



普通高等教育“十一五”国家级规划教材

丛书主编 谭浩强

高等院校计算机应用技术规划教材

基础教材系列

大学C语言 实用教程

潘旭华 主编

潘旭华 陈刚 姜书浩 赵玉刚 编著

高福成 主审

根据“中国高等院校计算机基础教育课程体系”组织编写

清华大学出版社





普通高等教育“十一五”国家级规划教材

丛书主编 谭浩强

高等院校计算机应用技术规划教材

基础教材系列

大学C语言实用教程

潘旭华 主编

潘旭华 陈刚 姜书浩 赵玉刚 编著

高福成 主审

清华大学出版社
北京

内 容 简 介

C语言是一门广泛应用的计算机语言,本书以程序设计为主线,以程序设计的需要带动语言知识的学习,系统介绍了C语言及其程序设计技术。全书共10章,包括简单的C程序设计、基本数据类型、数据运算、程序流程控制、数组和字符串、指针、函数、复合数据类型、文件、编译预处理和分割编译,通过丰富的C程序设计实例,详尽介绍了相应的算法知识,各章编排了一定量的练习题,以帮助读者在初步掌握语言的基础上,着重培养程序设计的能力。

本书内容充实,体系合理,概念清晰,讲解详尽,例题丰富,是学习C程序设计的理想教材,可作为高等学校本科和研究生教学用书。本书配有辅助教材《大学C语言实用教程实验指导与习题》(ISBN 978-7-302-24596-4)供读者学习使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

大学C语言实用教程/潘旭华主编. —北京:清华大学出版社,2011.3

(高等院校计算机应用技术规划教材——基础教材系列)

ISBN 978-7-302-24597-1

I. ①大… II. ①潘… III. C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第012440号

责任编辑:汪汉友

责任校对:焦丽丽

责任印制:孟凡玉

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦A座

邮 编:100084

邮 购:010-62786544

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260

印 张:23.25

字 数:537千字

版 次:2011年3月第1版

印 次:2011年3月第1次印刷

印 数:1~4000

定 价:36.00元

产品编号:040447-01

序

《高等院校计算机应用技术规划教材》

进入 21 世纪,计算机成为人类常用的现代工具,每一个有文化的人都应当了解计算机,学会使用计算机来处理各种的事务。

学习计算机知识有两种不同的方法:一种是侧重理论知识的学习,从原理入手,注重理论和概念;另一种是侧重于应用的学习,从实际入手,注重掌握其应用的方法和技能。不同的人应根据其具体情况选择不同的学习方法。对多数人来说,计算机是作为一种工具来使用的,应当以应用为目的、以应用为出发点。对于应用型人才来说,显然应当采用后一种学习方法,根据当前和今后的需要,选择学习的内容,围绕应用进行学习。

学习计算机应用知识,并不排斥学习必要的基础理论知识,要处理好这二者的关系。在学习过程中,有两种不同的学习模式:一种是金字塔模型,亦称为建筑模型,强调基础宽厚,先系统学习理论知识,打好基础以后再联系实际应用;另一种是生物模型,植物并不是先长好树根再长树干,长好树干才长树冠,而是树根、树干和树冠同步生长的。对计算机应用型人才教育来说,应该采用生物模型,随着应用的发展,不断学习和扩展有关的理论知识,而不是孤立地、无目的地学习理论知识。

传统的理论课程采用以下的三部曲:提出概念—解释概念—举例说明,这适合前面第一种侧重知识的学习方法。对于侧重应用的学习者,我们提倡新的三部曲:提出问题—解决问题—归纳分析。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。这种教学方法更适合于应用型人才。

检查学习好坏的标准,不是“知道不知道”,而是“会用不会用”,学习的目的主要在于应用。因此希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而可以指定学生通过上机掌握这些内容。这样做可以培养学生的自学能力,启发学生的求知欲望。

全国高等院校计算机基础教育研究会历来倡导计算机基础教育必须坚持面向应用的正确方向,要求构建以应用为中心的课程体系,大力推广新的教学三部曲,这是十分重要的指导思想,这些思想在《中国高等院校计算机基础课程》中作了充分的说明。本丛书完全符合并积极贯彻全国高等院校计算机基础教育研究会的指导思想,按照《中国高等院校计算机基础教育课程体系》组织编写。

这套《高等院校计算机应用技术规划教材》是根据广大应用型本科和高职高专院校的迫切需要而精心组织的,其中包括4个系列:

(1) 基础教材系列。该系列主要涵盖了计算机公共基础课程的教材。

(2) 应用型教材系列。适合作为培养应用型人才的本科院校和基础较好、要求较高的高职高专学校的主干教材。

(3) 实用技术教材系列。针对应用型院校和高职高专院校所需掌握的技能技术编写的教材。

(4) 实训教材系列。应用型本科院校和高职高专院校都可以选用这类实训教材。其特点是侧重实践环节,通过实践(而不是通过理论讲授)去获取知识,掌握应用。这是教学改革的一个重要方面。

本套教材是从1999年开始出版的,根据教学的需要和读者的意见,几年来多次修改完善,选题不断扩展,内容日益丰富,先后出版了60多种教材和参考书,范围包括计算机专业和非计算机专业的教材和参考书;必修课教材、选修课教材和自学参考的教材。不同专业可以从中选择所需要的部分。

为了保证教材的质量,我们遴选了有丰富教学经验的高校优秀教师分别作为本丛书各教材的作者,这些老师长期从事计算机的教学工作,对应用型的教学特点有较多的研究和实践经验。由于指导思想明确、作者水平较高,教材针对性强,质量较高,本丛书问世7年来,愈来愈得到各校师生的欢迎和好评,至今已发行了240多万册,是国内应用型高校的主流教材之一。2006年被教育部评为普通高等教育“十一五”国家级规划教材,向全国推荐。

由于我国的计算机应用技术教育正在蓬勃发展,许多问题有待深入讨论,新的经验也会层出不穷,我们会根据需要不断丰富本丛书的内容,扩充丛书的选题,以满足各校教学的需要。

本丛书肯定会有不足之处,请专家和读者不吝指正。

全国高等院校计算机基础教育研究会会长
《高等院校计算机应用技术规划教材》主编

谭浩强

2008年5月1日于北京清华园

前言

C语言是广泛应用的计算机语言,使用C语言进行程序设计是计算机程序员的基本功。当今流行的面向对象语言C++、Visual C++、C#以及Java语言都源于C。

在计算机日益普及、计算机技术日新月异、新型计算机软件层出不穷的今天,程序设计仍然是每一位当代大学生应当掌握的基本技能,是计算机素质教育的重要环节之一。那些希望用计算机解决本专业领域实际问题的有识之士,更需要加强程序设计的学习与训练。

学习计算机语言的唯一目的是应用,而应用要通过程序设计来体现。程序设计发展到今天,已经由技艺提高到了科学的层面,有了自己的一套基本原理和方法。进行程序设计,需要很强的逻辑思维能力,是一种极富创造性的智力劳动。对初学者而言,最使他们望而生畏的,往往也正是这一点。可以这样认为,语言是一种技能,程序设计是一门科学。因此,任何计算机语言及其程序设计的基本特点就是理论性和实践性并重,教学上应强调科学训练与技能培养并存。基于这一认识,本书以应用为目的,以提高程序设计能力为目标,以程序设计方法学为依据,系统介绍了C语言及其程序设计技术,把程序设计作为科学来讲授,把语言作为技能来培养,让读者在大量的程序设计实践中自然而然地熟悉和掌握。

根据C语言自身的特点,本书从内容上做了如下安排:第1章是简单的C程序设计,尽管简单,但是非常重要;第2章、第5章和第8章从易到难介绍了C语言的各种数据结构(基本数据类型、数组和字符串、复合数据类型);第3章专门介绍C语言的数据运算;第4章集中介绍C语言的控制结构和程序设计的基本思维方法,体现结构化程序设计的特点;第6章介绍指针,这是C语言的精髓和特色,也是学习的难点;第7章和第9章介绍函数和文件,这是模块化程序设计的需要;第10章介绍了C语言环境中的特殊而较为深入的问题(如编译预处理,分割编译等),这些问题基本不影响本书对初学者的要求,但对较高层次的读者来说又是必要的资料,可以作为自学之用;附录部分给出了Visual C++ 6.0操作指导、C语言运算符集、ASCII代码表、常用库函数、习题解答以及综合应用示例,便于读者的查阅。书中所有例题源程序均通过Visual C++ 6.0编译系统调试,为方便读者上机实践以及课后练习与检测,本书配有辅助教材《大学C语言实用教程实验指导与习题》。

本书的写作大纲由高福成教授和潘旭华教授共同制定,陈刚编写第1章、第2章、第3章和附录A~附录D及附录E部分内容;潘旭华编写第4章、第5章、第6章、第7章、附录F及附录E部分内容;姜书浩编写第8章及附录E部分内容;赵玉刚编写第9章、第10章及附录E部分内容。全书由潘旭华教授担任主编并统稿,高福成教授精心审阅了全部书稿。本书在编写和出版过程中,得到作者所在学校的大力支持,清华大学出版社的编辑为此付出了大量的辛勤劳动,在此一并表示感谢。

本书配套的电子教学资源(教学大纲、实验大纲、授课计划、电子教案等),读者可在清华大学出版社网站(<http://www.tup.com.cn>)本书相应的页面中下载。

由于作者学识水平所限,书中难免疏漏和错误,恳请读者不吝指正。

编者
2010年11月

编辑委员会

《高等院校计算机应用技术规划教材》

主 任 谭浩强
副 主 任 焦金生 陈 明 丁桂芝

委 员 (按姓氏笔画排序)

王智广	孔令德	刘 星	刘荫铭
安志远	安淑芝	孙 慧	李文英
李叶紫	李 琳	李雁翎	宋 红
陈 强	邵丽萍	尚晓航	张 玲
侯冬梅	郝 玲	赵丰年	秦建中
莫治雄	袁 玫	訾秀玲	薛淑斌
谢树煜	谢 琛		








目录

第 1 章 简单的 C 程序设计	1
1.1 程序和程序设计的概念	1
1.2 程序设计技术的 4 个层次	2
1.3 C 语言的产生和发展	4
1.4 C 程序的结构特点和书写风格	6
1.4.1 一个简单的 C 程序	6
1.4.2 C 程序的结构特点	6
1.4.3 C 程序的书写风格	9
1.5 简单的 C 程序设计	9
1.5.1 赋值语句的简单使用	9
1.5.2 格式输入输出函数的简单使用	10
1.5.3 库函数和标题文件	10
1.5.4 简单程序设计举例	11
1.6 C 程序的开发过程	12
1.7 Visual C++ 6.0 集成开发环境的使用	13
1.7.1 启动 Visual C++ 6.0 集成开发环境	13
1.7.2 在 Visual C++ 6.0 环境中开发 C 程序	13
习题 1	19
第 2 章 基本数据类型、常量和变量	21
2.1 基本数据类型	21
2.2 常量及其类型	23
2.3 变量的定义及初始化	26
2.4 符号常数	27

2.5	不同类型数据的输入输出	29
2.5.1	格式输出函数 printf()	29
2.5.2	格式输入函数 scanf()	34
2.5.3	单字符输入输出函数 getchar()和 putchar()	37
2.5.4	单字符输入函数 getche()和 getch()	39
习题 2	40
第 3 章	数据运算	44
3.1	算术运算	44
3.1.1	算术运算符	44
3.1.2	算术表达式及其求值规则	45
3.2	赋值运算	49
3.3	逗号运算(顺序运算)	52
3.4	关系运算和逻辑运算	53
3.5	测试数据长度运算符	57
3.6	常用数学函数	58
习题 3	60
第 4 章	程序流程控制	65
4.1	结构化程序设计的三种基本结构	65
4.2	选择结构程序设计	66
4.2.1	用 if...else 语句构成二分支选择结构	66
4.2.2	用 if...else if...else 语句构成多分支选择 结构	71
4.2.3	用条件表达式实现的选择结构	73
4.2.4	用 switch 语句构成多分支选择结构	75
4.2.5	用 switch 和 break 构成多分支选择结构	77
4.3	循环结构程序设计	78
4.3.1	for 循环	78
4.3.2	while 循环	82
4.3.3	do...while 循环	84
4.3.4	循环的嵌套	86
4.4	转移控制语句	88
4.4.1	break 语句	88

4.4.2	continue 语句在循环结构中的作用	90
4.4.3	goto 语句和标号	91
习题 4	92
第 5 章	数组和字符串	102
5.1	数组的概念	102
5.2	数组的定义和初始化	103
5.2.1	数组的定义	103
5.2.2	数组的存储	104
5.2.3	数组的初始化	105
5.2.4	通过初始化定义隐含尺寸数组	107
5.3	数组的基本操作	108
5.4	数组的应用	113
5.4.1	数值数组的应用	113
5.4.2	字符串处理函数和字符串的应用	120
习题 5	128
第 6 章	指针	133
6.1	地址、指针和指针变量	133
6.1.1	地址和地址的运算	133
6.1.2	指针和指针变量	136
6.2	指针的运算	139
6.3	用指针访问一维数组	141
6.4	用指针访问二维数组	143
6.5	用指针处理字符串	149
6.6	二级指针和多级指针	154
习题 6	156
第 7 章	函数	163
7.1	模块化程序设计的思想	163
7.2	C 函数的定义和调用	164
7.3	调用函数和被调用函数之间的数据传递	169
7.3.1	参数传递方式	169
7.3.2	函数返回值方式	178
7.3.3	全局变量方式	181

7.3.4	用函数指针调用函数和传递函数	181
7.4	存储类型对函数调用的影响	185
7.4.1	变量的存储类型	185
7.4.2	函数的存储类型对函数调用的影响	191
7.5	函数的递归调用	192
7.6	main()函数的参数和返回值	194
习题 7	197
第 8 章	复合数据类型	207
8.1	结构类型	207
8.1.1	结构类型的定义和存储模式	207
8.1.2	结构变量、结构数组和结构指针的定义和 初始化	208
8.1.3	访问结构变量和结构数组的成员	211
8.1.4	结构变量、结构数组和结构指针的赋值、输入和 输出	213
8.2	结构在函数间的传递	214
8.3	递归结构和内存动态分配的综合应用——链表的 操作	221
8.3.1	递归结构和链表的概念	221
8.3.2	内存的动态分配	222
8.3.3	动态链表的基本操作	223
8.4	联合类型	230
8.5	位运算和位段结构类型	235
8.5.1	位运算	235
8.5.2	位段结构类型	239
8.5.3	位段结构和位操作应用举例	242
8.6	枚举类型	246
8.7	类型定义	248
习题 8	250
第 9 章	文件	260
9.1	文件概述	260
9.2	文件的打开与关闭	262
9.3	文件的读写操作	264
9.3.1	文本文件读写函数	264

9.3.2	二进制文件读写函数	270
9.4	文件检测函数	272
9.5	文件的顺序存取和随机存取	274
	习题 9	283
	第 10 章 编译预处理和分割编译	290
10.1	编译预处理	290
10.1.1	宏定义	290
10.1.2	文件包含	295
10.1.3	条件编译	297
10.2	分割编译	299
	习题 10	303
	附录 A Visual C++ 6.0 操作指导	308
A.1	单文件 C 程序的开发	308
A.2	编辑已有的项目和程序	313
A.3	多文件 C 程序的开发	314
A.4	项目中文件的插入和删除	316
	附录 B C 语言运算符集	318
	附录 C ASCII 代码表	319
	附录 D C 常用库函数及其标题文件	320
D.1	I/O 函数(标题文件 stdio.h)	320
D.2	字符判别和转换函数(标题文件 ctype.h)	322
D.3	字符串函数(标题文件 string.h/mem.h)	322
D.4	数学函数(标题文件 math.h)	323
D.5	动态分配函数及其他(标题文件 stdlib.h)	324
	附录 E 习题参考答案	325
	附录 F C 程序应用开发案例	345

第1章

简单的 C 程序设计

C 语言是一种面向过程的通用程序设计语言,它以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作符集合、良好的程序可移植性和高效率的目标代码为特征。C 语言不仅具有高级语言的要素,还兼有低级语言的功能,因此既可用于编写系统程序,也可用于编写不同领域的应用程序。本章从程序设计的概念切入,介绍简单的 C 程序的编制及 Visual C++ 6.0 集成环境的使用方法。

1.1 程序和程序设计的概念

计算机是执行程序的工具。为解决某一实际问题而设计的指令序列就是程序。用计算机解决任何问题都离不开程序。

程序设计是把要求解的任务或设计好的算法转换为计算机程序的全过程。例如,让计算机求二次曲线 $y = ax^2 + bx + c$ 与 x 轴的交点。为了解决该问题,应遵循以下步骤。

(1) 根据所要解决的问题,确定要用计算机计算或加工处理的任务,如果是数值计算问题,应建立该问题的数学模型。本题的数学模型是求一元二次方程 $ax^2 + bx + c = 0$ 的实根。

(2) 确定让计算机完成任务的具体步骤,这就是算法。本题要先计算根的判别式,根据判别式可能大于 0、等于 0 或小于 0,分别对应两个不相等实根、两个相等实根或不存在实根的 3 种可能,也就是分别代表曲线与 x 轴可能有两个交点、一个交点或没有交点。

(3) 详细说明要输入的数据及要输出的结果。本题要输入的数据是方程的三个系数 a 、 b 、 c ,要输出的结果是方程的根 x_1 和 x_2 (即与 x 轴的交点)。

(4) 根据算法画出程序框图,也叫流程图。本题流程图如图 1.1 所示。

(5) 选择一种程序设计语言(例如 C 语言)编写程序,即把算法转换成程序。

(6) 上机调试程序,可以用 a 、 b 、 c 不同值的组合进行程序验证,找出并修改程序中的错误,直到程序正确为止。

(7) 整理文档,编写程序使用及维护说明书。

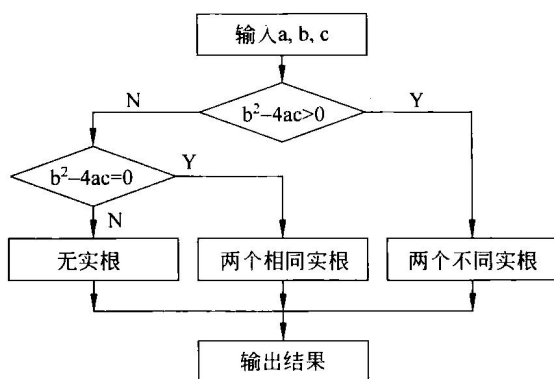


图 1.1 求解一元二次方程程序框图

1.2 程序设计技术的 4 个层次

程序设计与计算机同时诞生、相伴发展的,整个计算机产业就是在进行程序的设计与开发。因此,计算机人才培养与训练的主要任务就是让学生掌握程序设计及其相关理论的研究与开发能力。

程序设计技术的研究与培养是在不同层次上进行的,高水平的程序员必须在各个不同的层次上接受充分的训练,这些层次分别是算法、方法学、语言和环境与工具。

1. 计算机算法

计算机算法是用计算机解决问题的方法和步骤。计算机算法分为两大类:一类是数值算法,包括插值和拟合,线性和非线性方程的求解,常微分方程的求解,偏微分方程的求解等;另一类是非数值算法,诸如排序,字符识别,图形、图像及声音等多媒体信息的处理,运筹学方法等。算法理论除了要设计出正确的问题求解方法外,还要考虑算法在时间和空间这两个方面的复杂度,以使用最短的时间和最小空间代价解决一个给定的问题。例如,解决多元一次方程组问题,用克拉默法则(即行列式的方法)就不是一个好的算法,随着问题规模的扩大,其计算量将呈几何级数增加,而消元法则是较好的算法,在计算机中得到了广泛的应用。

算法是整个计算机科学的核心之一,一个高级程序员必须掌握有关算法设计与分析的知识,否则,他不会编出高水平高质量的程序,更不要说编写开拓性和创造性的程序了。

2. 程序设计方法学

程序设计方法学的研究是从解决 20 世纪 60 年代末期的“软件危机”开始的,其目标是在开发正确可靠软件的前提下,如何缩短软件开发周期,进而如何延长所设计的软件的使用周期,包括程序的可读性、可测试性、可维护性、可扩充性和可重用性。

结构化程序设计方法(structural programming, SP)的概念由 E. W. Dijkstra 在

20 世纪 60 年代末提出,其基本思想是采用“自顶向下,逐步求精”的程序设计方法和“单入口单出口”的控制结构。自顶向下、逐步求精的程序设计方法从问题本身开始,经过逐步细化,将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图;“单入口单出口”的思想认为一个复杂的程序,如果它仅是由顺序、选择和循环 3 种基本程序结构通过组合、嵌套构成,那么这个新构造的程序一定是一个单入口单出口的程序。据此就很容易编写出结构良好、易于调试的程序来。

结构化程序设计可以用下面的式子表示程序:

$$\text{程序} = \text{数据结构} + \text{算法} + \text{程序设计语言} + \text{语言环境}$$

面向对象程序设计方法(object oriented programming, OOP)既吸取了结构化程序设计的一切优点,又考虑了现实世界与面向对象解空间的映射关系,它所追求的目标是将现实世界的问题求解尽可能简单化。面向对象程序设计采用了数据抽象和信息隐藏技术,把数据的静态属性和对它的操作“封装”在一起,作为一个相互依存,不可分割的整体(类和对象)来处理;通过“继承”,可以扩展已存在的类,利用“多态”实现接口重用。面向对象程序设计方法的这 3 个基本特征,一方面提高了软件的可重用性、可扩展性、可维护性,增加了代码的灵活性,另一方面还符合人类的思维习惯,能够自然地表现现实世界的实体和问题,对软件开发过程具有重要的意义。

面向对象程序设计可以用下面的式子表示程序:

$$\text{程序} = \text{对象} + \text{对象} + \dots + \text{对象}$$

$$\text{对象} = \text{算法} + \text{数据结构} + \text{程序设计语言} + \text{语言环境}$$

3. 程序设计语言

程序设计语言,通常简称为编程语言,是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧,用来向计算机发出指令。程序设计语言是程序设计的基础,任何人学习程序设计都必须从学习一种语言开始。因此,在程序设计的 4 个层次中,语言是最基本最起码的要求。

在过去的几十年间,大量的程序设计语言被发明、被取代、被修改或组合在一起。尽管人们多次试图创造一种通用的程序设计语言,却没有一次尝试是成功的。之所以有那么多种不同的编程语言存在的原因是,编写程序的初衷其实也各不相同;新手与老手之间技术的差距非常大,而有许多语言对新手来说太难学;还有,不同程序之间的运行成本(runtime cost)各不相同。因此,实际应用的关键是选择一种最合适最有前途的语言。

计算机语言的种类千差万别,按语言级别有低级语言和高级语言之分;按用户要求有过程式语言和非过程式语言之分;按应用范围有通用语言和专用语言之分;按使用方式有交互式语言和非交互式语言之分;按成分性质,有顺序语言、并发语言和分布语言之分。

程序设计语言的基本成分不外 4 种。

- (1) 数据成分。用以描述程序中所涉及的数据。
- (2) 运算成分。用以描述程序中所包含的运算。

(3) 控制成分。用以表达程序中的控制结构。

(4) 传输成分。用以表达程序中数据的传输。

程序设计语言的发展趋势是模块化、简明性和形式化。模块化指的是不仅语言具有模块成分,程序由模块组成,而且语言本身的结构也是模块化的;简明性指的是涉及的基本概念不多,成分简单,结构清晰,易学易用;形式化指的是运用合适的形式体系,以描述语言的语法、语义、语用。

4. 程序设计环境与工具

程序设计环境是为程序员开发程序提供的环境,例如编辑器、连接器和调试器是现代编译系统为程序员提供的主要集成环境,此外,各种编译系统都提供了标准子程序,将专家已经编制好的诸如三角函数、对数函数、字符处理函数等常用程序,供程序员设计时使用。

程序设计工具也越来越多,包括设计分析工具、测试分析工具、测试实例生成、模块代码生成工具、文档生成工具、工程进度管理工具等,这些工具统称为 CASE(计算机辅助软件工程),现代程序设计工具越来越具有智能、代替和“强迫”程序员按照程序设计方法学和语法进行设计工作,它们可以在程序员水平不高的情况下,在一定程度上保证代码的质量。

在上述 4 个层次中,计算机算法是程序设计的核心,它需要深入的基础理论知识,在时间上相对稳定,在效果上往往在广泛的领域中起到突破作用;程序设计方法学是缩短程序设计周期、延长软件寿命、降低软件开发成本的关键,它的理论更新于计算机自身应用规模的发展同步,是一种难以抗拒的发展趋势,不是可以由人们自由选择的,相对于算法而言变化较快,但相对于各种语言及其版本而言则相对稳定;计算机语言是程序设计的基础,其变化快于算法和程序设计方法,而慢于程序设计环境与工具;处于 4 个层次最底层的程序设计环境与工具正向着多样化、智能化、集成化发展,越来越面向一般用户,易学好用,灵活方便,逐步成为程序设计的必要条件,但它在 4 个层次中最不稳定,几乎每年都在变化更新。

1.3 C 语言的产生和发展

C 语言的原型是 ALGOL 60(ALGOriThm Language)。ALGOL 60 是 1960 年由国际计算机委员会设计的一种面向过程的结构化程序设计语言,用它编写的程序具有可读性和可移植性好的特点。但是,它不能直接对硬件进行操作,不宜用来编写系统程序。系统程序主要用汇编语言编写,而汇编语言是面向硬件的,用它编写的程序可读性和可移植性都比较差。为此,人们开始考虑设计一种集高级语言和低级语言功能于一身的语言,以便用它来编写可读性和可移植性都比较好的系统程序。

1963 年,英国的剑桥大学和伦敦大学首先将 ALGOL 60 发展成 CPL(Combined Programming Language),该语言已比较接近于硬件,但规模较大,难以实用。

1967 年,剑桥大学的 Martin Richards 将 CPL 改制成 BCPL(Basic Combined