高等学校计算机专业规划教材

C语言程序设计教程

苏小红 陈惠鹏
温东新 李秀坤 编著
王宇颖 主审

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等学校计算机专业规划教材

C语言程序设计教程

苏小红　陈惠鹏
温东新　李秀坤　编　著

王宇颖　主　审

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是新世纪高等学校计算机专业规划教材之一。

本书由9章组成，内容主要包括：C语言程序设计基础知识、简单的C程序设计、程序的控制结构与结构化程序设计方法、函数与模块化程序设计方法、数组与指针、结构体与共用体、关于函数应用的高级话题、文件操作、图形和声音的制作以及七个附录等。

为了提高读者的学习兴趣，本书在例题、习题和实验题目的选择上作了精心的安排，不仅使其具有实用性，而且具有趣味性，同时采用启发式的写作风格，不易理解的概念和算法采用打比方的方式进行类比说明，以提高读者的分析问题和解决问题的能力。

本书可作为高等院校计算机和非计算机专业的教科书和参考书。

图书在版编目(CIP)数据

C语言程序设计教程/苏小红等编著 .—北京：电子工业出版社，2002.2
(高等学校计算机专业规划教材)
ISBN 7-5053-7478-8

Ⅰ.C… Ⅱ.苏… 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(2002)第005521号

责任编辑：陈晓明　　特约编辑：李双庆
印　　刷：北京东光印刷厂
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京市海淀区万寿路173信箱　邮编100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：22　字数：563千字
版　　次：2002年2月第1版　2002年2月第1次印刷
印　　数：8000册　定价：26.00元

凡购买电子工业出版社的图书，如有缺损问题，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

《高等学校计算机专业规划教材》

编委会名单

出 版 说 明

中国上海计算机学会教育专业委员会受国务院信息产业部的委托，为了适应我国21世纪计算机各类人才的需要，根据学科技术发展的总趋势，结合我国高等学校教育工作的现状，立足培养的学生能跟上国际计算机学科技术发展水平，于2001年4月参照IEEE和ACM计算机教程2001大纲组织编写与其配套的22种教材，现推荐给国内的院校，作为教学之用。

为了使这套教材体现现代计算机教学的特点，编出特色，来自上海交通大学、复旦大学、国防科技大学、哈尔滨工业大学、华东师范大学、东南大学、华东理工大学、上海大学、重庆大学、东华大学等十几所大学的专家、教授成立了以陈火旺院士为主任委员的编写委员会，并多次集中开会，在研究、学习、借鉴2000年6月ACM和IEEE/CS联合专题组发展的《Computing curricula 2001》报告的基础上，深入讨论了结合我国高等学校计算机本科教育的实际而推出的"93教程"的教学情况以及《2000计算机学科教学计划》，结合当前计算机技术飞速发展的现实——对计算机学科的教学内容不断提出更新的要求，特别是为了全面推进素质教育，以及培养学生的创新精神和实践能力，提出了新的编写思路，使这套教材的知识点建立在"高"、"新"的平台上，反映当前计算机科学发展的前沿和趋势。

ACM和IEEE2001教程的思想是将计算机学科领域的知识分解为几个主要的科目（算法与数据结构、计算机体系结构、人工智能与机器人学、数据学与信息检索、人机通信、数值计算、操作系统、程序设计语言、软件工程）并作为学科的公共要求；对计算机学科的教学归结为理论、抽象和设计三个过程，并强调教学一定要与社会需求相结合。另外，还提出了贯穿于计算机学科中常出现的基本概念，并将这些概念在教材中予以清晰的介绍，灵活的应用，以更好地帮助学生，使之成为一个优秀的计算机工作者。

为了保证这套教材的审编和出版质量，以陈火旺院士为主任委员的教材编委会的专家教授们在2001年4月召开了全体编委、作者讨论会，制订了编写要求和编审程序。编委们对所有教材的编写提纲进行了讨论，对教材的质量做了专门的要求，并设立专门的负责人选。参加这套教材的编审者都是来自全国重点高校的在计算机领域从事教学和科研的专家和学者，他们具有丰富的教学经验，严谨的治学态度，较高的学术水平。

这套教材的出版得到电子工业出版社的积极支持。他们把这套教材列为重点图书出版，并制订了专门的编审出版规定和出版流程，组织了专门的编辑力量和协调结构。

我们希望这套教材的出版，对我国的计算机教育事业的发展做出应有的贡献。

编委会

2002.1

前　言

初学者经常发现，学习第一门编程语言（尤其是C语言）是一个充满挫折的艰难历程。一方面，抽象的算法与程序设计过程使他们感到用C语言进行程序设计高不可攀，学习C语言索然无味，学习时总是提不起精神来。另一方面，调试程序时所遇到的各种困难又使他们对C语言望而却步，使其学习C语言的计划常常半途而废。即使勉强坚持学完了，也不十分清楚C语言究竟能帮助他们解决现实生活中的什么问题，或者即使能用它去解决一些问题，但总是发现自己编写出来的程序晦涩难懂，隔一段时间后，有时连自己都看不明白了。

本书在介绍C语言的同时，将试图为读者解答这些疑惑。本书作者有着近十年的教授C语言的经历，在授课中发现了在学生中普遍存在的上述问题，在自己的多年教学实践中，曾尝试过用各种方法去帮助学生来解决这些困难，有过失败的教训，也有一些成功的经验。现在，作者将本书奉献给广大的读者，希望读者在用本书作为教材学习C语言时，不再有望而却步的感觉，相反，可以在一种轻松、愉快的气氛中探索程序设计的奥妙。如果读者您对C语言已有了一些了解，相信本书中那些充满趣味性和实用性的实例一定会让您爱不释手，那些巧妙地贯穿于实例中的关于程序设计方法的介绍也会让您耳目一新，对您有所裨益。

程序设计是一项细致的活动，主要是熟练地掌握和使用编程语言。当然，它也是一项逻辑活动，可以完全独立于具体的编程语言，不受它所依托的具体语言的限制。它涉及到很多认识上的技巧，例如，对操作环境和相关开发工具的熟悉，对数据结构、算法的合理运用和测试，对机器内部工作的了解等等。因此，以往我们在介绍程序设计语言时那种只侧重语法知识介绍的教学方法常会给学生一个误导，使学生误以为学习程序设计就是记住那些语法规范而已，因而，常常去赶时髦，一味地追求多学几门程序设计语言，而忽略了对程序设计方法的掌握，忽略了好的程序设计习惯的培养，忽略了缜密思维方式的训练，就好比学习演唱时，忽略了对乐理知识的学习以及发声的训练，一味地追求多学几首歌一样。本书的最大特点就是尽量避免生硬的说教，用这种打比方的浅显易懂的方式将有关程序设计方面的许多知识自然而然地娓娓道来。

为了您阅读方便，并尽快了解本书的全貌，下面对本书的特色作简要介绍。

1．编写本书的一个首要的出发点就是：不仅要使它成为大学本科学生学习C语言的一本合适的教材，还要使它适合那些想要自学的读者朋友们。因此，本书在内容组织与讲解方面作了精心的安排。主要有以下几个方面：

（1）结合简单直观的图示进行难点内容的讲解，因为很多时候，看一张图要比看一段话更直接、更清楚得多。例如，在介绍简单变量和指针作函数参数的区别时，讲再多的话，学生也理解不了，可是画两个图再讲解，其中的差别之处立刻就一目了然了。还有许多不易理解的算法（如排序、查找、插入、删除等）也结合了丰富的图示进行讲解，相信读者一看就会明白的。

（2）语言叙述通俗易懂，一些易混淆和难理解的概念尽量通过打比方的方法来进行类比讲解。例如，算法与编程语言、直接寻址与间接寻址、二维数组的行地址与列地址等都是用了比较贴切的比喻来形容它们之间的联系与差别。

（3）章节安排由浅入深、循序渐进，前后章节内容衔接紧密，难易程度过渡自然。例如本书是围绕着结构化与模块化程序设计这个中心，沿着数据结构从简单到复杂这样一条主线来逐步展开有关函数内容的介绍的，这样既可将难点问题分散开来，又可做到在内容上的温故而知新，不仅突出了我们以C语言为依托介绍程序设计方法的宗旨，还强化了使用函数进行编程的训练。

2．编写本书的第二个出发点就是要提高学生对学习程序设计的兴趣，让学生体味到学习程序设计不再是一件枯燥乏味的事情，而真正感到“乐在其中，用在其中”。

根据我们多年从事C语言教学的经验，发现例题的选择尤其重要，因此，我们将书中的程序实例视为学生饭桌上的菜肴，力求将其做得不仅营养丰富，而且符合学生胃口，这样，学生才会乐于去品尝，吃得饱、吃得香。因此，本书在例题、习题和实验题目方面作了精心的挑选和设计，这些程序主要来源于生活，都是发生在我们身边的最熟悉的现实问题，不但内容丰富，涉及面广，而且生动有趣。另外，书后习题题型丰富，有针对性，且每道题都给出必要的提示，启发读者的思路，帮助读者自学和练习，使读者打消对程序设计的畏惧心理。

3．编写本书的第三个出发点就是要加强读者的缜密思维方式的训练和勇于推陈出新的能力的培养。因此，不同于其他书籍，本书的绝大部分程序实例都是以“一题多种解决方案、一题多种编程方法”的形式出现的，让读者在程序设计时不局限于一种解题思路和一种实现方法，在程序的设计与编写过程中加深对各种语句功能、语法规范、程序结构以及编程方法和技巧的理解，通过一题多问、一题多解带动读者去发掘、去探索、去寻求更好的解决途径，从而达到提高分析问题和解决问题的能力。

4．编写本书的第四个出发点就是要以C语言为依托，贯穿算法设计、数据结构设计以及程序设计方法和软件工程思想的介绍，帮助读者在学习和掌握一门语言的同时养成良好的程序设计习惯。

全书章节与内容安排由苏小红和王宇颖统编制定，第3，5章及附录由苏小红编写，第1，6，7，9章由陈惠鹏编写，第2，4，8章由温东新编写。多媒体教学课件由李秀坤制作，另外发行。在书稿的录入与校对中，刘松波、陶海军、秦兵、刘秉权、李希然、李东也做了大量工作。

在本书写作过程中，王宇颖教授在百忙之中审阅了全部初稿，院系领导徐晓飞教授、廖明宏教授也给予了大力支持，王开铸、周明德、王庆北、刘开昌等其他许多教师对本书提出了许多宝贵意见，在此，一并表示衷心感谢。

因编者水平有限，书中错误在所难免，恳请批评指正，并多多提出您的宝贵意见。

作者

2001年12月

于哈尔滨工业大学

目　录

第 1 章　C 语言程序设计基础知识 ..... (1)
1.1　引言 ..... (1)
1.1.1　计算机语言与人类语言 ..... (1)
1.1.2　程序语言的简史 ..... (2)
1.2　C 语言的简介 ..... (3)
1.2.1　C 语言的发展历史 ..... (3)
1.2.2　C 语言的特点 ..... (4)
1.3　第一个 C 语言程序 ..... (5)
1.4　计算机程序编制的几个步骤 ..... (6)
1.4.1　用自然语言写文章的步骤 ..... (6)
1.4.2　用计算机语言编制程序 ..... (7)
1.4.3　一个编程实例 ..... (9)
1.5　C 语言常用符号 ..... (10)
1.6　计算机的构成对 C 语言的影响 ..... (11)
1.6.1　冯·诺依曼体系结构 ..... (12)
1.6.2　计算机存储模型 ..... (12)
习题一 ..... (13)
第 2 章　简单的 C 程序设计 ..... (14)
2.1　各种进位制的转换 ..... (14)
2.1.1　常用进位制 ..... (14)
2.1.2　进位制间的转换 ..... (15)
2.2　基本数据类型 ..... (16)
2.2.1　整型数据（Integer） ..... (17)
2.2.2　实型数据(Float) ..... (20)
2.2.3　字符型数据 (Character) ..... (22)
2.2.4　符号常量 ..... (24)
2.3　常用运算符及表达式 ..... (25)
2.3.1　算术运算符和算术表达式 ..... (25)
2.3.2　赋值运算符和赋值表达式 ..... (27)
2.3.3　增 1 和减 1 运算符 ..... (28)
2.3.4　位式运算 ..... (29)
2.3.5　逗号运算符和逗号表达式 ..... (31)
2.4　表达式语句 ..... (31)
2.5　基本输入输出操作的实现 ..... (32)
2.5.1　字符输入输出操作的实现 ..... (32)

2.5.2 有格式输入输出操作的实现 (33)
2.6 输入输出操作中常见的错误分析 (40)
2.7 程序举例 (41)
2.8 上机实验内容 (42)
实验一 写出运行结果程序练习 (42)
实验二 按打印结果要求编写程序练习 (43)
实验三 简单编程练习 (43)
习题二 (43)
第3章 程序的控制结构与结构化程序设计方法 (46)
3.1 算法与算法的表示方法 (46)
3.1.1 算法的概念 (46)
3.1.2 算法的表示方法 (47)
3.2 顺序结构程序设计 (49)
3.2.1 顺序结构的流程图表示 (49)
3.2.2 顺序结构应用举例 (50)
3.3 选择结构程序设计 (54)
3.3.1 选择结构的应用场合 (54)
3.3.2 关系运算符和关系表达式 (54)
3.3.3 逻辑运算符和逻辑表达式 (56)
3.3.4 选择结构的流程图表示 (57)
3.3.5 条件语句 (57)
3.3.6 开关语句 (65)
3.4 循环结构 (69)
3.4.1 循环结构的应用场合 (69)
3.4.2 循环结构的流程图表示 (69)
3.4.3 循环语句 (70)
3.4.4 单重循环问题应用举例 (72)
3.4.5 嵌套循环及其应用举例 (84)
3.4.6 转移控制语句 (88)
*3.5 结构化程序设计方法简介 (94)
*3.6 自顶向下、逐步求精的程序设计方法 (95)
3.6.1 什么是逐步求精方法？ (95)
3.6.2 什么是自顶向下的程序设计方法？ (95)
3.6.3 逐步求精实现技术 (96)
3.6.4 应用举例 (97)
*3.7 简单的程序调试方法 (98)
3.7.1 程序中常见的出错原因 (99)
3.7.2 Turbo C集成环境下的跟踪调试方法 (100)
3.7.3 其他排错方法 (101)
3.8 上机实验内容 (102)

实验一　身高预测 …… (102)
实验二　计算到期存款本息之和 …… (102)
实验三　猜数游戏 …… (102)
实验四　存款预算 …… (102)
实验五　抓交通肇事犯 …… (103)
实验六　求解不等式 …… (104)
实验七　计算礼炮声响次数 …… (104)
*实验八　寻找最佳存款方案 …… (105)
习题三 …… (107)
第4章　函数与模块化设计方法 …… (114)
4.1　函数 …… (114)
4.1.1　函数的分类 …… (114)
4.1.2　函数的定义 …… (115)
4.1.3　函数的返回值 …… (116)
4.1.4　函数的调用与参数传递 …… (116)
4.1.5　函数原型的说明 …… (119)
4.2　宏定义 …… (119)
4.3　变量的作用域和存储类 …… (121)
4.3.1　变量的作用域 …… (121)
4.3.2　变量的存储类 …… (125)
*4.4　模块化程序设计方法 …… (127)
4.4.1　模块化程序设计方法的指导思想 …… (127)
4.4.2　模块分解的原则 …… (128)
*4.5　应用设计实例 …… (129)
*4.6　多文件方式组织的程序 …… (134)
4.7　上机实验内容 …… (136)
实验一　小学生算术题 I …… (136)
实验二　小学生算术题II …… (137)
实验三　计算最大公约数和最小公倍数 …… (138)
实验四　分析程序的运行结果 …… (139)
习题四 …… (140)
第5章　数组与指针 …… (143)
5.1　数组 (Arrays) …… (143)
5.1.1　数组类型的应用场合 …… (143)
5.1.2　定义、引用和初始化 …… (143)
5.1.3　一维数组应用举例 …… (147)
5.1.4　一维数组名作函数参数 …… (152)
5.1.5　二维数组及二维数组名作函数参数应用举例 …… (161)
5.1.6　字符数组（Character Arrays） …… (166)
5.2　指针 (Pointers) …… (171)

5.2.1 指针的概念 ..... (171)
5.2.2 为什么引入指针的概念 ..... (173)
5.2.3 变量的指针与变量的指针作为函数参数 ..... (175)
5.2.4 字符指针与字符指针作为函数参数 ..... (181)
5.3 指针和数组间的联系 ..... (185)
5.3.1 一维数组的地址和指针 ..... (185)
5.3.2 二维数组的地址和指针 ..... (191)
5.4 指针数组(Pointer Arrays) ..... (196)
5.5 指向指针的指针 (Pointers to Pointers) ..... (200)
5.6 带参数的main函数和命令行参数 ..... (201)
*5.7 动态数组的实现 ..... (203)
5.7.1 动态内存分配函数 ..... (203)
5.7.2 一维动态数组的实现 ..... (205)
5.7.3 二维动态数组的实现 ..... (206)
*5.8 关于面向过程的程序设计 ..... (207)
*5.9 关于防御性程序设计 ..... (207)
*5.10 关于程序质量的重要性 ..... (208)
5.11 上机实验内容 ..... (209)
实验一 产值翻番 ..... (209)
实验二 餐饮服务调查打分 ..... (210)
实验三 简单的口令检查程序 ..... (211)
实验四 学生成绩统计 ..... (212)
实验五 排名次 ..... (214)
*实验六 大奖赛现场统分 ..... (217)
习题五 ..... (220)
第6章 结构体与共用体 ..... (226)
6.1 问题的提出 ..... (226)
6.2 结构体类型与结构体变量 ..... (228)
6.2.1 结构体的声明 (Declaration of Structure) ..... (228)
6.2.2 定义结构体变量 ..... (229)
6.2.3 定义指向结构体的指针 ..... (232)
6.2.4 结构体变量的引用 ..... (232)
6.2.5 结构体变量的初始化 ..... (234)
6.3 结构体数组 ..... (236)
6.3.1 结构体数组的定义 ..... (236)
6.3.2 结构体数组的应用实例 ..... (237)
6.3.3 结构体数组与指针 ..... (241)
6.4 结构体与函数 ..... (242)
6.5 动态数据结构 ..... (245)
6.5.1 问题的提出 ..... (245)

6.5.2 链表的定义 ..... (246)
6.5.3 链表的特点及操作原理 ..... (247)
6.5.4 动态链表的建立 ..... (248)
6.5.5 链表的删除操作 ..... (250)
6.5.6 链表的插入操作 ..... (251)
6.6 typedef 的使用 ..... (253)
6.7 共用体（Union） ..... (254)
*6.8 位段（Bit Field） ..... (256)
6.9 枚举常量（Enumeration） ..... (259)
6.10 上机实验内容 ..... (260)
实验一 模拟数字式时钟设计 ..... (260)
实验二 学生成绩管理与统计 ..... (260)
实验三 动态建立学生成绩管理程序 ..... (261)
习题六 ..... (261)
第7章 关于函数应用的高级话题 ..... (265)
7.1 递归(Recursive Call) ..... (265)
7.1.1 递归问题的提出 ..... (265)
7.1.2 递归函数 ..... (266)
7.2 返回指针值的函数（Function Return Pointer） ..... (270)
7.3 函数指针（Pointers to Function） ..... (271)
*7.4 一个综合应用的实例 ..... (275)
7.5 上机实验内容 ..... (283)
实验一 Fibonacci 数列 ..... (283)
实验二 求游戏人员的年纪 ..... (284)
*实验三 字母排列组合游戏 ..... (284)
*实验四 函数指针编程练习 ..... (286)
习题七 ..... (286)
第8章 文件操作 ..... (288)
8.1 C文件概述 ..... (288)
8.2 文件指针 ..... (289)
8.3 文件的打开和关闭 ..... (289)
8.4 标准 I/O 及其重定向 ..... (291)
8.5 文件的读写 ..... (292)
8.5.1 按字符读写文件 ..... (292)
8.5.2 按数据块读写文件 ..... (293)
8.5.3 按格式读写文件 ..... (295)
*8.6 文件的定位 ..... (296)
*8.7 非缓冲文件系统 ..... (298)
8.8 上机实验内容 ..... (300)
实验一 文件内容追加 ..... (300)

实验二　模拟 DOS 下的 COPY 命令 ..... (300)
实验三　字符文件读写练习 ..... (301)
实验四　格式文件读写练习 ..... (301)
实验五　通信录存取 ..... (301)
习题八 ..... (302)
第 9 章　图形和声音的制作 ..... (303)
9.1　图形模式初始化 ..... (303)
9.2　一个图形程序实例 ..... (305)
9.3　一个声音程序实例 ..... (315)
9.4　上机实验内容 ..... (317)
实验一　设计一个菜单操作界面 ..... (317)
实验二　设计一个简易的键盘电子琴 ..... (317)
习题九 ..... (318)
附录 ..... (319)
附录 A　C 语言的关键字 ..... (319)
附录 B　运算符的优先级与结合性 ..... (319)
附录 C　常用字符与 ASCII 码对照表 ..... (320)
附录 D　常用的 ANSI C 库函数 ..... (321)
附录 E　常用的 Turbo C 屏幕窗口和图形函数 ..... (325)
附录 F　Turbo C 集成环境简介 ..... (330)
附录 G　Visual C++ 集成环境下运行标准 C 程序的方法 ..... (334)
参考文献 ..... (337)

第 1 章　C 语言程序设计基础知识

学 习 目 标

（1）了解程序语言的发展简史和计算机语言的分类。
（2）了解 C 语言的发展简史及 C 语言的特点。
（3）了解计算机程序编制的步骤。
（4）了解 C 语言程序的编辑、编译、链接和调试的过程。
（5）了解 C 语言的常用符号和计算机内存模型。

难 点 内 容

（1）C 语言编制的全过程。
（2）计算机内存模型。

1.1　引言

1.1.1　计算机语言与人类语言

在长期的历史发展过程中，人类为了交流思想、表达感情、交换信息，逐步发明了语言。在不同的地理环境、历史条件下生成了不同的语言形式，诸如，汉语、英语、法语、俄语、日语……通常我们称这种语言为自然语言（Natural Language）。为了某种专门需要，人类又发明了一些新的交流工具，也称之为语言，例如旗语、哑语，我们称之为人工语言（Artificial Language）。计算机出现以后，人类为了能够更好地与计算机进行通信，发明了专门与计算机打交道的交流工具，称之为程序设计语言（Programming Language）。

无论是何种专门用途，语言只有一个功能：用于交流。交流不能是单方面的，必然涉及双方面、多方面，为了便于理解，双方面或多方面必须遵循一种约定，而且这种约定在交流的过程中不应该发生变化。所以语言在几千年的发展中，逐步形成一种大家都遵守的约定，这种约定分四个层次，可以总结为：字（Word）、词或词组（Phrase）、句子或段落、篇章，其依托关系可以表示为图 1-1。最终表达人类思想的是篇章（文章），而句子、段落、词、词组、字都是为形成篇章而服务的，而语言最基本的单元是字。

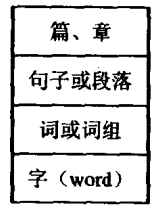

图 1-1　语言的层次关系图

1946 年第一台计算机(ENIAC)出现后，人类一直在想尽各种方法力图与计算机进行交流。人类很自然地想到运用人类最熟悉的交流工具语言与计算机进行交流，而人类的语言有很多缺欠，最主要的不足是人类的语言具有歧义。所谓的歧义，是说一段相同的文字在不同的场合、不同的环境、不同的语气表达方式下会有不同的理解和解释，当然人类可以根据周围环境判断某段文字的具体含义，而计算机并不具有这种技能，所以人类设计出词汇量少、语法

简单、意义明确的语言作为人类与计算机通信的载体，这样的载体通常称为程序设计语言。显然，计算机语言应该有基本的单元；由基本单元组成的相当于自然语言中的词或词组的构造单元；由基本单元与构造单元按照某种约定组成一篇文章与计算机进行交流，而这种约定我们可以称之为语法，最后形成的文章我们称之为程序。这种结构表示为图 1-2。

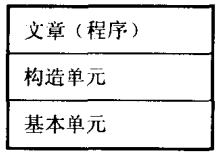

图 1-2　计算机语言的层次图

将图 1-1 与图 1-2 对应起来，可以看出自然语言与计算机语言的关系，见图 1-3。

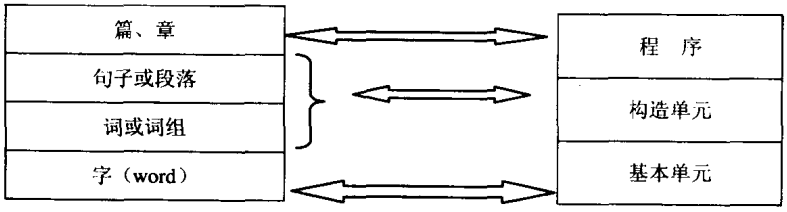

图 1-3　自然语言与计算机语言层次

1.1.2　程序语言的简史

随着第一台计算机 ENIAC 的诞生，计算机语言随即产生，最初的计算机语言为机器语言（Machine Language），也就是二进制语言，计算机可以直接执行。20 世纪 50 年代的人还记得那种纸带式的程序输入方式，现在的计算机工作人员只能在计算机博物馆里才能见到带孔的纸带。汇编语言的出现使人们可以摆脱直接面对二进制的烦恼，可以用助记符的形式操作计算机，但是仍然摆脱不了汇编语言与计算机紧密相关的困苦，一直到 20 世纪 50 年代，符号化概念开始出现。到了 1955 年巴克斯领导的小组针对 IBM704 硬件开发出了 FORTRAN（源自 FORmula TRANslator），获得了巨大的成功，很快 FORTRAN 成为了编程的主流。而在欧洲，为了对抗 IBM 对计算机的统治，1960 年在彼得·劳尔的领导下研究成功了 ALGOL 语言，很快在欧洲获得了成功。从 1946 到 1956 年 10 年的时间，计算机语言从最初的基于二进制的语言发展到可以编译的语言，产生了第一次飞跃；20 世纪 70 年代出现的结构化程序设计语言 Ada，C，Prolog，Pascal，SmallTalk 的产生，出现了第二次飞跃，而在第二次飞跃中，C 语言的出现在计算机界留下了非常深的印记；如果说 20 世纪 80 年代以前的语言是面向过程的编程语言，到了 80 年代后，计算机语言跨入了面向对象的编程，出现了 C++，ML，Perl，Postscript 等与自然语言更接近的语言，出现了第三次飞跃；20 世纪 90 年代 Java 的出现标志着单机语言向网络语言的跨越，出现了第四次飞跃。可以看得到，计算机语言几乎是每隔十年就出现一次飞跃，进入 21 世纪计算机语言又会出现什么样的变化呢？整个世界在拭目以待。

多种计算机语言出现在不同的年代，同时也有其功能上的不同，主要侧重于以下几个方向。

1. 基于数学计算的语言

早期的计算机主要用于军事用途，诸如导弹弹道计算等（ENIAC），数字计算成为早期计算机的主要应用。在计算机发明的初期，计算机语言以二进制为主，二进制是计算机的主要基础，是计算机惟一能识别和执行的语言，由二进制组成的语言通常被称为机器语言，任

何其他后出现的语言都必须最终翻译成二进制才能被计算机执行。诸如 FORTRAN 是为 IBM704 量身定做的，主要面向数学计算。正因为 FORTRAN 强大的科学计算能力，限制了他在其他方面的发展。

2. 商业语言

1960 年在美国国防部的主持下快速开发出了一种用于开发商业应用的程序的语言——COBOL 语言，1968 年实现了标准化。

3. 人工智能语言

20 世纪 50 年代人们对人工智能产生了浓厚的兴趣，此时 Rand 公司发明的 IPL-V 语言引起了人们的普遍关注，但由于其低水平的设计而导致用途有限。麻省理工学院的 John McCarthy 为 IBM 704 设计了 LISP 语言，很快成为了行业标准。LISP 是基于表处理的函数语言，而 Prologe 则是面向特殊用途的语言，它的控制结构及实现策略均基于数学逻辑中的概念。

4. 系统语言

虽然其他语言发展很快，但是在早期计算机资源比较匮乏的条件下，系统领域一直用汇编语言来开发，虽然很多先驱也努力开发出一些系统程序设计语言（如 CPL 和 BCPL），但一直未能得到广泛的应用。一直到 C 语言出现，并且由 C 语言写成的 UNIX 系统被计算机界广泛接纳后，一切才发生了变化，高级语言才进入了系统语言行列。

1.2 C 语言的简介

1.2.1 C 语言的发展历史

20 世纪 60 年代末期，AT&T 贝尔实验室取消了开发 Multics 操作系统计划，然而，开发一个有用的、有别于 DOS 的多用户操作系统仍然是 Ken Thompson 的理想，他开始设计开发 UNIX 系统。选用一种合适的开发语言开发 UNIX 是 Ken Thompson 所面临的首要问题，当时 AT&T 的 Multics 操作系统是用当时比较流行的 PL/I 语言开发的，但是 PL/I 语言编程十分不方便。具有丰富编程经验的 Ken Thompson 决定首先发明一种语言作为开发 UNIX 的工具语言（为了某种目的，设计一种合适的工具是人类的一种聪明方式，计算机编程人员尤其需要具有这样一种思想），他根据由 Martin Richards 开发的 BCPL（一种低层次且缺乏运行支持的系统语言）设计了 B 语言，BCPL 和 B 语言是一种无类型（typeless）语言。

1970 年，UNIX 项目组获得了 DEC 公司的一台 PDP-11 计算机，UNIX 项目组的成员逐渐发现 B 语言的劣势，在 Dennis Ritchie 的领导下，UNIX 项目组的成员在 B 语言的基础上加入了类型（Datatype）和结构（Structure），形成了 C 语言的原始框架。

1978 年一本名为 C Programming Language 的书的出版，标志着 C 语言的诞生，从此计算机编程出现了一场革命。20 世纪 80 年代，随着商业版 UNIX 操作系统的广泛应用，C 语言开始流行起来。

1982 年随着 C 语言普及性的提高，ANSI（American National Standards Institute）专门成立了一个委员会，目的是制定“一个没有歧义、与机器无关的 C 语言”，仍然保留原 C 语言

的主体。该工作于 1989 年完成[ANSI 1989]，该标准 1990 年被接受并作为国际标准（ISO/IEC9899），这就是我们今天常说的 ANSI C。我们这本书也主要介绍 ANSI C。

一开始，C 语言是为编 UNIX 而开发的一种工具语言。随着 C 语言标准的制定，C 成为了一种通用性语言，UNIX 系统上的很多应用程序都是用 C 语言编制的。

C 语言发展到今天已经出现了分支，在计算机的主流 PC 机上，由于 Windows 的广泛使用，基于 Windows 编程的 Microsoft Visual Studio 已经成为主流，Visual C++ 是程序员选择的主要编程工具。C 语言作为最初的有效的面向过程的编程工具，逐步被面向对象的编程工具 C++所替代，而在嵌入式系统的环境下，C 语言依然发挥着其不可替代的作用。例如，在掌上电脑 Palm 系统，其支持的开发语言即为 C 语言；在单片机系统中，C 语言逐步取代原来的汇编级的编程，转而支持 C 语言编程，这里包括用量很大的 Intel 8X1 系列单片机、Motorola 公司的 6800 系列单片机、Philips 公司的单片机系列等。所以，C 语言已经成为昨日黄花的说法是完全错误的。

1.2.2 C 语言的特点

C 语言有以下特点：

（1）语言简练、紧凑，使用方便、灵活。C 语言常用符号（Tokens）包括六类：标识符（Identifiers）、关键字（Keywords）、常量（Constants）、字符串（String littorals）、操作符（Operators）、分割符（Separator）。C 语言一共只有 32 个关键字，9 种控制语句。程序书写形式自由，标识符的定义非常灵活，支持大小写敏感。C 语言的这些特点使得编程人员的个性得以发挥，程序编制变得不再枯燥乏味。

（2）运算符丰富。C 语言共有 34 种运算符，将括号、赋值、类型转换都以运算符形式出现，从而使得 C 语言可以非常方便、灵活地表示各种算法，既弥补了其他语言在算法表示上的缺欠，同时也使程序更易读、简捷。C 语言从其一产生，即进行了严格的数据类型定义，这些数据类型可以分为基本数据类型（Basic Datatype）和构造数据类型（Construct Datatype）。读者会从不断的学习当中体会到丰富的数据类型的优势。

（3）C 语言是结构化程序设计语言。支持三种结构化程序设计机构：顺序结构，选择结构，循环结构。结构化编程方式是从 20 世纪 60 年代末 70 年代初兴起的一种编程方法，该方法的使用使得程序的可读性增强，易于维护。

（4）语法限制不十分严格。这是一个非常好的特点，使得程序员在书写程序的时候按照自己喜欢的方式和格局进行设计，使个性得以充分发挥，编程将不再是一个非常死板的工作。

（5）程序可移植性好。对许多程序工程而言，一项重要的标准就是能否将编制和运行良好的程序从它运行的计算机系统很方便地移植到其他的计算机系统，诸如在 DOS 下编制的一个程序，经过较少的改动即可以在 UNIX 系统下运行；在 IBM 及其兼容机上编制的程序经过较少改动可以在 SUN 计算机系统上运行。所以 C 语言在标准制定时，就以其独立于特定机器的特征而成为一个容易移植的语言。

（6）程序的易验证性。某种语言编写的程序是否可靠常常是软件工程中的一个核心问题。有许多方法可以用来判断一个程序是否能够正常、有效地运行。一个程序可以通过形式化判断方法来进行正确性判断，可以通过读并且浏览检查程序语句来非形式化地证明其正确性，也可以按规则说明实验数据输入并检查输出结果来测试，等等。C 语言的语义和句法结构的简单为简化程序验证提供了重要的前提。