高等学校
电子信息类
规划教材

冯博琴　刘路放　主编

精讲多练

C语言

西安交通大学出版社

高等学校电子信息类规划教材

精讲多练C语言

冯博琴　刘路放　主编

西安交通大学出版社

内 容 提 要

本书是按电子部的《1996～2000年全国电子信息类专业教材编审出版计划》，由计算机教学指导委员会编审、推荐出版的。

本书旨在使学生掌握使用C语言进行应用程序设计的基本技能，着眼于培养学生独立编程的能力和对程序设计语言的悟性。按"精讲多练"教学思路，根据C语言和程序设计特点，全书分为两大部分，各5个单元。第一部分为C的基本内容，包括控制结构、基本数据类型、表达式和函数、编译预处理以及调试技术等，第二部分为C的高级编程技术，包括指针应用和高级数据结构。每一个单元均有7个主题，方便教和学。本书还配有配套的教学软件和投影胶片。

本书可作为大专院校程序设计语言课程的教材，也可供工程技术人员参考。

(陕)新登字007号

高等学校电子信息类规划教材

精讲多练C语言

冯博琴，刘路放　主编

责任编辑　陆诗娣

责任校对　舜　华　高国强

西安交通大学出版社出版发行

(西安市咸宁西路28号　邮政编码：710049　电话：(029)3268316)

西安市德力彩印厂印装

各地新华书店经销

*

开本：787×1092　1/16　印张：19.75　字数：471千字

1997年9月第1版　1997年9月第1次印刷

印数：1——5 000

ISBN7-5605-0946-0/TP·165　定价：19.00元

前言

本教材系按电子工业部的《1996～2000年全国电子信息类教材编审出版计划》，由计算机教学指导委员会编审、推荐出版。本教材由西安交通大学冯博琴、刘路放担任主编，主审王宇颖，责任编委迟忠先。

本教材的参考学时为50学时，其主要内容含两大部分，共10个单元，第一部分（第1～5单元）是C语言的基本内容，包括基本数据类型、控制结构、表达式和函数、编译预处理以及C的调试技术。学习了这些内容之后，学生应能编写、调试和运行小型的应用程序，并对结构化程序设计方法有所了解，为编写较大型应用程序打下了基础。第二部分（第6～10单元）是C语言的高级编程技术，包括高级数据类型：指针、结构体、共用体和文件的概念、相互联系和使用，学习这些内容之后，学生应能理解和掌握有关概念，具备运用高级数据结构和编程技术来编制较大型和比较复杂的应用程序的能力。每一个单元均按7个主题来组织，即：本单元教学目标、学习要求、授课内容、自学内容、调试技术（前5个单元）、实用编程（后5个单元）、程序设计举例、单元上机练习题目。

使用本教材时应注意"精讲"的授课内容和"多练"的组织方式，尽可能采用现代教育技术和手段，如联机大屏幕投影、CAI、投影仪等，均有利于加强讲课效果和节省学时。本教材有配套的教学软件和教学投影胶片，包括联机演示编辑软件、各单元的课堂讲授内容、例题和课内外上机练习。对于从未学过程序设计语言的学生，建议每单元授课2学时，否则可只讲1学时；上机练习应在讲课之后立即进行，每单元应不少于3～4学时。

顾刚、杨振平、杨棋、常虹分别参加了第1～3单元、第4～6单元、第7～8单元和第10单元的部分初稿编写工作，全书由刘路放统稿。程序设计语言教学改革是西安交通大学计算机系列课程内容和体系改革项目的一部分，得到学校和电子信息学院的支持及指导，电子信息学院副院长钱德沛教授十分关注本书的出版，在此一并表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编者

1997年7月

使用本书的建议

“精讲多练”的教学方法很适合于计算机程序设计语言的教学，本书就是按这种方法组织C语言教学的教材。为了用好本书，在此简要地介绍作者在编写本书时的一些思考，权作使用本书的建议，同时也求教于同行，欢迎不吝赐教。

1. 本书的目标不是简单地定位在C语言的讲授上。作者认为程序设计语言正朝着简单化、方便用户的方向发展；再则，程序设计语言种类繁多，层出不穷，令人应接不暇。因此初学者已无必要花大量时间去“死抠”一种语言的细节或使用技巧，故本书以C为工具，企图讲清楚程序设计的基本技能，使学生领悟程序设计语言“无非是什么”，怎样用它进行编写、调试、运行一个实用性的结构化程序。

2. 基于以上指导思想，本书讲授C时采用“大手笔”，即根据程序设计一般原理和C的特色，把全书分为10个单元。为了便于“精讲多练”的教学安排，每单元都有7个主题：

- **教学目标和学习要求** 这两个主题是为便于师生检验学习效果而设置的。
- **授课内容** 是建议教师课堂讲授的内容，一般而言授课内容是本单元所有教学内容之“纲”，起着联系本单元其它主题的作用。授课内容只讲思路和重点、难点，不可能面面俱到，但对要讲的内容则是浓墨重彩，力求写出内容的精髓。
- **自学内容** 它与授课内容一起组成了本单元的基本教学内容，通常它是授课内容的延伸和继续，由学生课外自学。必须强调的是，自学内容并非不重要，也不能省略。对于从未学过程序设计语言的学生，教师应在授课时间中抽出5～10分钟对这部分内容作些介绍，以引导自学。
- **调试技术** 在第1～5单元有该主题，它用于介绍如何调试、连接和运行C语言程序，强调编程实践是本书的重要特色。第1单元中调试技术中的部分内容可在授课时讲授，其它单元的调试技术一般由学生自学，同时也可以作为学生上机的实验指导书。辅导教师在指

导学生上机时可对这部分内容进行现场辅导。

- **实用编程** 在第6～10单元用'实用编程'代替第1～5单元的'调试技术',该主题介绍了界面、菜单、中断、鼠标、图形等实用编程技术,是编写现代风格的应用程序不可忽视的技术内容。这部分一般也是作为自学内容,也可作习题课或辅导课选讲内容,由于这些内容比较新颖实用,可能会成为学生答疑的重点,因此建议教师在备课时将这部分的例题分析清楚,补足缺少的函数和主程序,上机调试通过。本书配有教师用磁盘,内容包括书中所有例题、习题中的源程序,供教师备课时使用,也可直接用于联机电化教学的演示。
- **程序设计举例** 为了补充授课内容和自学内容部分的例题,设置了该项主题。这些例题与本单元的授课内容、自学或实用编程等部分内容相关,且均有算法和结果分析,是学生复习本单元内容的重复参考资料。
- **单元上机练习题目** 每个单元均配有若干上机练习题目,供学生上机练习用。这些练习题均为程序设计题,初学者一般的做法是先编程,再上机;有了一定经验之后,在写出较详细的伪代码程序之后就可直接上机。本书配备的练习题工作量较大,机时不足时可以酌情选做。

3."精讲多练"是对师生双方的要求,'授课内容'可作为教师"精讲"的建议,其它6项主题均与"多练"有关。多练绝不意味泡机时,必须做与教学目标要求相联系的动手上机练习题;"多练"还要求学生学会独立钻研技术的能力,每单元的'自学内容'就是为此目的设置的。这种组织方式,学生的负担会重一些,但今后他们自己飞起来会轻松一些。因此希望教师把"精讲"、"多练"两方面都要落到实处,才能收到较好效果。

编者

1997年7月

目　录

第 3 单元　数据类型

第 4 单元　表达式与函数

第 5 单元　编译预处理

第6单元 结构体和共用体类型

第7单元 指针的概念

第 8 单元　指针与函数

第 9 单元　指针与结构体

第 10 单元　文件

第1单元 Hello, C!

● 本单元教学目标

介绍C语言程序的基本结构以及在计算机上输入、编译、调试和运行C程序的基本方法和步骤。

● 学习要求

通过本单元的学习，了解C语言程序的基本特点，熟悉C语言程序的编辑、编译、调试和运行的基本过程。

● 授课内容

1.1 C语言是最好的程序设计语言

最初，C语言是作为编写系统软件（如操作系统、编译程序等，由计算机厂商或专业软件公司为某种型号或系列的计算机设计、发行）的程序设计语言设计出来的，例如UNIX操作系统就是使用C语言编写的。由于UNIX是一个非常成功的操作系统，被移植到许多计算机系统中，得到了广泛的应用。这样，C语言也就随着UNIX的发展而为更多的人们所熟悉。

凡是使用过C语言的程序员们都知道，C语言有很多不同于其它程序设计语言的特点，例如：

- 数据类型丰富多采：除了整型、浮点型等基本数据类型和数组外，C语言还允许程序员们定义诸如结构、联合等高级数据类型，用以描述比较复杂的数据对象。特别是C语言的指针类型，功能强大，应用也极为灵活，恰当使用可以简化程序结构，提高运行效率，是C程序的重要特色之一。
- 控制结构简明清晰：C语言的控制结构（包括实现分支、循环和模块的语言成分和语句）完全符合结构化程序设计的要求，程序可读性好，便于调试和维护，很适合于编写大型程序。

- 高效率的目标代码：C语言具有丰富的运算符(达数十种之多)和直接控制计算机硬件的能力，在必要时可以直接对字节、位甚至内存地址和寄存器操作，可以直接调用汇编语言编写的子程序，甚至可以在程序中直接嵌入汇编代码。加之各种C编译程序的设计都很注重考虑效率因素，因此C语言程序编译后生成的目标代码长度短，运行速度快，在各种高级编程语言中它的效率最高。

这些特点使得C语言迅速流行起来，不仅大多数专业程序员将C语言列为首选编程语言，就是在非计算机专业的广大工程技术人员和计算机爱好者中间也出现了许多使用C语言的程序设计高手。C语言的应用不再局限于开发系统软件，而是扩大到包括CAD(计算机辅助设计)、数据库与信息处理、科学计算、实时控制、图形图像处理、网络通讯、人工智能和机器人等几乎所有的计算机应用领域。

当然，C语言也有一些缺点，例如为了提高目标代码效率和编程的灵活性，有意取消了一些编译检查功能，特别是缺乏数据类型的一致性检测和不进行数组下标越界检查。因此，一般来说C程序是比较难于调试的，动辄会因为出现难于查出的运行错误而导致运行结果出错甚至死机，使得初学者往往将调试C程序视为畏途。然而瑕不掩瑜，从综合指标来看，C语言仍然可以说是目前最好的程序设计语言。只要掌握了C语言编程的基本规律，同时尽可能多地在计算机上进行C程序的调试及运行实践，就一定能够学好、用好C语言。

1.2 C程序的基本结构

我们通过下面两个例题说明C程序的基本结构。

[例 1-1] 在屏幕上显示一句话：

Hello,C!

程　序：

```
/*-----------------------------------------
    程序 HELLO.C：在屏幕上显示：Hello,C!
-----------------------------------------*/
main()
{
    printf("Hello, C! \n");
}
```

输　出：Hello, C!

分　析：该程序非常简单，仅由一个主函数构成。在主函数中又只有一个语句，在该语句中调用了库函数printf()，用于在屏幕上输出一个字符串。输出字符串中的"\n"是转义字符(参阅1.5："库函数printf()与scanf()的使用方法")，表示在屏幕上输出到这里时将光标移到下一行的起始位置，以后的输出内容就从这个位置开始显示。

程序的前 3 行是注解。C 语言的注解可以是由"/*"和"*/"括起来的任何文字[①],可以出现在程序中的任何地方，用来说明程序段的功能、变量的作用以及程序员认为应该向程序阅读者说明的任何内容。在将 C 程序编译成目标代码时所有的注解都会被忽略掉，因此即使使用了很多注解也不会影响目标码的效率。恰当地应用注解可以使程序清晰易懂、便于调试，便于程序员之间的交流与协作，因此在自己编写的每个程序中都使用精心撰写的注解是一个良好的编程习惯。

[例 1-2]　计算太阳和地球之间的万有引力。

算法分析：由普通物理知，两个质量分别为 m_1 和 m_2 的物体之间的万有引力与两个物体质量的乘积成正比，与两个物体质心之间的距离 R 的平方成反比：

$$F = G\frac{m_1 \times m_2}{R^2} \tag{1.1}$$

式中的 G 为引力恒量，其具体数值与式中各量的量纲有关。如果取质量的单位为 kg，距离的单位为 m，力的单位为 N，则：

$$G \approx 6.672\ 59 \times 10^{-11} m^3 \cdot s^{-2} kg^{-1} \tag{1.2}$$

因此，只要将太阳的质量 1.987×10^{30}kg 和地球的质量 5.975×10^{24}kg 以及两者之间的距离 1.495×10^{11}m 代入上式，即可算出太阳和地球之间的万有引力。

程　序：

```
/*-------------------------------------------------------------
    程序 GRAV.C: 计算太阳和地球之间的万有引力
-------------------------------------------------------------*/
float grav(float m1,float m2,float distance)
{
    float g, G = 6.67259E-11;

    g = G*m1*m2/(distance*distance);
    return g;
}

main()
{
    float g, Msun = 1.987E30, Mearth = 5.975E24;

    g = grav(Msun,Mearth,1.495E11);
```

① 传统的 C 不允许在注解中再嵌套其它注解。但现在已有许多 C 编译程序放宽了对嵌套注解的限制，例如 Turbo C 设有一个开关参数，将该参数设置为打开状态就可以在程序中使用嵌套的注解了。允许使用嵌套的注解对调试程序相当有利。

```
    printf("The gravitation between sun and earth is %f Newton. \n",g);
}
```

输　　出：The gravitation between sun and earth is 3.54444E+25 Newton.

分　　析：在设计该程序时，我们将计算任意两个质点之间的引力公式单独编写为一个函数 grav()，然后在主函数中调用此函数来计算太阳和地球之间的万有引力。这样做有两个好处：

(1)首先，简化了主函数的编写。在编写主函数时，可以认为已经有了一个用于计算万有引力的现成函数 grav()，只需要按要求填写好参数后调用此函数就可以得到计算结果，和调用标准的库函数方法完全相同。这样，即使 grav()是由其他程序员编写的，我们在不必了解其内部结构的情况下也可以方便地调用它。

(2)如果以后还要计算其它物体之间的引力，例如地球和月球之间的万有引力，就不必再编写相应的程序段了，只需在调用 grav()函数时换上相应的参数即可。按这种方法设计程序就称为模块化程序设计，它是结构化程序设计方法（将在第 2 单元中详细介绍结构化程序设计方法）的一部分。

注意：例 1-2 中调用库函数 printf()的方法比例 1-1 中稍微复杂些。字符串（称为格式串）中的格式说明符%f 表示此处要输出一个浮点类型的数据（就是后面的浮点类型变量 g 的值）。对于不同类型的数据要选用不同的格式符，具体使用方法请参阅 1.5:"库函数 printf()与 scanf()的使用方法"。

由上面两个例题可以看出 C 程序的基本单位是函数。C 语言的函数有三种，第一种是主函数，名为 main()。每个程序中只能有一个、也必需有一个主函数。在运行时，程序从主函数中的第一条语句开始执行。

第二种是用户自己编写的函数子程序，如例 1-2 中的函数 grav()。程序中用户自定义函数的数目不限，当然也可以没有。实际上一个实用的 C 语言应用程序通常都包含许多用户自定义函数，而每个程序模块（函数）的功能单一、结构简单，便于编写和逐个调试。

第三种是 C 语言为用户提供的标准库函数，例如 printf()。为了便于程序员编写各种类型的应用程序，C 语言的开发者已经将许多常用的算法预先编写成函数，然后将其编译后生成的目标文件统一存放在函数库中。这样，程序员在编写程序时如果需要实现这些功能时就不必自己再编写程序实现了，只要直接调用相应的库函数即可。

各种版本的 C 语言提供的库函数的数目也各不相同，一般都在数百个以上。附录 2 列出了 Turbo C 2.0 中一批常用的库函数。

另外，针对某个特殊应用领域，例如 CAD 工程绘图，也有由第三方（即除了提供 C 语言编译软件的公司和应用程序员之外的独立软件开发商）提供的软件包，其中包含了若干面向具体应用领域的库函数。在安装好软件包之后，就可以象使用标准库函数那样使用软件包中的函数了。

1.3　用 C 语言解决实际问题的步骤

面对一个在实际工作中遇到的需要使用计算机解决的难题，怎样着手这项工作呢？下面以设计在科学计算方面的应用程序为例进行说明：

(1) 根据实际问题选用适当的数学模型。这方面的内容其实也是各应用学科的主要研究领域之一。

(2) 根据数学模型选用适当的数值计算方法。例如求解常微分方程组可以选用龙格－库塔方法，求解偏微分方程可以选用差分法或者有限元法等，这方面的知识属于“数值分析”或“计算方法”学科研究的范围。

(3) 根据选定的数值计算方法编写程序。除了非常简单的问题可以直接写出相应的 C 程序之外(在值得使用计算机解决的应用问题中这种情况并不多)，一般都应该采用第 2 单元中介绍的“逐步求精”的结构化程序设计方法来编程。

(4) 反复上机调试程序，直到改正了所有的编译错误和运行错误。在调试过程中应该精心选择典型数据进行试算，避免因调试数据不能反映实际数据的特征面引起计算偏差和运行错误。

(5) 调试通以后，根据实际数据运行程序，得到计算结果。

(6) 对结果进行分析，并将其应用于解决实际问题。

以上过程也可以用框图表示，如图 1-1 所示。

应该说明的是：如果要利用 C 语言开发一个应用系统，例如管理信息系统、数据库应用系统、计算机辅助教学系统或者实时控制系统等，一般要比编写一个数值计算方面的应用程序复杂得多，上面介绍的开发步骤就显得过于简单了。这时要遵循软件工程的方法进行应用系统开发，例如采用瀑布开发模型或者快速原型法。这方面的内容已经超出本课程的范围，有兴趣的读者可以参看有关软件工程方面的书籍和资料。

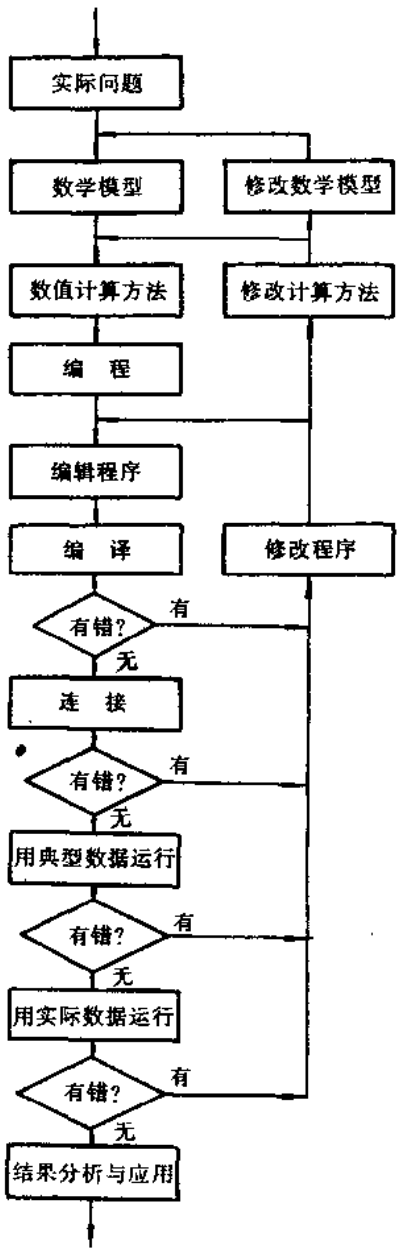

图 1-1　用 C 语言解决实际问题的步骤