



张居敏
飞思科技产品研发中心

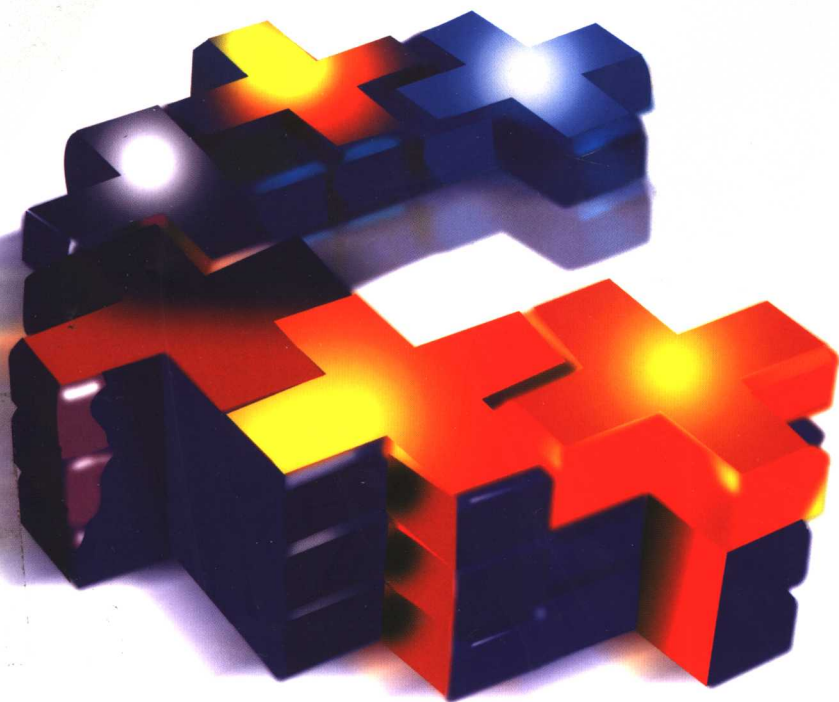
编著
监制

兴趣，才是最好的老师。

让我们在妙趣横生的实例中开始C语言编程之旅

C语言编程 精要12讲

兼论从C到C++的平滑过渡



(本书所用编译环境: Visual C++ 6.0)



电子工业出版社

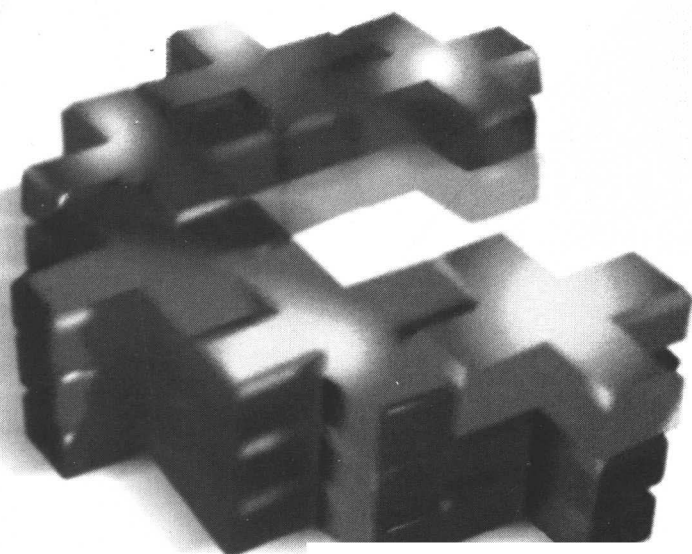
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

张居敏
飞思科技产品研发中心

编著
监制

TP312
2140
2006

C语言编程 精要12讲



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

本书注重学生学习兴趣的培养，而不是枯燥的 C 语言编程技术的介绍。本书用 12 讲的形式科学地融合了 C 语言编程中的核心内容，用众多妙趣横生的实例来表现 C 语言深奥的编程原理，把学习过程变得更轻松，让读者对知识的把握更深刻。

本书涵盖了 C 语言的基本知识，包括编辑环境的使用；还深入讲解了条件语句、for 循环、do while 循环、数组、函数、指针、宏命令和结构体等 C 语言编程的必备知识。同时，本书还考虑到如何从 C 过渡到 C++ 的实际问题，用一定的篇幅提供了该过渡中的基本要领。

本书可作为各类大中专院校的 C 语言教材，或者广大编程爱好者的自学教程。

未经许可，不得以任何方式复制或抄袭本书的部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

C 语言编程精要 12 讲 / 张居敏编著. —北京: 电子工业出版社, 2006.10
ISBN 7-121-03111-6

I. C... II. 张... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 099544 号

责任编辑: 李泽才

印刷: 北京智力达印刷有限公司

装订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京海淀区万寿路 173 信箱 邮编: 100036

开本: 787×1092 1/16 印张: 18.75 字数: 480 千字

印次: 2006 年 10 月第 1 次印刷

印数: 6 000 册 定价: 29.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

本书有三大特点。

◆ **第一大特点：真正适合老师课堂授课与青少年独立自学。**

只要你有一台电脑，在本书的指导下，就完全可以不需要任何人的指导而独立地学会 C 语言程序设计。书中几乎每个例题都配有必要的上机操作步骤和说明，重要操作步骤还配有图解注释，真正做到理论语法学习与上机编程实践的完美结合；并多次介绍程序调试的具体操作过程，“手把手”地教你如何独立排除程序中的各种错误（因为编程过程中错误是不可避免的）。

市场上其他同类书籍的可操作性有待提高，主要问题是讲理论、讲语法者偏多，讲例题时很少涉及具体的上机操作步骤。因此，读者很难把书中的理论性例题转化为可以操作的上机实践，这就造成了青少年的自学困难。程序设计不同于数理化，其理论性、逻辑性都不是太强，但实践性很强。作者认为只要能做到书本理论与上机编程实践的密切结合，就能够独立地学会程序设计。实践性知识只能在实践中获取。例如游泳，不论你看过多少本这方面的理论专著，不下水就不可能学会游泳；作者个人认为程序设计也是一门实践性非常强的课程，要学会编程，就必须上机编程，必须把书上的理论例题转化为某种上机实践。基于这种观点，本书在上机编程实践方面加大了篇幅，几乎每个例题都配有相应的操作步骤。

◆ **第二大特点：内容涉及面广泛，用 12 讲的形式讲解了 C 语言编程的核心内容。**

人是杂食性动物，饮食不仅讲究营养元素的丰富，还要讲求食物色、香、味等的科学搭配，否则再有营养的食物也不受欢迎。爱因斯坦曾举过这样一个例子，说即便是一条贪吃的狗，如果每天都强迫它吃按其生理需求而科学调配的全能型营养饲料，它非但不会健壮起来，相反只会因为厌食而一天天地消瘦。对于知识也一样，人对知识的获取也是多方面的，太单调了就会乏味、败胃口，让人反感。很多程序设计方面的书，它们所包含的语法知识是非常全面的，然而对读者而言，却枯燥乏味，让人难以接受，更难以持久阅读，再“刻苦”的学生也会反感、厌学、敬而远之。

本书充分考虑到了“人是杂食性动物”这一生理特点，在书中补充了很多非程序设计方面的生活常识。例如，在讲解 C 程序的基本输出输入时介绍了速算法中的“首同尾合十”心算原理，介绍了皮鞋鞋码的标识方法及含义，介绍了人肥胖程度的科学界定依据；在讲解条件语句时介绍了青少年未来身高的科学预测方法，介绍了我国严重缺水的国情现状以及阶梯型收取水费的操作方法；在 switch 语句的语法举例中全面介绍了天文历法知识，如公历的来龙去脉；在循环语句中给出了求解任何复杂方程（只要能写出方程表达式）的通用求解程序及其简单的求解原理（二分法），介绍了圆周率 π 的前 22 位巧记法，介绍了信息加密学知识并给出了具有很高应用价值的加密实例，并用 C 程序分析了手机短信恶作剧（测定幸运指数）；在数组一讲中给出了歌咏比赛中常用的评分规则；在函数一讲中介绍了考古学年代鉴定中用到的碳 14 法理论依据，并编写了人-机猜枚大战游戏完整程序；在指针讲节中介绍了 K 金项链含金量的计算方法，真金首饰品的粗略鉴定方法，机密文件的加密、解密原理与方法；在结构体讲节中介绍了事关读者未来切身利益的城镇购房问题，并

用 C 程序分析了具体的购房贷款合同纠纷，还介绍了大学生助学贷款还款方式及利息计算方式等问题。

◆ 第三大特点：始终把对读者兴趣的培养放在首要地位。

每一门学科都是一种艺术，都有能使你入迷的独特艺术魅力。本书作为入门书，作者不求大而全，而求有所为有所不为，旨在展示软件学科的诱人魅力。本书弱化了一部分复杂的语法性知识点，舍弃了过于复杂的专业性语法知识点（当读者积累了一定的编程基础后，如果工作中有必要的话再从别的 C 语言书籍中补充这方面的知识也未尝不可）。本书涵盖了 C 语言语法知识点的 95% 以上，囊括了 C 语言的精华。作者认为，兴趣的培养甚至比知识的获取还重要。从某种程度上来说，天才就是对某个学科过分感兴趣而入迷的“疯子”。

在爱因斯坦的一次生日宴会上，有人在贺辞上说：“他那对人类科学事业的献身精神……”当时他老人家就站出来纠正说，“我并没有那么伟大，做学问时也从来没考虑过献身人类科学事业什么的……我做学问的动机是为了满足个人兴趣，这正如有的人喜欢打牌消遣时光一样，我就喜欢在物理学方面打发时间”。天才就是这样，兴趣是第一要素，一种强大的内在驱动力，只要有兴趣，无师自然通。

总之，本书不仅是一本有关 C 语言程序设计的书，更是一本青年必读性质的各方面生活常识的“大杂烩”。绝大多数 C 语言语法规则都寓教于生活常识性实例应用之中。相信读者能够轻松自如地阅读本书，绝不会感到枯燥无味而无法独立地持续阅读下去。

本书读者定位：理想读者群为初中二年级以上、大学本科生及低年级研究生，或该年龄段的青少年朋友们，或者是对电脑感兴趣的所有朋友们，但主要读者群是青少年朋友们。

编书目的：编写本书有两个目的。第一，带你入门，使你具有独立的自学程序设计的能力；第二，尽量使你对电脑软件的编写感兴趣。自古英雄出少年，培养少年天才。

学习要求：书中的每个例题，就如同钢笔字帖上的每一个规范汉字，读者可以先抄写下来（专心致志地抄，用心地抄，抄是一种临摹），再按书上的编程步骤亲自上机操作一遍，待把所有的例题都这样转化为电脑上的可执行程序之后，相信你就再也不需要临摹而能独立地随心所欲、得心应手地编写 C 程序了。这时候书店里 C 语言方面的书绝大部分你都应该能独立地看懂了，而且具备了相当的自学能力，也极有可能因为编程兴趣而改写你的人生，成为中国未来的比尔·盖茨。

张居敏

2006 年 9 月于江城南湖

 联系方式

咨询电话：(010) 68134545 88254160

电子邮件：support@fecit.com.cn

服务网址：<http://www.fecit.com.cn> <http://www.fecit.net>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

第 1 讲	C 语言编程起步	1
1.1	从比尔·盖茨谈起	1
1.2	为什么要学习程序设计	3
1.3	怎样学习程序设计	4
1.4	用 VC 6.0 编写 C 程序实例步骤	5
1.5	编程步骤小结	11
1.6	编程常用英语四级词汇归纳	15
第 2 讲	C 程序基本常识	17
2.1	C 程序基本结构、变量定义与输出语句	17
2.2	变量的赋值与输出语句应用：一元二次方程通解程序	23
2.3	变量的键盘赋值方法：一元二次方程通解程序的完善	25
2.3.1	输入输出语句的典型应用：鞋码和首同尾合十	30
2.3.2	实数科学计数法：C 语言表示方式	33
2.4	程序中错误的排除方法：肥胖数的测量	36
2.5	程序的单步执行：一元二次方程通解程序的调试	39
第 3 讲	条件语句	45
3.1	if 条件语句的引入：一元二次方程通解程序	45
3.2	if 条件语句及其应用：一论身高预测	48
3.3	if 条件语句程序调试：已知三角形三边求面积	51
3.4	if 语句综合运用：闰年规则的确定依据	58
3.5	if 语句的嵌套：水费计算问题	61
3.6	字符型变量的定义：字母大小写转化	65
3.7	switch 语句及其典型应用：公历历法的计算	70
第 4 讲	for 循环语句	77
4.1	for 循环语句的语法格式：同学聚会握手次数计算	77
4.2	for 语句的应用：程序调试方法	80
4.2.1	for 语句典型应用：抓交通肇事犯	80
4.2.2	C 程序的调试：象棋起源传说	84
4.3	for 循环语句的嵌套：整数解二元一次方程组的 C 程序求解	88
第 5 讲	do{}while() 循环语句	91
5.1	do{}while() 语句的语法格式及功能：巧记 π 的前 22 位特殊典故	91
5.2	do{}while() 语句的典型应用之一：信息加密	98
5.3	do{}while() 语句的精彩应用之二：二分法解无理方程	102
第 6 讲	while() 循环语句	111
6.1	语法格式及功能：数列和是正无穷大验证	111
6.2	手机短信恶作剧案例分析：测试你的性格指数	113
6.3	while() 语句的典型实例：韩信点兵	120
第 7 讲	数组	125
7.1	数组的定义及应用：歌咏比赛评分规则	125
7.2	数组的典型应用：猴子吃桃	128
7.3	字符数组和字符串：特殊的输入输出方式	135

第 8 讲	函数	137
8.1	数学库函数及其用法：古文物年代鉴定依据	137
8.2	其他库函数及其应用	141
8.2.1	随机函数的典型应用之一：电脑随机出算术考试题和首同尾合十题	141
8.2.2	随机函数的典型应用之二：喝酒时人机猜枚大战游戏	145
8.2.3	其他标准库函数及其典型应用：信息加密程序的进一步完善	152
8.3	用户自定义函数：编写方法及应用示例	158
8.3.1	用户自定义函数编写步骤：求解整数方程 $x!+y!+z!=xyz$	158
8.3.2	用户自定义函数的应用举例：再论二分法求解无理方程	165
第 9 讲	指针	169
9.1	指针的语法定义与使用方法	169
9.2	指针应用举例：K 金首饰含金量计算	171
9.3	机密文件的加密与解密方法	173
9.4	机密文件网络传输：加密解密	184
第 10 讲	宏命令	191
10.1	文件包含宏命令含义及应用：用户自定义函数的改造	191
10.2	条件编译宏命令	193
第 11 讲	结构体	199
11.1	结构体定义方法及应用：硕士研究生入学考试成绩单分析	199
11.2	房奴：商品房的奴隶	210
11.3	结构体的应用举例：等本金法还款机理性分析	212
11.4	结构体的应用举例：等本息法还款机理性分析	222
11.5	案例分析：购房贷款合同纠纷	228
11.6	案例分析：大学生助学贷款还款过程剖析	236
第 12 讲	C++语言程序设计初步	243
12.1	C++语言对 C 语言功能的全部继承：二论身高预测	243
12.2	C++程序对 C 程序输入输出函数的改造：三论身高预测	247
12.3	C++中的类：硕士研究生入学考试成绩计算	253
12.4	包含类的 C++程序规范编写过程示例	259
12.4.1	示例一：研究生入学考试例题程序的标准改造	259
12.4.2	示例二：大学生助学贷款还款计划	268
附录 A	常用字符及对应的 ASCII 编码（附表 1 和附表 2）	285
附录 B	常用库函数	287

第 1 讲 C 语言编程起步

本讲首先介绍世界首富比尔·盖茨，他成功的人生经历值得青年人深思。接下来介绍软件行业在未来信息社会中的基础性核心作用，并指出信息社会就是软件社会，就是软件充分渗透到各行各业，并与硬件完美结合的社会。通过对中印两国软件行业的对比分析，指出我国现行软件教育制度中存在的缺陷。最后重点介绍 C 语言程序设计中涉及到的编程软件、具体的示例性上机编程步骤等。

1.1 从比尔·盖茨谈起

我们先了解一位软件行业的鼻祖人物——比尔·盖茨。他的行业至尊地位就如同秦始皇影响中国历史，就如同拿破仑影响法国历史。

比尔·盖茨（以下称盖茨）1955 年生于美国西雅图州，1975 年（20 岁）大学二年级时弃学从商，白手起家创建微软公司。1986 年（31 岁）进入《财富》亿万富豪榜，个人资产约 3 亿美元，成为有史以来最年轻的亿万富翁。1991 年仅次于华尔街大亨巴菲特（Warren Buffett）而成为全美第二富豪。1992 年个人资产约 70 亿美元。从 1994 至今连续十几年雄踞世界富翁排行榜榜首，2005 年个人固定净资产为 480 亿美元。盖茨捐助的基金额可与当今社会最大的慈善基金组织相比。

盖茨 13 岁读初中时开始在学校接触计算机。早在上世纪 60 年代就让这些初中生直接接触计算机，这是相当罕见的。事实证明，对盖茨成才起到决定性——基础作用的就是在这段时间里。他从此开始接触并逐步迷上了自己未来的事业：电脑程序设计。据他的好友保罗·艾伦（后来和盖茨一起创立微软公司，现为世界首富排行榜第 7 名，个人资产 210 亿美元。以下称艾伦）回忆说：“我们当时经常一直玩到三更半夜，我们都爱死了电脑软件工作，那时我们玩得真开心”。

当时，盖茨与同伴艾伦一起从电脑操作手册上学会了 BASIC 语言，并写出了他的第一个软件程序，目的是为了玩三连棋。接着盖茨读了一本有关拿破仑军事战略的书，受此启发，他设计了一种叫做“冒险”的计算机游戏，游戏的目标是“统治世界”。

盖茨和艾伦不久就开始利用晚上时间为一家当地公司干活。这家公司买了一台 DEC 计算机，盖茨、艾伦及其他孩子们被雇来寻找电脑程序中的错误（又叫“臭虫”）。更好的说法也许是公司“允许”他们玩这台电脑，因为孩子们得不到任何报酬，纯粹是为了好玩儿。这家公司之所以愿意这样做，因为它与电脑销售商签有合同：只要能找出 DEC 软件中的错误，就不必为使用 DEC 电脑而付费。由于该程序软件是新开发出来的，很复杂，难免存在不少错误（但他们与客户之间签署这种合同只是软件行业的一种惯例：意在发现软件中最细微的缺陷）。半年下来，这些初中生们收获甚丰，盖茨的成绩尤为突出。他们根据自己的发现，为软件中的错误列举了一份长达 300 页的《问题报告书》。DEC 公司大为震惊，并

发现这帮初中生们总能挑出毛病，于是不得不终止合同。

比尔 16 岁读高中时英特尔 8008 芯片问世，他和艾伦一起花 360 美元买了一块，用它启动了一台机器，以便分析从城市道路交通监视器传来的信息，并成立了一家数据公司。但结果以失败而告终：市政当局不屑于从这两个高中生手里购买产品。

盖茨在 9 年级时候开始了一种新的叛逆。他的学习成绩并不算好，但他决定在不带一本书回家的情况下使每门功课都达到 A 级水平。他不再去上数学课了，因为已经自学了很多内容，并在全国性的能力测试中名列十佳，这也使他学会了独立。到了 10 年级，他开始在学校传授计算机知识，并为校方编写学生座位排序软件。

这就是盖茨的中学生活。这期间他自学并积累了很多程序设计方面的理论知识与实践经验。

盖茨 18 岁中学毕业，并以全国资优生身份考入哈佛大学。在大学里他仍然无法抵抗电脑的诱惑，经常逃课，一连几天呆在电脑实验室里整晚整晚地写程序、打游戏。后来他从 MITS 公司的 Altair 机器受到启发，看到了电脑商机及其未来电脑的发展趋势，于是就给该公司打电话，说可以为 Altair 机器提供一套 BASIC 编译器。得到回答说：“我们每天都收到很多类似的来信和电话，但不论是谁，首先写完该程序者就能得到这份工作”。于是盖茨整整 8 个星期，一直在学生宿舍里没日没夜地编写、调试程序。最后终于成功了，两个月通宵达旦的心血和智慧产生了世界上第一个 BASIC 编译器，MITS 公司对此也非常满意。

后来，盖茨敏感地意识到，计算机行业发展太快了，等到大学毕业，他可能就失去了一个千载难逢的巨大商机。于是在 1975 年 6 月（大学二年级），盖茨经过认真的考虑，说服了自己，说服了万分震惊的父母，毅然决定放弃这所世界上最好大学的毕业证书，弃学创业。

他和好友保罗成立了微软公司，在一家灰尘弥漫的汽车旅馆中租赁了一间办公室，开始了艰苦的微软创业旅程。他们挤在这个杂乱无章、噪音纷扰的小房间里，没日没夜地写程序，根本就不知道什么是疲倦和劳累。

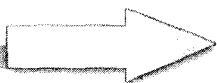
盖茨就是这样独自白手起家，艰难地、一步一步地打天下的。

盖茨向来以工作狂而著称。尽管微软公司向来以员工习惯于拼命加班而闻名，但员工们还是心悦诚服地说，他们之中没有谁比盖茨更辛苦。即使到了 39 岁结婚时，他还经常加班工作到晚上 10 点钟以后，对于以前任何一个世界级富翁来说，这都是史无前例的。

这就是比尔·盖茨。我们现在使用的计算机 Windows 操作系统软件就是由他及他创办的微软公司设计的。从盖茨艰苦创业辉煌的人生经历中我们可以得到几点启示。

1. 英雄出少年。一个人的成功往往与其少年时候积累下来的相应能力和专业的功底分不开。比尔在中学时就练就了过硬的软件编程基本功。他自学能力强：从 Basic 手册中自己学会了 Basic 编程，甚至还自学了数学中的很多内容，并在全国能力测试中名列前十名，这也使他学会了独立；他动手能力强：13 岁时就自己编写简单的游戏软件，16 岁开始用软件知识解决实际问题（成立了数据公司），并为校方编写学生座位排序程序等。理论联系实际不仅造就了他出色的动手能力，更造就了他关注社会问题，关注行业发展，造就了他敏锐的社会洞察力，使他敢于果断放弃大学生涯，毫不犹豫地抓住了人生的重大历史性机遇。

2. 实践性知识只能在实践中获取，只能通过实践来获取。比如游泳、拉二胡等，任何高深的理论，任何精彩的理论讲解都代替不了实践环节。计算机编程就是一门实践性很强



的课程，它不需要高深的数理化知识作基础，初中生的综合能力包括英语水平就足以入门了。然而我国大学计算机系的毕业生，不能独立写出一行完整程序代码者比比皆是。可见，我国在这一专业中的教育，理论与实践的结合是有脱节的。

我国之所以没有比尔·盖茨这样的软件天才，我认为有两个原因。第一是“计算机教育要从娃娃抓起”的指示没有真正落到实处，学生涉及软件方面的知识比较晚，英雄无法出少年；待到大学毕业就业时再参加软件编程集训等，可以说要想成就一代霸业，这可能为时已晚。第二是上机实践环节不够重视，培养软件人才的方式多以课堂讲解、纸上谈兵为主，很多本来比较容易的软件知识都因此而披上了高深莫测的神秘面纱，迫使青少年望而生畏。

1.2 为什么要学习程序设计

在未来社会里，不懂软件有可能就是一种新型文盲。因为现今社会，正在完成由工业时代向信息时代的转变，与信息技术无关的职业已从1970年的95%降到2000年的35%，到现在已经近乎为零。

未来的信息社会是一个软件主宰天下的社会。硬件是信息产业的筋骨，软件是其灵魂。信息的搜集、加工和存储等都必须由相应的软件来实现。软件产业是信息技术、通信技术、管理技术等多种现代科技的物质基础，它能极大地推动传统产业的改造和装备水平的提高。在智能化控制方面，数字化技术革命将使几乎所有的产品及设备中都可以嵌入软件：冰箱、彩电、洗衣机、手机等电器产品，汽车、拖拉机等机械产品无不朝着自动化、智能化方向发展。任何机械产品，实现智能化控制的惟一出路就是软件系统的嵌入，程序控制是智能控制的惟一出路。与此同时，产品的设计（CAD 绘图）、测试（虚拟仪器）和制造（数控机床）等也将依靠软件来进行。总之，未来社会里各行各业都将越来越依赖于软件产品，软件主导硬件是必然的趋势。而软件的开发、应用和普及不仅需要一定的技术和设备条件，更需要有一大批高素质的软件工程师、测试工程师、程序开发工程师等人才。

我们的邻国印度十分重视软件人才的培养和软件工作。印度也有着数千年文化传统，对外开放比中国晚了10年，国民生产总值及人均值不及中国的1/2，吸引外资也只有中国的1/10左右，但他们在全球软件方面已经创造了一个神话：软件出口仅次于美国，成为世界第二大软件出口国，软件产品出口的规模、质量和成本等3项硬指标均居全球第一。当前，全世界获得IT业最高等级CMM5级证书的企业只有75家，而印度就占了40家。2005年比尔·盖茨在访问印度时曾说，印度的软件人才令他感到吃惊，要说最能充分利用数字时代优势的国家，非印度莫属。

上世纪80年代以来，印度政府一直把培养信息技术人才和发展信息技术产业作为长远战略加以实施。认为这一产业“能够为贫困者擦去眼里的泪水，能够改变整个国家和民族的命运”。首先，印度政府很重视计算机教育，且这方面的教育从小孩就开始进行。教育部门制定了统一的中小学计算机教学大纲，甚至对中小学数学教材的内容也作了相应调整：从中学开始数学考试中就很少出现选择题了，而多以综合分析题为主，旨在促使学生进行完整的思考，强化其逻辑思维能力的培养。大多数印度软件工程师都认为他们在软件方面的强大实力主要得益于数学。当然印度以前是殖民地国家，作为官方语言的英语相当普及，

这也为其软件业的发展扫清了语言障碍。

而在我国，从小学到中学乃至大学，计算机课程在不少人眼里都是些陪衬或摆设，而且，在计算机的普及以及专业教育中，大都忽略了程序设计技能方面的实践性上机编程训练。

如何抓紧提升我国的软件工作，使更多的人懂得软件，我们有以下几点启示。

首先，计算机知识普及要“从娃娃抓起”。要完善中小学计算机教材，注意教材的及时更新，消除教材中缺少程序设计方面内容等不正常现象。我们应该在中学就立足培养优秀的软件程序设计人才，而不是在大学生难就业时再实施这方面的社会性编程集训等。

再者，就是要明确各级学校学生的程序设计学习任务。比如，中学生必须掌握一到二门程序设计语言，而计算机系的大学生必须全面掌握高级程序设计语言（C++、DELPHI等）、网络开发语言（JAVA、ASP.NET等）、数据库语言（SQL）等，以及一定的软件系统分析能力。

还有就是更新教育观念。我们在所谓的主科与副科之间，在高考科目与非高考科目之间应该寻找一些平衡点。我们在培养高考状元的同时，也要加强计算机软件教育，避免埋没这些可能在软件行业里有很大发展潜力的少年天才。

1.3 怎样学习程序设计

微软公司希望新进员工，要有过书写几十万行程序代码的实际经验。为什么？道理很简单，实践性的知识只能在实践中获取：程序设计是一门典型的实践性课程，只能在实际上机编程过程中提高自己的编程能力；相反，学编程的不上机，就好比是学游泳的不下水，效果可想而知。

学习任何一门课程，入门很重要。什么书最好？自己能看懂，能从中获取知识的书最好。编程方面的书大体可分为三类：教材、初级入门书和能力提高书。教材是教师课堂讲授用的材料，是为教师有效组织课堂教学而编写的书，不适合学生自学；初级入门书比较好，但要以自己能看懂，具有可操作性为准则，很多这方面书的一个共性缺点就是只讲编程语法规则，不讲在电脑上的具体操作步骤，这就是可操作性很差的书；还有，就是能力提高的书，它假定读者已经有相当的基础，因此不适合初学者入门之用。

学书法首先要找一本字帖，学绘画也要从临摹开始。同样的道理，学习程序设计，也要从“临摹”入手。首先把书上的每个例题都逐个的认真抄写两遍，争取每个例题都达到不看书而能够默写出来的水平。抄，本身就是一种临摹。宋代文人苏轼“过目不忘”的本领就源于他三抄汉书之苦功；再把抄写下来的例题程序逐个地都输入电脑中，将其转化为一个个可以执行的程序，并通过程序的单步执行尽量理解其运行机理（即执行顺序），慢慢地自然就入门了。很多初学者容易忽略上机实践这一环节，这是大错特错的。只有通过实践，你对程序运行机理的理解深度和广度才能提高，才能激发并培养编程兴趣，进而激活这方面的天赋与潜能，否则只能是越学越没意思，越学越没兴趣，最后不得不敬而远之。

记得我当年学习程序设计时，没有人指导，就自己到图书馆借两本勉强能看懂的入门书，拿到宿舍里在电脑桌上摊开，打开电脑，一边看书，一边尽可能地把书上的程序代码输入电脑，转化为可以执行的微程序段。这不是为了验证书上是否有排版印刷或讲解错误，而是为了加深理解，为了获取更深层次上的编程知识。就这样，我不但很快学会了编程，

而且也极大地培养了自己在这方面的兴趣。

惟有通过实践，才可以加深理解，才可以培养兴趣。

总之，学习程序的窍门是：抄例题+默写例题+上机操作=最短时间学会C程序设计。

微软创始人比尔·盖茨，包括他的伙伴艾伦·保罗，可以说他们的编程知识基本上都是靠自学靠实践摸索出来的，绝不是有哪位知识渊博的老师传授给他们的。这方面的真理只有一条：只要有兴趣，无师自然通；而且还可以说，兴趣的培养比知识的获取更重要。

第一，要有一台可以自己支配的电脑。第二，就是要安装一个VC 6.0编辑器软件。对于C语言程序设计，到市场上买一张Visual C++ 6.0光盘就可以了，英文版本和中文版本的都可以（实际上中文版本的VC 6.0仅仅是主菜单被汉化了了的英文版）。这里要说明一下，本书所有的程序例题采用的都是Visual C++ 6.0英文版本的编辑器。买到软件后再安装一下就可以正式开始编C程序了。第三，就是到大书店挑选一本操作性很强的C语言程序设计入门性指导书。一本既讲理论，又讲怎样把例题输入电脑的附有必要操作步骤方面的书。第四，就是一边看书，要一边上机实践。

1.4 用VC 6.0编写C程序实例步骤

下面先用VC 6.0编写一个简单的C程序。假定你的电脑上已经安装好VC 6.0英文版本的编辑器，至于如何安装这个软件，相信大家按照光盘上的安装步骤都会安装成功的，这比游戏软件的安装简单得多。安装步骤如下。

第一步，打开VC 6.0编程界面，如图1-1所示。用鼠标依次单击以下命令：**【开始】**→**【所有程序】**→**【Microsoft Visual Studio 6.0】**→**【Microsoft Visual C++ 6.0】**。

图1-1所示为Windows XP操作系统。若读者用的是其他操作系统，则**【开始】**菜单的内容可能有所不同，但最终命令是相同的，执行**【Microsoft Visual Studio 6.0】**→**【Microsoft Visual C++ 6.0】**命令。

至此，我们打开了VC 6.0编程界面，如图1-2所示（已经关闭“Do you know...”对话框）。

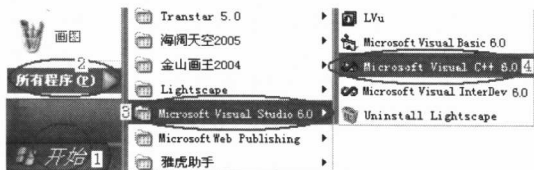


图 1-1 打开 VC 6.0 编程界面

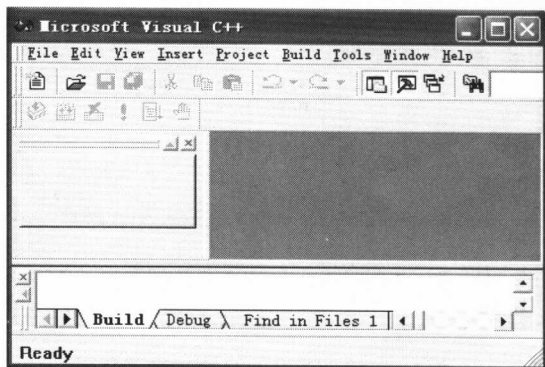


图 1-2 VC 6.0 编程界面

第二步，建立 C 程序文件。在图 1-2 所示的界面中，依次单击以下菜单命令，如图 1-3 所示：执行【File】→【New...】（【文件】→【新建...】）命令。

在弹出的对话框窗口中选择相关选项，如图 1-4 所示。图中标出了选择顺序，输入文件名称，注意一定不能漏掉 C 程序的文件名后缀“.c”，如图 1-4 中所示输入的文件名称为顺序 3：【one.c】。这一步最关键！

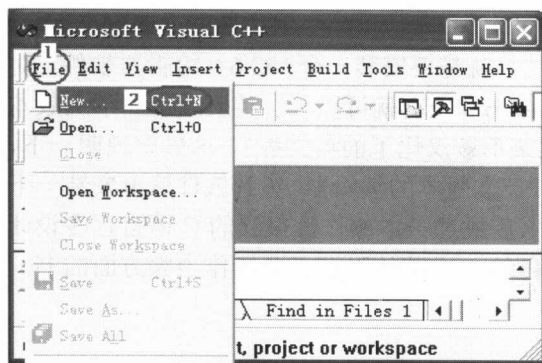


图 1-3 建立 C 程序文件

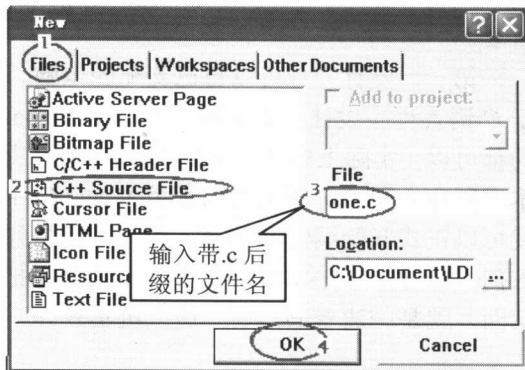


图 1-4 选择相关选项

第三步，编写并运行标准的 C 程序框架代码。在随后激活的界面窗口中通过键盘手工输入以下 C 程序标准的框架代码，如图 1-5 所示。注意不要输错（所谓程序标准框架代码，就是每个 C 程序都必须惟一拥有 C 程序代码）。

```
#include<stdio.h>
main()
{
}

```

接下来需要编译、运行框架程序，如图 1-6 所示。然后执行【Build】→【Compile one.c】命令。

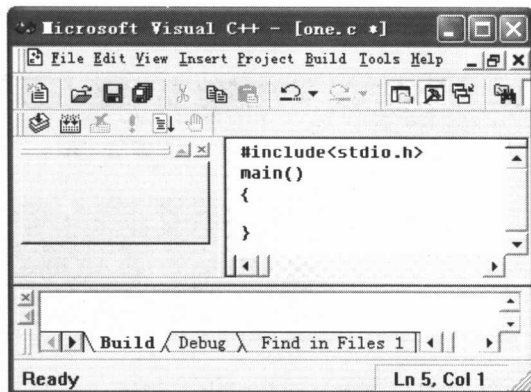


图 1-5 输入标准的 C 程序框架代码

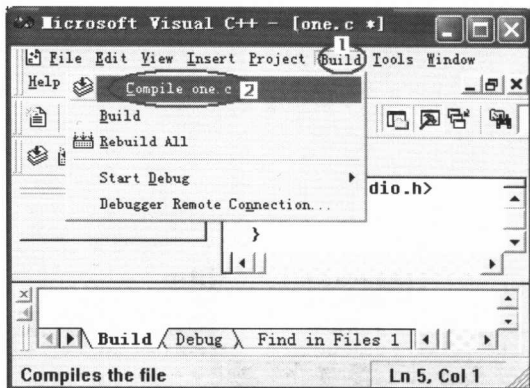


图 1-6 编译 C 框架程序

关闭随后弹出的两个对话框，如图 1-7 和图 1-8 所示。

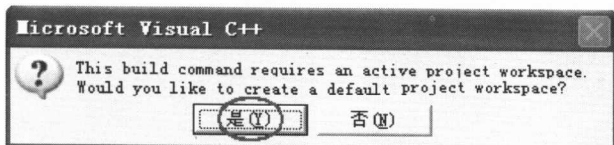


图 1-7 单击【是】按钮关闭该对话框

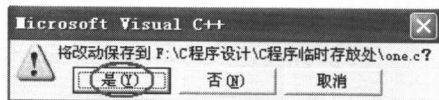


图 1-8 单击【是】按钮关闭该对话框

框架程序最终的编译结果如图 1-9 所示，没有错误，没有警告（0 errors, 0 warnings）。



注意

如果编译结果（输出区）显示有错误或者警告，说明代码输入有错误，如果是初学者可删除所有的程序代码，再次输入代码，然后再编译（重复图 1-5、图 1-6、图 1-7 和图 1-8 所示步骤），直到输出区显示编译结果没有错误、没有警告（0 errors, 0 warnings）为止。

编译后还需要运行 C 的框架程序，执行【Build】→【Execute one.c】命令，如图 1-10 所示。关闭随后弹出的对话框，如图 1-11 所示，则程序框架运行结果如图 1-12 所示。

Press any key to continue！（按任意键继续），按键盘上的【Enter】键即可返回到代码编辑窗口。

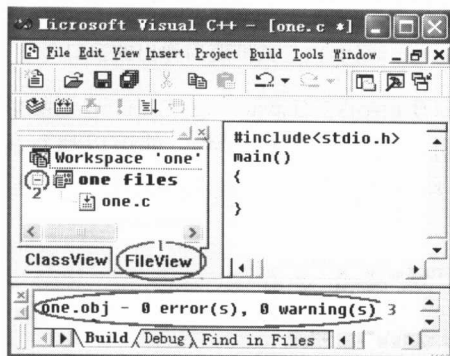


图 1-9 框架程序的编译结果

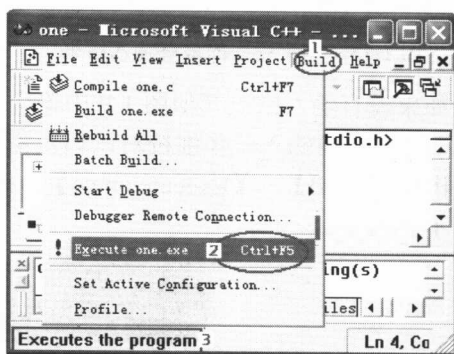


图 1-10 运行框架程序

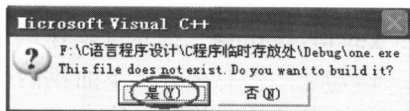


图 1-11 关掉这个对话框

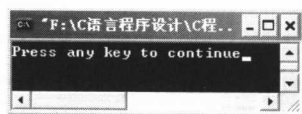


图 1-12 框架程序的运行结果

至此，整个编程的前期准备工作已经完成，接下来就可以正式编写 C 程序了。下面就编写一个简单的 C 程序。

例 1.4-1 向屏幕输出如下语句：I love computer

按【Enter】键，关掉图 1-12 所示的窗口，在图 1-9 所示的窗口编辑区里的大括号内添加以下程序语句（其中加黑加粗部分为新添加的代码，其余为原框架代码），注意不要有输入错误。

```
#include<stdio.h>
main()
{
    printf("I love computer ");
}
```

添加代码后窗口如图 1-13 所示。

编译并运行程序：执行【Build】→【Compile one.c】（【组建】→【编译 one.c】）命令，关掉随后弹出的窗口，如图 1-14 所示。

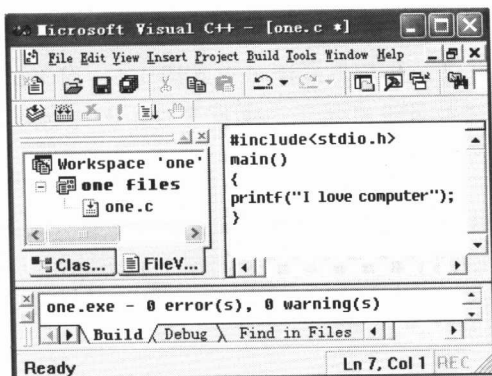


图 1-13 例 1 程序的最终输入结果

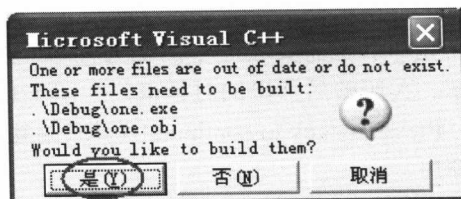


图 1-14 关掉随后弹出的窗口

如果没有输入错误（如图 1-9 中输出区中所示：0 errors, 0 warnings），否则删除该行代码，再次重新输入或仔细校对以纠正错误，之后再次编译并运行程序），就可以执行程序了。执行【Build】→【Execute one.c】命令，（如图 1-10 所示），就可得到程序的运行结果，如图 1-15 所示，

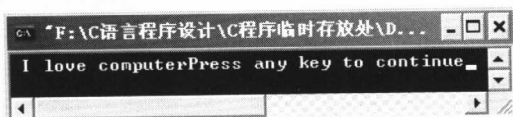


图 1-15 例 1 程序运行结果

屏幕上除了显示 C 语言框架固有的“Press any key to continue”提示性操作语句外，还显示了语句：I love computer。美中不足的是这两条语句之间没有分开。下面我们再在原来输出语句（I love computer）的末尾添加两个字符：\n（其作用是换行。注意\n不要误写为/n或者ln等，否则起不到换行的作用）。修改后程序代码如下（加黑加粗部分为新添加代码，其余为原程序代码）：

```
#include<stdio.h>
main()
{
    printf("I love computer\n");
}
```

执行【编译】→【执行 one.c】命令，关掉随后弹出的对话框。添加以后的运行结果如图 1-16 所示（与图 1-15 运行结果对比可知\n的作用是换行）。



图 1-16 添加\n后的运行结果

对程序解释如下：

本程序实际上只包含一句程序代码：printf("I love computer\n")。

除了上面一行代码外，其余全部是C程序框架代码（每个具体的C程序都必须惟一拥有通用代码）。

printf：为 print（打印）与 function（函数）的缩写，打印函数的功能就是把其后小括号中的双引号内的所有字符都“打印”到显示器屏幕上。

下面再编写几个简单的C程序。

例 1.4-2 求计算式 45×78 的值。

关掉图 1-16 所示的窗口，删除图 1-13 所示框架内程序代码：

```
printf("I love computer\n");
```

重新输入如下所示加黑加粗代码，注意不要有书写错误，特别是最后一行代码出错的可能性很大。

```
#include<stdio.h>
main()
{
    int a , b , c ;
    a=45;
    b=78;
    c=a*b;
    printf("%d\n" , c);
}
```

添加代码后窗口如图 1-17 所示。

编译并运行程序：执行【Build】→【Compile one.c】命令，关掉随后弹出的对话框，如图 1-18 所示。

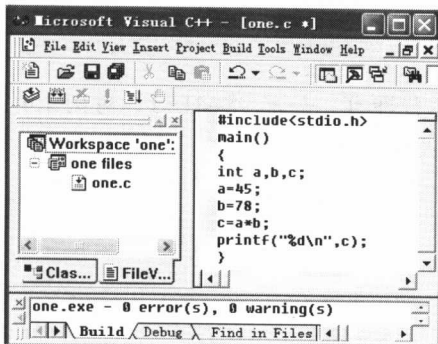


图 1-17 例 2 程序的最终输入结果

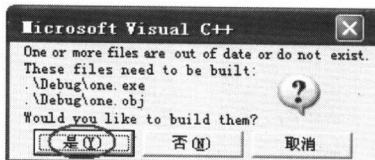


图 1-18 关掉该对话框

如果没有输入错误，就可以执行程序了。执行【编译】→【执行 one.c】命令，程序运行结果如图 1-19 所示，即 $45 \times 78 = 3510$ 。

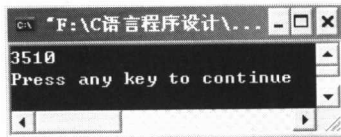


图 1-19 例 2 程序运行结果，乘积 3510

按键盘上任意键（一般是【Enter】，或者空格键），即可返回编辑窗口（“Press any key to continue”）。

注意

1. 第一行代码：int a,b,c; 中 int 和 a 之间必须加一个空格，不能连写为 inta,b,c。
2. printf: print function, (标准) 打印(输出) 函数，作用是向显示器输出变量值，其中双引号内的\n 不要误写为/n。
3. 如果对程序运行时提示有错误而不能执行，可以试着删除大括号内的所有代码而重新输入。
4. 以上所有的输入都不能在任何汉字输入法中进行，而只能在如图 1-20 所示英文输入法中进行。

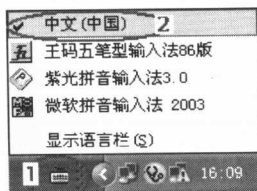


图 1-20 程序代码的输入不能在任何汉字系统下输入

初次编程时错误是不可避免的，因此要有足够的耐心，一旦入门后就顺手了。

练习题

仿照例 1.4-2 程序，分别求下面计算式的值： 89×98 ， $45 + 65$ ， $98 - 41$ 。

提示

对上面程序代码略加改造再运行即可，如求 $98 - 41$ 时，对例 1.4-2 代码改造如下（加黑加粗部分为改造代码，其余为原代码）：

```
main()
{
    int a , b , c ;
    a=98;
    b=41;
    c=a-b;
    printf("%d\n" , c);
}
```