

高等学校教材

新概念C语言

李一波 张森悦 孙玉霞 王亚杰 编著



NEUPRESS
东北大学出版社

新 概 念 C 语 言

李一波 张森悦 孙玉霞 王亚杰 编著

东 北 大 学 出 版 社

· 沈 阳 ·

© 李一波 等 2004

图书在版编目 (CIP) 数据

新概念 C 语言 / 李一波等编著. —沈阳: 东北大学出版社, 2004. 1
ISBN 7-81054-850-6

I. 新… II. 李… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 003903 号

出版者: 东北大学出版社

地址: 沈阳市和平区文化路 3 号巷 11 号

邮编: 110004

电话: 024-83687331(市场部) 83680267(社务室)

传真: 024-83680180(市场部) 83680265(社务室)

E-mail: neuph@neupress.com

http://www.neupress.com

印刷者: 东北大学印刷厂

发行者: 东北大学出版社

幅面尺寸: 184mm × 260mm

印 张: 19.375

字 数: 480 千字

出版时间: 2004 年 1 月第 1 版

印刷时间: 2004 年 1 月第 1 次印刷

印 数: 1~4000 册

责任编辑: 任彦斌

责任校对: 李 叶

封面设计: 任 何 唐 人

责任出版: 杨华宁

定 价: 20.00 元

前 言

C 语言作为非计算机专业大学生一门系列课程,现在仍然是绝大多数院校的主要基础课程之一,是非计算机专业大学生算法语言教学的主要载体。C 语言以它的功能丰富、表达能力强、使用灵活、应用面广、目标程序效率高、可移植性好、能对硬件直接进行操作等优点越来越赢得人们的青睐。加之用 C 语言可以编写出紧凑、高效、风格优美的程序和其结构化特性,C 语言已成为社会认可的程序语言。

但也正是 C 语言的灵活性(及指针等语法现象)等优点,使本应以算法教学为主的 C 语言教学变成了 C 语言的语法学习课程,极大地冲击了教学的核心内容和目的。同时还因为 C 语言语法现象过于复杂、书写过于灵活等特征,导致 C 语言成为一种不易入门的语言。而其不易入门性又限制了初学者的学习兴趣,从而最终导致 C 语言教学效果不够理想。C 语言教学效果较差,随着扩大招生而日益突出。

为了激发学生的学习兴趣,恢复其“学习一种程序设计语言,掌握基本算法和计算机化思维模式的程序设计语言”教学的本来目的,在多年教学实践的基础上我们重新设计了 C 语言的教学模式和课程内容安排形成了一种全新模式的 C 语言教材——“新概念 C 语言”。

“新概念 C 语言”突破了以往任何一种语言教材的旧的模式,将教学内容分为入门篇和提高篇两个篇章。在入门篇中只引进程序设计必要的语法现象,达到快速入门、激发兴趣的目的。在入门篇和提高篇之间插一个强化上机实验周,巩固学习内容。在提高篇中完成完整的语法、算法、程序设计思想等教学目的的学习任务。由于学生对语言已具有了初步的了解并掌握了最基本的语法和程序设计思想,能设计较简单的程序,所以在提高篇的学习中,不论对灵活语法的学习和掌握,还是对程序设计思想的掌握都更加容易,从而可以较容易达到教学目标。

入门篇主要包括下面语法现象:C 语言历史,简单 C 语言程序,标识符,整型常量、整型变量(int),实型常量、实型变量(float double),字符型常量、字符变量,算术运算符与算术表达式,关系运算符与关系表达式,逻辑运算符与逻辑表达式,顺序结构,选择结构之 if 语句,循环结构之 for 循环,一维数组,字符数组与字符串,函数和文件。

需要说明的是为了达到简化和容易入门的目的,在入门篇中介绍的都是最基本的语法现象,用这些虽基本但很完整的语法现象也可以完成复杂的程序设计任务。

另一个特殊的处理在于函数,本书函数内容(入门篇第 6 章)与以往任何教材不同的是,先讲了函数的定义和引用,而后介绍了库函数,在入门篇第 6 章即函数内容讲授之前,建议所有例子都不用输入/输出等函数,都直接进行赋值,用集成环境的 debug 功能观察结果,以达到强化学生程序调试能力和了解库函数不是语言内容的双重目的。

总的来讲,入门篇就是达到一个掌握基本语法现象,学会使用一些功能(例如文件功能)的目的,为提高篇学习打下良好的基础。

提高篇包含全部入门篇未涉及的 C 语言语法现象,但对内容顺序做了有目的的调整。这主要体现在以下几方面。

1. 先讲结构后讲指针,这又和所有现存教材不一样。在学习了基本数据类型和最简单的构造数据类型——数组之后,直接引入用户构造数据类型和用户自定义数据类型是很有必要的,而且先讲结构也为指针的最主要应用——链表的应用事先做了铺垫。结构是类的基础,

指针是 C 语言的灵魂但又是程序不稳定的主要原因,也是现在很多语言环境放弃指针的原因,本书教学中有意弱化指针内容,所以进行了如此的顺序调整。

2. 弱化了自加自减、逗号运算符和条件表达式等灵活语法,将所有这些语法现象都在提高篇的最后第 11 章讲授,目的是只要求学生了解,不要求掌握。

3. 本书还放弃了原来 C 语言教学中的“最短代码、最快执行”的教学理念,代之以“最可读、模块化程度高”的教学理念。因为现在的计算机内存不再成为紧缺资源,所以具体在提高篇的最后才介绍 short int 等节省内存的语法,在例子和教学中少用 int 和 float,多用 long 和 double 等。

4. 本书用大量的附录(1~12)来简化正文内容,让学生学会使用手册和环境的在线帮助,也不像有的教材将语法现象讲得非常细,而是让学生自己学会举一反三,融会贯通。

本书还附有基于 Internet 网的远程作业提交与批改系统,基于 Intranet 的 C/S 模式的无纸化 C 语言考试系统(模拟版),以适应网络化进程。

总之,本书通篇体现一个新字,通过我院的两轮教学使用取得了预期的教学效果。由于采用了与以往不同的教学模式和过程,所以在教学中还需注意以下问题:

(1) 在入门篇教学中,在介绍函数之前不能用任何 C 库函数,输入输出和变量观察只用 Turbo C 2.0 集成环境。而在函数一章也一改以往传统,先介绍函数定义,再讲库函数和为什么开发商要提供库函数。

(2) 入门篇文件这一章只要学生直接学习如何使用文件和为什么要用文件,而文件指针等在提高篇介绍。

(3) 在结束入门篇学习后进入提高篇,这时学生已经可以设计简单 C 程序并具备了最基本的程序设计思想,提高篇开篇将算法,从而确立算法在教学中的中心地位。

(4) 整篇的另一个特征是体现了最新的程序设计理念,结构算法应优化,但代码和内存节省可以不必考虑。这主要体现在程序例子要多用 float 数据类型,只在必须用 int 的地方用 int。而整型的各种变形都在最后简介。

(5) 在提高篇另一个与众不同之处是,本文先讲结构后讲指针,并弱化指针的重要性,直接介绍链表,强调指针在算法和数据结构中的应用,使学习更有针对性。

(6) 将 C 语言的灵活语法现象后移弱化。

李一波提出了本书的编著方案并编写了全部书稿;张森悦负责全部书稿的修改、校对和部分例题的编写工作,并提出了合理的部分内容调整方案;孙玉霞为本书编写了全部作业和附录四,王亚杰审校了全书并组织了院内配套实验教材。王亚杰、王晓斌、郭熹崑等参加了关于教材编写的讨论,并提出了宝贵的修改意见。考试软件由李一波、王晓斌、王亚杰、张森悦、刘学平等开发。

编 者
2004 年 1 月

目 录

第一部分 入门篇

第 1 章 C 语言的产生及其工作流程	3
1.1 C 语言的产生	3
1.2 程序和程序设计	3
1.2.1 C 程序	3
1.2.2 程序设计	4
习题	4
第 2 章 C 程序设计的初步知识	5
2.1 简单 C 程序的构成和格式	5
2.2 常量、变量和标识符	6
2.2.1 标识符	6
2.2.2 常量	7
2.2.3 用定义一个符号名的方法来代表一个常量	7
2.2.4 变量	8
2.3 整型数据	8
2.3.1 整型常量	8
2.3.2 整型变量	8
2.4 实型数据	9
2.4.1 实型常量	9
2.4.2 实型变量	9
2.5 字符型数据	10
2.5.1 字符常量	10
2.5.2 字符串常量	11
2.5.3 字符变量	11
2.6 算术表达式	11
2.6.1 基本的算术运算符	11
2.6.2 运算符的优先级、结合性和算术表达式	12
2.6.3 强制类型转换表达式	13
2.7 赋值表达式	13
2.7.1 赋值运算符和赋值表达式	13
2.7.2 赋值运算中的类型转换	14
习题	15
第 3 章 顺序结构程序	16
3.1 复合语句和空语句	16

3.1.1 复合语句	16
3.1.2 空语句	16
3.2 程序举例	16
习题	17
第 4 章 选择结构	19
4.1 关系运算和逻辑运算	19
4.1.1 C 语言中的逻辑值	19
4.1.2 关系运算符和关系表达式	19
4.1.3 逻辑运算符和逻辑表达式	20
4.2 if 语句和用 if 语句构成的选择结构	21
习题	22
第 5 章 循环结构	25
5.1 for 语句和用 for 语句构成的循环结构	25
5.1.1 for 语句构成的循环结构	25
5.1.2 for 循环的执行过程	25
5.1.3 有关 for 语句的说明	25
5.2 循环结构的嵌套	27
习题	28
第 6 章 函数	30
6.1 函数的定义和返回值	30
6.1.1 函数定义的语法	30
6.1.2 函数的返回值	30
6.2 函数的调用	32
6.2.1 函数的调用方式	32
6.2.2 函数调用时的语法要求	32
6.3 函数的说明	33
6.3.1 函数说明的形式	33
6.3.2 函数说明的位置	34
6.4 调用函数和被调用函数之间的数据传递	34
6.5 库函数	36
6.6 数据输入输出及常用库函数	37
6.6.1 printf 函数(格式输出函数)	37
6.6.2 scanf 函数(格式输入函数)	39
6.6.3 调用 putchar 和 getchar 函数输出和输入字符	40
6.7 程序举例	41
习题	43

第 7 章 数组	45
7.1 一维数组的定义和一维数组元素的引用	45
7.1.1 一维数组的定义	45
7.1.2 一维数组元素的引用	46
7.1.3 一维数组的初始化	46
7.1.4 一维数组的定义和数组元素引用举例	47
7.2 函数之间对一维数组和数组元素的引用	47
7.2.1 数组元素做实参	47
7.2.2 数组名做实参	49
7.3 一维数组应用举例	50
7.4 字符数组	53
7.4.1 字符数组的定义	53
7.4.2 字符数组的初始化	54
7.4.3 字符数组的引用	54
7.4.4 字符串与字符串结束标志	55
7.4.5 字符数组的输入输出	56
7.4.6 字符串处理函数	58
7.4.7 字符数组应用举例	60
习题	61
第 8 章 文件	63
8.1 C 语言文件的概念	63
8.2 打开文件	64
8.3 关闭文件	65
8.4 调用 <code>getc (fgetc)</code> 和 <code>putc (fputc)</code> 等常用函数进行输入和输出	65
8.5 判断文件结束函数 <code>feof</code>	68
习题	69

第二部分 提高篇

第 9 章 算法	73
9.1 算法	73
9.2 结构化程序设计和模块化结构	74
9.2.1 结构化程序	74
9.2.2 模块化结构	76
习题	76
第 10 章 结构体和用户定义类型	78
10.1 结构体类型	78

10.1.1 结构体类型的说明	78
10.1.2 结构体类型的变量、数组的定义	80
10.1.3 给结构体变量、数组赋初值	81
10.1.4 引用结构体变量中的数据	82
10.1.5 结构体数组应用举例	84
10.2 typedef 说明一种新类型名	85
习题	87
第 11 章 指针、链表和动态内存分配	89
11.1 地址和指针	89
11.2 指针变量的定义和指针变量的基类型	90
11.3 指针变量的赋值及基本操作	91
11.3.1 给指针变量赋地址值	91
11.3.2 给指针变量赋“空”值	91
11.3.3 通过指针来引用一个存储单元	92
11.3.4 移动指针	94
11.3.5 指针比较	95
11.4 函数之间地址值的传递	95
11.4.1 形参为指针变量时, 实参和形参之间的数据传递	95
11.4.2 通过传送地址值, 在被调用函数中直接改变调用函数中变量的值	96
11.4.3 函数返回地址值	97
11.5 动态存储分配	98
11.6 链表的建立及基本操作	100
11.6.1 结构体中含有可以指向本结构体的指针成员	100
11.6.2 动态链表的概念	101
11.7 数组指针和指向数组的指针变量	106
11.7.1 指向数组元素的指针	106
11.7.2 通过指针引用数组元素	106
11.7.3 数组名做函数参数	110
11.8 字符串的指针和指向字符串的指针变量	120
11.8.1 字符串的表示形式	120
11.8.2 字符串指针做函数参数	123
11.8.3 对使用字符指针变量和字符数组的讨论	125
11.9 用结构体变量和指向结构体的指针做函数参数	128
习题	130
第 12 章 多维数组	132
12.1 二维数组的定义和引用	132
12.1.1 二维数组的定义	132
12.1.2 二维数组的引用	132

12.2 二维数组的初始化.....	133
12.3 二维数组应用举例.....	134
12.4 多维数组的定义和引用.....	136
习题.....	136
第 13 章 选择结构	138
13.1 switch 语句以及用 switch 语句和 break 语句构成的选择结构.....	138
13.1.1 switch 语句.....	138
13.1.2 switch 语句的执行过程.....	138
13.1.3 在 switch 语句体中使用 break 语句.....	140
13.2 语句标号和 goto 语句 (选讲内容)	141
13.2.1 语句标号.....	141
13.2.2 goto 语句.....	141
13.3 嵌套的 if 语句 (选讲内容).....	142
习题.....	145
第 14 章 循环结构	147
14.1 while 语句和用 while 语句构成的循环结构	147
14.1.1 while 循环的一般形式.....	147
14.1.2 while 循环的执行过程.....	147
14.2 do-while 语句和用 do-while 语句构成的循环结构.....	149
14.2.1 do-while 语句的循环结构.....	149
14.2.2 do-while 循环的执行过程.....	149
14.3 break 和 continue 语句在循环体中的应用	151
14.3.1 break 语句.....	151
14.3.2 continue 语句.....	151
14.4 程序举例	152
习题.....	154
第 15 章 函数	157
15.1 函数的嵌套调用	157
15.2 函数的递归调用.....	160
15.3 带命令行参数的主函数 (指针数组做 main 函数的形参).....	166
15.3.1 指针数组.....	166
15.3.2 指针数组做 main 函数的形参.....	167
习题.....	169
第 16 章 文件	171
16.1 文件指针.....	171
16.2 流式文件操作.....	172
16.3 文件定位函数	173

16.3.1	fseek 函数	173
16.3.2	ftell 函数	174
16.3.3	rewind 函数	174
	习题	174
第 17 章	自加、自减、逗号和位操作符和条件表达式构成的选择结构	177
17.1	自加 (++)、自减 (--) 运算符和逗号运算符	177
17.2	复合的赋值表达式	178
17.3	逗号运算符和逗号表达式	178
17.4	位运算	178
17.4.1	位运算符	178
17.4.2	位运算符的运算功能	179
17.5	条件表达式构成的选择结构	181
	习题	182
第 18 章	其他变量类型、变量的作用域、变量的存储类别和编译预处理	184
18.1	其他变量类型	184
18.1.1	短整型和长整型数据	184
18.1.2	无符号型数据	184
18.1.3	共用体数据类型	184
18.2	局部变量和全局变量	187
18.2.1	局部变量	187
18.2.2	全局变量	188
18.3	变量的存储类别	191
18.3.1	动态存储方式和静态存储方式	191
18.3.2	auto 变量	192
18.3.3	用 static 声明局部变量	192
18.3.4	register 变量	194
18.3.5	用 extern 声明外部变量	196
18.3.6	用 static 声明外部变量	198
18.3.7	关于变量的声明和定义	198
18.3.8	存储类别小结	199
18.4	编译预处理	201
18.4.1	宏替换	201
18.4.2	文件包括	204
18.4.3	条件编译 (选讲内容)	204
18.5	如何运行一个多文件的程序	207
	习题	208
第 19 章	C++ 对 C 的扩充	211
19.1	C++ 的特点	211

19.2 最简单的 C++ 程序	211
19.3 C++ 的输入输出	212
19.3.1 用 cout 进行输出	213
19.3.2 用 cin 进行插入	214
19.4 函数的重载	214
19.5 带缺省参数的函数	217
19.6 变量的引用类型	218
19.6.1 引用的类型	218
19.6.2 引用的简单说明	218
19.6.3 引用作为函数参数	219
19.7 内置函数	221
19.8 作用域运算符	222
19.9 动态分配/撤销内存的运算符 new 和 delete	223
19.10 小 结	224
第 20 章 C++ 的面向对象基础	226
20.1 概述	226
20.2 类和对象	227
20.3 构造函数和析构函数	230
20.3.1 构造函数	230
20.3.2 析构函数	232
20.4 继承与派生	235
20.4.1 继承与派生的概念	235
20.4.2 建立派生类的方法	237
20.4.3 公用派生类	238
20.4.4 私有派生类	240
20.4.5 保护成员	241
20.4.6 派生类的构造函数	242
20.4.7 继承在软件开发中的重要意义	245

附录及参考文献

附录 I 美国标准信息交换码——ASCII 码表	246
附录 II C 语言中的关键字	247
附录 III C 语言类型转换规则	248
附录 IV C 语言中运算符的优先级和结合性	249
附录 V C 语言常用语法提要	252
附录 VI C 库函数	256
附录 VII C 语言转义字符	262

附录Ⅷ	常见错误分析和程序调试·····	263
附录Ⅸ	程序调试·····	277
附录Ⅹ	Turbo C 2.0 集成环境简介·····	279
附录Ⅺ	如何在 Windows XX 和无盘环境下使用 Turbo C 2.0·····	289
附录Ⅻ	计算机水平测试系统使用说明·····	295
参考文献	·····	297

第 一 部 分

人 门 篇

第 1 章 C 语言的产生及其工作流程

1.1 C 语言的产生

C 语言是国际上广泛流行的、很有发展前途的计算机高级语言。它适合于作为系统描述语言，既可用于写系统软件，也可用于写应用软件。

以前的操作系统等系统软件主要是用汇编语言编写的。由于汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好改用高级语言，但一般高级语言难以实现汇编语言的某些功能。人们设想能否找到一种既具有一般高级语言特性，又具有低级语言特性的语言，集它们的优点于一身。于是，C 语言就在这种情况下应运而生。

1.2 程序和程序设计

1.2.1 程序

当今，计算机已广泛应用于社会生活的各个领域，成为大众的现代工具。但是，不熟悉计算机的人仍然把它想像得十分神秘。其实，计算机不过是一种具有内部存储能力、由程序自动控制的电子设备。人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中，当人们给出命令之后，它就按指令操作顺序自动进行。人们把这种可以连续执行的一条条指令的集合称为“程序”。可以说，程序就是人与机器进行“对话”的语言，也就是人们所说的“程序设计语言”。

目前，在社会上使用的程序设计语言有上百种，它们都被称为计算机的“高级语言”，如 BASIC，PASCAL 以及本书的程序设计语言等。这些语言都是用接近人们习惯的自然语言和数学语言作为语言的表达形式，人们学习和操作起来感到十分方便。

但是，对于计算机本身来说，它并不能直接识别由高级语言编写的程序，它只能接受和处理由 0 和 1 的代码构成的二进制指令或数据。由于这种形式的指令是面向机器的，因此也称为“机器语言”。

人们把由高级语言编写的程序称为“源程序”，把由二进制代码表示的程序称为“目标程序”。如何把源程序转换成机器能够接受的目标程序，软件工作者编制了一系列软件，通

过这些软件可以把用户按规定语法写出的语句一一翻译成二进制的机器指令。这种具有翻译功能的软件称为“编译程序”。每一种高级语言都有与它对应的编译程序。例如，C 语言编译程序就是这样的一种软件，C 语言编译程序的功能如图 1-1 所示。



图 1-1 C 语言编译程序的功能

人们所写的每条 C 语句，经过编译（Compile）最终都将转换成二进制的机器指令。由 C 语言构成的指令序列称 C 源程序，按 C 语言的语法编写 C 程序的过程，称为 C 语言的代码编写。

C 语言源程序经过 C 语言编译之后生成一个后缀为 xx.OBJ 的二进制文件（称为目标文件）。最后还要由称为“连接程序”（Link）的软件，把此 xx.OBJ 文件与 C 语言提供的各种库函数连接起来生成一个后缀为 xx.EXE 的可执行文件。在 DOS 状态下，只需打入此文件的名字 xx（而不必打入后缀.EXE），该执行文件就可运行。

1.2.2 程序设计

简单的程序设计一般包含以下几个部分。

- ① 确定数据结构。根据任务书提出的要求、指定的输入数据和输出的结果，确定存放数据的数据结构。
- ② 确定算法。针对存放数据的数据结构来确定解决问题、完成任务的一步一步的步骤。有关算法的概念将在提高篇中介绍。
- ③ 编码。根据确定的数据结构和算法，使用稳定的计算机语言编写程序代码，输入到计算机并保存在磁盘上，简称编程。
- ④ 在计算机上调试程序。消除由于疏忽而引起的语法错误或逻辑错误，用各种可能的输入数据对程序进行测试，使之对各种合理的数据都能得到正确的结果，对不合理的数据能进行适当的处理。
- ⑤ 整理并写出文档资料。

习 题

- 1.1 C 语言有哪些优点？
- 1.2 如何理解程序和程序设计语言，设计一个简单的程序应包含哪几部分？
- 1.3 上机运行 C 程序一般需要哪几个步骤？生成哪些文件？