务**：010-62795954，jsjjc@tup.tsinghua.edu.cn
**质　量　反　馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn
**印 刷 者**：北京四季青印刷厂
**装 订 者**：三河市新茂装订有限公司
**经　　销**：全国新华书店
**开　　本**：185×260　**印　张**：19.5　**字　数**：475 千字
**版　　次**：2011 年 6 月第 1 版　　**印　次**：2011 年 6 月第 1 次印刷
**印　　数**：1～3000
**定　　价**：29.50 元

产品编号：037981-01

程序设计基础是计算机专业的一门最基础的课程，也是学生接触的第一门专业基础课。通过该课程的学习，学生不仅要掌握程序设计语言本身，更重要的是培养程序设计的基本能力，特别是用计算机解决问题的思想、习惯与方法。

本书内容共分为三部分，分别介绍程序的基本结构、批量数据的组织方式与处理技巧、常用算法的思路与应用典型案例。第一部分主要介绍程序结构与函数，力求结合案例以简练的内容介绍最常用的知识与离散数据的处理方法，是程序设计能力的基础；第二部分主要以经典案例与实践训练相结合的方法介绍批量数据的组织方法与处理技巧，是对程序设计能力的拔高；第三部分主要以基本算法理论与经典案例结合的方式介绍常用算法，是程序设计能力的升华。

与传统的 C 语言教材相比较，本书虽然仍以 C 语言作为工具，但是不再以琐碎的 C 语言语法知识作为内容的核心，而是以程序设计的基本思想与方法作为主要结构，在理论体系中突出重点，淡化不常用、非必须且难理解的语法内容，引入常用的算法设计（如分治与递归、动态规划、贪心等）方法应用案例与练习，使得教材内容体现比较完整且重点突出的程序设计知识体系。

全书由田爱奎、张先伟策划、统稿。其中第 1、2 章由田爱奎编写，第 7、10、11、12 章由张先伟编写，第 5、6、8、9 章由张立红编写，第 3、4 章由王云编写，刘晓红、马新娟等在教材编写过程中也提出了许多有益的意见与建议，在此一并表示感谢。由于时间仓促，作者水平有限，书中难免有纰漏之处，敬请读者批评指正。

编者

2011.5

# 第1章　程序设计引论

## 1.1　计算机程序与计算机语言

### 1.1.1　计算机程序

随着信息技术的飞速发展，计算机的应用已经深入到人们日常生产生活的方方面面。从表面来看，似乎计算机是无所不能的，可以自动完成许多工作，但实际上计算机所做的每一项工作都是由程序员预先设计的。程序员要事先设计完成针对各项任务的程序代码，把这些代码输入到计算机中，在处理该任务时计算机会按照程序代码中处理指令的要求完成处理工作。

所谓程序，是一组计算机能识别和执行的指令的集合，其中每一条指令使计算机执行特定的操作，这一组指令就是计算机处理某项具体事务过程中的一个操作序列（或称之为处理流程）。计算机执行该程序时，实际上是在“自动地”执行程序中的各条指令所对应的操作，有条不紊地完成处理工作。

编写一个计算机程序就像写一篇文章，写文章首先要学会基本的字、词、句等语法知识，因为文章本身就是一些基本字、词、句的组合体，不掌握这些语法知识就无法写出别人可以读懂的文章。但仅仅只是字、词、句的组合体还不是一篇精彩的文章，好的文章应该有精巧的组织结构来把这些字、词、句合理地组织在一起，通过这样的文章架构来表达出深刻的思想内涵。

要想写出一个计算机程序，也至少需要两个基本要素：编程语言开发工具与编程方法。编程语言开发工具是程序实现的载体，正如写文章需要的字、词、句语法知识一样，编程语言开发工具给出了计算机能够识别的各种基本操作的“字、词、句”表示形式，提供了人机交互的基本平台；编程方法是程序设计的灵魂，给出了解决实际问题的具体策略，也就是指定了一个计算机程序所需要的操作以及这些操作之间的组合方式，正如文章中字、词、句的组织架构一样。

总之，计算机的一切操作都是由程序控制的，计算机帮助人们解决实际问题的过程就是一个执行具体程序的过程。因此，程序是计算机系统中最基本的概念。只有了解程序设计，才能真正了解计算机是怎样工作的；也只有掌握了程序设计，也才能更好地使用计算机。

### 1.1.2　计算机语言

语言是一种交流的工具。“人有人言，兽有兽语”，不管是人与人之间还是动物之间，要相互交流思想、传递感情，最直接的工具就是语言。人们要利用计算机解决实际问题，就要把自己解决问题的思想以特定的方式传递给计算机，让计算机能够按照所接受的信息

去执行相应的操作，再把执行的结果反馈给用户，这是人与计算机之间交流的过程。人与计算机之间的交流也需要一种“语言”，借助这种语言，使得计算机和人都能读懂对方传递给自己的信息，从而实现人与计算机之间有效的信息交流。这种能够实现人与计算机交流的“语言”载体被称为计算机语言。

随着计算机技术的不断发展，计算机语言也经历了一个从低级到高级、从简单到复杂的发展过程。迄今为止，计算机语言发展主要经历了以下三个阶段。

**1．机器语言**

用二进制位串形式表示的指令是计算机可以直接识别和执行的指令，人们把这样的指令称为机器指令（Machine Instruction）。机器指令是用 0 和 1 组成的一串代码，它们有一定的位数，并分成若干段，各段的编码表示不同的含义，例如某台计算机字长为 16 位，即有 16 个二进制数组成一条指令或其他信息。16 个 0 和 1 可组成各种排列组合，通过线路变成电信号，让计算机执行各种不同的操作。例如，某种计算机的指令为 1011011000000000，它表示让计算机进行一次加法操作；而指令 1011010100000000 则表示进行一次减法操作。它们的前八位表示操作码，而后八位表示地址码。从上面两条指令可以看出，它们只是在操作码中从左边第 0 位算起的第 7 和第 8 位不同。这种机型可包含 256（2 的 8 次方）个不同的指令，由于在一个任务中需要计算机执行的操作有很多，要使计算机识别和执行不同的操作，就要给出许多条由 0、1 位串组成的不同指令，计算机按照指令的要求执行各种操作。

由机器指令及其语法规则所构成的集合就是该计算机的机器语言（Machine Language）。机器语言又称为二进制代码语言，计算机可以直接识别，不需要进行任何翻译。由于每台计算机的指令、指令格式和代码所代表的含义都是硬性规定的，故称之为面向机器的语言，简称机器语言。

机器语言是第一代计算机语言，对不同型号的计算机来说一般是不同的。既然是面向机器的，这种用机器语言编写的程序是计算机可以直接识别的符号，但是由于其 01 代码串的形式与人们习惯用的自然语言差别太大，不容易学，也不容易读。因此初期只有极少数的计算机专业人员会编写计算机程序。

**2．汇编语言**

为了使编程语言编写的程序更具备可读性，人们逐渐考虑用一些符号来表示指令中的操作码和地址码，就诞生了符号语言 （Symbolic Language）。它是用一些英文字母和数字符号表示一个指令，例如用 ADD 代表“加法操作”，SUB 代表“减法操作”，LD 代表“数据传送”等。如机器指令 1011011000000000 可以改用符号指令代替：ADD A, B（执行 A+B→A，将寄存器 A 中的数与寄存器 B 中的数相加，放到寄存器 A 中）。

由于计算机只能识别二进制位串形式的指令（比如 1011011000000000），并不能直接识别和执行符号语言的指令如“ADD A,B”，这时计算机就需要用一种称为汇编程序的软件，首先用该软件把用户用符号语言的表示的指令“ADD A,B”转换为机器指令 1011011000000000，然后再完成该指令的执行。这种由符号语言的指令转换为对应机器语言指令的过程称为“代真”或“汇编”，因此，符号语言又称为汇编语言（Assembler Language）。

汇编语言比机器语言易于读写、调试和修改，同时具有机器语言全部优点。但在编写复杂程序时，代码量仍然较大，而且汇编语言依赖于具体的处理器体系结构，不能通用，

因此不能直接在不同处理器体系结构之间移植。人们把机器语言与汇编语言称为计算机低级语言（Low Level Language）。

**3．高级语言**

由于汇编语言依赖于硬件体系，且助记符量大难记，于是人们又发明了更加易用的所谓高级语言。这种语言的语法和结构类似普通英文，语句表述形式符合人的自然语言表达的句式与习惯，且由于该语言远离对硬件的直接操作，不依赖于具体机器，用它写出的程序具备可移植性，所以该语言的编程知识更方便用户的学习与使用。

高级语言与人的“距离”更近，而与具体机器“距离”较远，因此计算机就不能直接识别由高级语言所编写的程序。程序运行之前要首先进行“翻译”，目的是把机器不能识别的高级语言语句转换为可以直接识别的二进制位串指令形式。具体过程就是用一种称为编译程序的软件把用高级语言编写的程序（称为源程序，Source Program）转换为机器指令的程序（称为目标程序，Object Program），然后让计算机执行机器指令程序，最后得到结果。

从 20 世纪 50 年代第一个高级语言 FORTRAN 诞生之后，先后出现了 2000 多种不同的高级语言。每种高级语言都有其特定的用途，其中影响较大的有 FORTRAN（20 世纪 50 年代推出的第一个计算机高级语言）、ALGOL（适合数值计算）、BASIC（适合初学者的小型会话语言）、COBOL（适合商业管理）、Pascal（适合教学的结构程序设计语言）、PL/1（大型通用语言）、LISP 和 PROLOG（人工智能语言）、C（系统描述语言）、C++（支持面向对象程序设计的大型语言）、Visual Basic（支持面向对象程序设计的语言）和 Java（适于网络的语言）等。

在上述高级语言中，C 语言是 Combined Language（组合语言）的简称，是一种功能强大且有代表性的语言。它既具有高级语言的特点，又具有汇编语言的特点。它可以作为操作系统设计语言，编写系统应用程序；也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。因此，它的应用范围广泛，不仅仅是在软件开发上，而且各类科研（比如单片机以及嵌入式系统开发等）都需要用到 C 语言。

因此，本书选择以 C 语言作为工具载体，在 C 语言基本语法知识的基础上来介绍程序设计的基本方法与设计技巧。

## 1.2 C 语言的发展

20 世纪 70 年代初，Dennis Ritchie 在贝尔实验室设计了 C 语言。C 语言的前身可以追溯到 ALGOL（1960 年），历经剑桥的 CPL（1963 年）、Martin Richards 的 BCPL（1967 年）以及 Ken Thompson 在贝尔实验室所开发的 B 语言（1970 年）发展而来。尽管 C 语言是一种通用用途的编程语言，但它在传统上是用于系统编程的，比如著名的 UNIX 操作系统就是用 C 语言编写的。

C 语言流行的原因是多方面的。它小巧、高效，是一种功能强大的编程语言，并且具有丰富的运行时函数库。它提供了对计算机的精确控制，却没有采用太多的隐藏机制。由于 C 语言的标准化早在十多年前就已完成，自问世以来，先后经历了 C89、C95 和 C99 三次修订，功能不断得以修正与补充，因此 C 语言问世以后得到迅速推广，成为世界上应用最广泛的程序设计高级语言，也是计算机开发人员必须掌握的一个基本编程工具。

C 语言有以下一些主要特点：

（1）语言简洁、紧凑，使用方便、灵活。C 语言（C99）一共有 37 个关键字、9 种控制语句，程序书写形式自由，主要用小写字母表示，压缩了一切不必要的成分。C 语言程序比其他许多高级语言简练，源程序短，因此输入程序时工作量少。

实际上，C 是一个很小的内核语言，只包括极少的与硬件有关的成分，C 语言不直接提供输入和输出语句、有关文件操作的语句和动态内存管理的语句等（这些操作是由编译系统所提供的库函数来实现的），C 的编译系统相当简洁。

（2）运算符丰富。C 语言的运算符包含的范围很广泛，共有 34 种运算符，C 语言把括号、赋值和强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型多种多样。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。

（3）数据类型丰富。C 语言提供的数据类型包括：整型、浮点型、字符型、数组类型、指针类型、结构体类型和共用体类型等（C99 又扩充了复数浮点类型、超长整型（long long）和布尔类型（bool）等）。尤其是指针类型数据，使用十分灵活，能用来实现各种复杂的数据结构（如链表、树、栈等）的运算。

（4）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）。用函数作为程序的模块单位，便于实现程序的模块化。C 语言是完全模块化和结构化的语言。

（5）语法限制不太严格，程序设计自由度大。例如，对数组下标越界不进行检查，由程序编写者自己保证程序的正确性。程序员应当仔细检查程序，保证其无误，而不要过分依赖 C 语言编译程序查错。对于不熟练的人员，编一个正确的 C 语言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，要求更高一些。

（6）C 语言允许直接访问物理地址，能进行位（bit）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，又具有低级语言的许多功能，可用来编写系统软件。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。

（7）用 C 语言编写的程序可移植性好。由于 C 语言的编译系统相当简洁，因此很容易移植到新的系统。而且 C 语言编译系统在新的系统上运行时，可以直接编译“标准链接库”中的大部分功能，不需要修改源代码，因为标准链接库是用可移植的 C 语言写的。因此，几乎在所有的计算机系统中都可以使用 C 语言。

（8）生成目标代码质量高，程序执行效率高。

许多大的软件都用 C 语言编写，因为 C 语言的可移植性好、硬件控制能力强、表达和运算能力强。许多以前只能用汇编语言处理的问题，后来可以改用 C 语言来处理了。目前流行的嵌入式系统的程序设计也大都是通过 C 语言编程实现的。

## 1.3 简单的 C 程序构成

为了使用 C 语言编写程序，首先应该了解 C 语言程序的结构，下面通过几个简单例子来介绍 C 程序的基本构成。

### 1.3.1 最简单的 C 语言程序举例

先看一个最简单的 C 语言程序：

【例 1.1】 在屏幕上输出以下一行信息。

```
Hello World!
```

**解题思路：** 在主函数中用 printf 函数原样输出以上文字。

**程序：**

```
#include <stdio.h>                          // 编译预处理指令
int main()                                  // 定义主函数
{                                           // 函数开始的标志
    printf("Hello World!\n");               // 输出所指定的一行信息
    return 0;                               // 使函数返回值为 0

}                                           // 函数结束的标志
```

**运行结果：**

```
Hello World!
Press any key to continue
```

以上运行结果是在 Visual C++ 6.0 环境下运行程序时屏幕的显示结果。其中第 1 行是程序运行后输出的结果，第 2 行是 Visual C++ 6.0 在程序本次运行结束后发给用户的提示信息。各个程序运行结束的提示信息均为这样的内容，为节省篇幅，以后的程序运行结果不再显示该行。

C 的程序结构由编译预处理、程序主体和注释构成。

每个以“#”开头的行，称为编译预处理行，常常用来完成文件包含、宏定义等功能，程序第 1 行“#include <stdio.h>”的作用就是一个文件包含的预处理行。在使用函数库中的输入输出函数时，编译系统要求程序提供此函数所在文件的有关信息，stdio.h 是系统提供的库文件的名称，输入输出函数的相关信息已事先放在该文件中。通过文件包含操作把 stdio.h 头文件调入后，在当前程序中就可以使用 stdio.h 文件中所定义的各个库函数。

程序的主体由一个或者多个函数构成，其中必须有一个 main 函数。在本例中 main 是函数的名字，表示“主函数”。函数体由花括号“{ }”括起来。本例中主函数内有两个语句，程序第 4 行是一个输出语句，printf 是 C 编译系统提供的函数库中的输出函数。printf 函数中双撇号内的字符串“Hello World!”按原样输出。“\n”是换行符，在输出“Hello World!”后，显示屏上的光标位置移到下一行的开头。光标位置称为输出的当前位置，即下一个输出的字符出现在此位置。每个语句最后都有一个分号，表示语句结束。

在程序中，行的右侧如果有“//”，则表示从“//”到本行结束是“注释”， 注释不影响程序的运行，用来对程序有关部分进行必要的说明。C 语言允许用以下两种注释方式：

（1）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