



应用型本科计算机规划教材

高级语言程序设计

学习与实验指导

黄翠兰 主编



厦门大学出版社
XIAMEN UNIVERSITY PRESS

高级语言程序设计 学习与实验指导

主 编 黄翠兰

副主编(以姓氏笔画为序)

王小峰 陈 敏

郭永宁 翁 伟

厦 门 大 学 出 版 社

图书在版编目(CIP)数据

高级语言程序设计学习与实验指导/黄翠兰主编. —厦门:厦门大学出版社, 2009. 6
(应用型本科计算机规划教材)

ISBN 978-7-5615-3256-0

I. 高… II. 黄… III. C 语言-程序设计-高等学校-教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 088049 号

厦门大学出版社出版发行

(地址:厦门市软件园二期望海路 39 号 邮编:361008)

<http://www.xmupress.com>

xmup@public.xm.fj.cn

厦门金凯龙印刷有限公司印刷

2009 年 6 月 1 版 2009 年 6 月第 1 次印刷

开本:787×1092 1/16 印张:22.75

字数:535 千字 印数:1~3000 册

定价:33.00 元

如有印装质量问题请与承印厂调换

内容简介

本书是《高级语言程序设计》(黄翠兰主编,厦门大学出版社,2008年版)的配套学习与实验指导书,但也可以独立使用。全书分为五章及一个附录。第1章包括《高级语言程序设计》各章节的要点回顾、例题分析及习题解答;第2章给出了一些典型的课程设计任务书;第3章是各章节配套的实验内容及实验步骤;第4章给出了两套模拟试题及其参考答案;第5章包括20个常见问题及为解决这些问题而编写的C程序;附录部分包括四个较复杂完整项目的解决方案,供读者参考、学习、仿照。

本书是作者多年教学实践、上机辅导、软件开发、交流沟通的经验总结,力求将C语言的知识点融入具体的学习、实验、课程设计中,培养读者的实际操作能力。读者在学习过程中,可以在学习每个知识点后,通过阅读“要点回顾”复习所学知识,并通过“例题分析”加强对知识点的理解;然后完成本书第三章的相应实验内容;在有余力的情况下,可以多做一些习题,也可对照“习题解答”中的参考程序代码自检,日积月累不断地提高程序设计能力。

前言

在C语言的初级学习阶段,实践是最好的老师。对于一个程序员来说,编程经验好比一个武者随身携带的用熟了的兵器,随时会起到非常重要的作用。而从实践的点滴中积累起来的,就是所谓的经验。通过不断的实际上机编程,不但能够记下那些常用的函数和代码,熟悉程序的结构,从而使初学者暂时脱离书本来编写程序,而且能够积累各方面的编程经验。

若把程序设计语言当作一种自然语言来看待,那么用哑呀学语来形容一个程序员的入门学习就是一种恰当的比喻了。在语言学习的第一阶段,模仿是不可避免的,也是一种非常重要的学习形式。因此,初学者不妨先理解并上机调试书上一些简单的程序,然后加入一些自己的思路,通过比较分析并解决新的问题,达到学习并使用语句的目的。

程序设计是一门科学,也是一门艺术。培养良好的程序设计技能、技巧需要掌握很多知识,不只是记住某种开发语言的规则、语法,更重要的是必须通过阅读程序以及不断地进行编程实践,日积月累就形成了自己独特的编程风格。

编程时,既要遵守规范,又可以有个人的独到之处。良好的编程习惯有助于对程序本身的理解。在编写C程序时,应注意以下几点:

- (1)程序代码中穿插一些必要的注释。
- (2)标识符命名见名知义。
- (3)程序语句书写格式规范,对于嵌套语句使用缩进。
- (4)以模块化方式考虑程序结构,以函数形式书写较复杂的程序。

不管一个程序员多么有经验,都不能保证自己编写的程序一次就能通过编译并运行成功。或多或少会出现一些问题,有时是无心之漏,有时是思维方式或逻辑关系错误。事实上,处理程序中的错误,特别是逻辑错误是程序开发过程中重要的一环。一旦程序比较庞大而复杂时,调试工作就将变得异常艰巨。程序的调试过程包括发现错误、定位错误、修改错误。

作为计算机专业学习的一大主要内容,编程语言的学习过程将漫长而艰辛,但也乐趣无穷。而学好第一门编程语言,将会为以后学习其他编程语言、计算机课程提供一个良好的平台。一个优秀的计算机专业毕业生,必须是一个优秀的程序员。

提高程序设计能力的一个重要途径是学习书上别人编写的程序,从中掌握解决问题的核心方法和关键步骤。当你知道语言的特点和构造以后,真正要掌握一种语言的途径就是大胆地编写程序,循序渐进,勤思考,勤动手。当你绞尽脑汁编写好一个程序,实现一个算法时,成功的喜悦会让你体会到学习的快乐,并激励着你攻克更大的困难,编写出更完善的程序,解决更复杂的算法。

对于初学者,学习编程不会是一蹴而就的事。笔者经过二十多年的编程历程,有以下几点深刻体会:





1. 纸上得来终觉浅,绝知此事要躬行。——实践是工科学生学习过程中非常重要的一个环节。近千年前,南宋诗人陆游就用这诗句告诉他的儿子:“从书本上得到的知识终归是浅薄的,未能理解知识的真谛。要真正理解书中的深刻道理,必须亲身去躬行实践。”

2. 学习编程,有两颗必不可少的“心”:耐心和细心。

3. 学习的过程要经过几个反复,才能前后贯穿,积累应该掌握的知识。要记住“曙光在前头”和“千金难买回头看看”。“千金难买回头看看”是学习知识的重要方法,就是说,学习后面的知识,不要忘了回头弄清遗留下的问题和加深对前面知识的理解。

4. 上机运行所有程序设计实例,有能力的,所有习题都做一遍。

5. 少玩游戏,多做题。游戏具有强烈的吸引力,大家都知道玩游戏会上瘾,可很多人会控制不住,浪费了大量的时间和精力,把许多该做的事耽搁了。有些事情一经耽搁,就再也没有机会补救了,比如老师上课的时候没有注意听,当你想学、想听的时候,老师已经不再讲你错过的那些内容了。

6. 记住西方的一句格言:Do the right things at the right time.

本书作为《高级语言程序设计》一书的配套学习与实验指导书,重在让读者回顾各章节的知识点的同时,用例题分析重点、难点,加深读者对程序语言语法以及编程要点的理解。在此基础上,读者通过独立完成各部分的实验内容,利用所学知识编写程序。最后通过模拟题检查自己对该程序设计语言各个知识点的掌握情况,查找差距并发现存在的不足,及时温故和强化。

本书由厦门理工学院黄翠兰任主编,并负责全书的统稿、定稿;福建师范大学福清分校郭永宁、王小峰,福建工程学院陈敏及厦门理工学院翁伟任副主编。具体分工如下:黄翠兰负责第一章第1节、第二章一部分、第三章、第四章及附录一部分,王小峰负责第一章第6、7、8、10节,陈敏负责第一章第5、11、12、13节及附录一部分,郭永宁负责第一章第3、4节,翁伟负责第一章第2、9节、第二章一部分及附录一部分。

在本书写作过程中,得到许多同事、朋友的关心,感谢所有支持和关心本书出版的人们!特别感谢厦门大学出版社宋文艳副总编辑及睦蔚老师的热心指导和大力支持!

我们精益求精,但难免存在一些不足之处,恳请读者批评指正。如果您在使用本书时遇到问题,可以发邮件到 clhuang@xmut.edu.cn 与我们联系。

最后用一首诗与大家共勉。

劝君莫惜金缕衣,
劝君须惜少年时。
花开堪折直须折,
莫待无花空折枝。

——《金缕衣》杜秋娘(唐)

黄翠兰
2009年4月

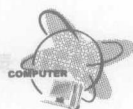


目 录

前言

第 1 章 基础知识及习题解答	1
1.1 第一章 C 程序设计概述	1
1.1.1 要点回顾	1
1.1.2 例题分析	2
1.1.3 C 语言学习:从入门到精通	3
1.1.4 习题解答	13
1.2 第二章 基本数据类型、运算符和表达式	17
1.2.1 要点回顾	17
1.2.2 例题分析	18
1.2.3 习题解答	20
1.3 第三章 程序的控制结构	26
1.3.1 要点回顾	26
1.3.2 例题分析	30
1.3.3 习题解答	35
1.4 第四章 数组	63
1.4.1 要点回顾	63
1.4.2 例题分析	66
1.4.3 习题解答	71
1.5 第五章 结构体与共用体	90
1.5.1 要点回顾	90
1.5.2 例题分析	93
1.5.3 习题解答	99
1.6 第六章 函数	106
1.6.1 要点回顾	106
1.6.2 例题分析	109
1.6.3 习题解答	113
1.7 第七章 指针	123
1.7.1 要点回顾	123
1.7.2 例题分析	124
1.7.3 习题解答	132



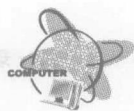


1.8 第八章 指针的应用——链表	144
1.8.1 要点回顾	144
1.8.2 例题分析	146
1.8.3 习题解答	150
1.9 第九章 位运算	163
1.9.1 要点回顾	163
1.9.2 例题分析	163
1.9.3 习题解答	164
1.10 第十章 文件	167
1.10.1 要点回顾	167
1.10.2 例题分析	168
1.10.3 习题解答	175
1.11 第十一章 编译预处理	190
1.11.1 要点回顾	191
1.11.2 例题分析	193
1.11.3 习题解答	195
1.12 第十二章 C语言绘图功能简介	197
1.12.1 要点回顾	197
1.12.2 例题分析	201
1.12.3 习题解答	202
1.13 第十三章 综合实例	211
1.13.1 要点回顾	211
1.13.2 例题分析	211
1.13.3 习题解答	214
第2章 C语言课程设计题目精粹	224
2.1 课程设计任务书1	224
2.1.1 基本编程题	224
2.1.2 猜数字游戏	224
2.1.3 三角形绘制程序	225
2.1.4 地图的着色问题	225
2.1.5 爬山问题	225
2.1.6 求大数阶乘的位数及具体数值	225
2.2 课程设计任务书2	226
2.2.1 图书管理系统	226
2.2.2 打字测试系统	226
2.2.3 学生成绩管理系统	228
2.2.4 模拟下雨	229



2.2.5 学生考勤系统	229
2.2.6 高校工资管理系统	229
2.2.7 具备通信录功能的系统	230
第3章 C语言上机实验	231
3.1 实验一 Turbo C 实验环境及配置	231
3.1.1 实验目的与要求	231
3.1.2 实验过程与内容	231
3.2 实验二 C语言数据类型	234
3.2.1 实验目的与要求	234
3.2.2 实验过程与内容	234
3.3 实验三 C语言程序结构	235
3.3.1 实验目的与要求	235
3.3.2 实验过程与内容	236
3.4 实验四 数组	237
3.4.1 实验目的与要求	237
3.4.2 实验过程与内容	237
3.5 实验五 结构体和共用体	238
3.5.1 实验目的与要求	238
3.5.2 实验过程与内容	239
3.6 实验六 函数	239
3.6.1 实验目的与要求	239
3.6.2 实验过程与内容	240
3.7 实验七 指针	241
3.7.1 实验目的与要求	241
3.7.2 实验过程与内容	241
3.8 实验八 指针的应用——链表	242
3.8.1 实验目的与要求	242
3.8.2 实验过程与内容	242
3.9 实验九 位运算	243
3.9.1 实验目的与要求	243
3.9.2 实验过程与内容	243
3.10 实验十 文件	243
3.10.1 实验目的与要求	243
3.10.2 实验过程与内容	243
3.11 实验十一 编译预处理	244
3.11.1 实验目的与要求	244
3.11.2 实验过程与内容	244





3.12 实验十二 C语言绘图功能简介	244
3.12.1 实验目的与要求	244
3.12.2 实验过程与内容	245
3.13 实验十三 综合实例	245
3.13.1 实验目的与要求	245
3.13.2 实验过程与内容	245
第4章 典型算法及其源程序	247
第5章 测试题	264
5.1 模拟试卷一	264
5.2 模拟试卷二	273
5.3 模拟试卷参考答案	282
5.3.1 模拟试卷一参考答案	282
5.3.2 模拟试卷二参考答案	285
附录	289
1. 打字测试系统参考程序代码	289
2. 图书管理系统的源程序参考代码	315
3. 具备通信录功能的系统的参考程序代码	339
4. 学生成绩简易管理系统的源程序参考代码	349



第 1 章

基础知识及习题解答

1.1 第一章 C 程序设计概述

1.1.1 要点回顾

1. 一个计算机系统的物理组件组成了计算机硬件,这些组件是看得见、摸得着的。在硬件能够工作之前必须给计算机编程,以便告诉硬件应该做什么。这些程序通常称为软件。只有硬件和软件结合在一起,计算机才能进行指定的计算,解决相应的问题。

2. 程序设计语言就是人机交流的语言,称为程序设计语言或计算机语言。计算机语言通常分为三类:机器语言、汇编语言和高级语言。

3. 算法是解决问题的一种策略。算法可以用自然语言描述,也可以用流程图来表示。常见的流程图有传统流程图和 N-S 流程图。

4. 结构化程序设计包括三种控制结构:顺序结构、选择结构和循环结构。它的基本思路是:把一个复杂问题的求解过程分阶段进行,每个阶段所处理的问题都控制在人们易理解和处理的范围内。即:自顶而下,逐步细化;模块化设计;结构化编码。

5. C 语言程序的特点:

(1)C 语言是一种结构化的程序设计语言,包括顺序结构、选择结构和循环结构。

(2)C 语言中提供的数据类型有有符号基本整型、有符号短整型、有符号长整型、无符号基本整型、无符号短整型、无符号长整型、双精度实型、单精度实型、字符型、枚举类型、指针类型、文件等,以及由上述类型构造的类型,如数组、结构体、共用体等。

(3)C 语言具有与汇编语言的接口。在具有汇编编译器的情况下,可以在 C 程序中调用汇编子程序或直接插入汇编指令,也可以在汇编语言程序中调用 C 函数。

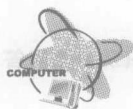
(4)C 程序中区分大小写。一般用户自定义标识符时,要能见名知义,如 count、PRICE 等。

(5)C 程序由函数组成,当且仅当包含一个 main 函数时,C 程序才能被单独运行,其他函数被 main 函数直接或间接调用。

(6)C 语言提供了丰富的函数,如输入输出函数(`printf()`和 `scanf()`函数)、数学函数、图形处理函数等。

(7)C 语言的头文件功能。可以把实现一定功能的函数或符号常量、类型定义等集中在一起,保存为头文件,供其他程序引用。Turbo C 提供了丰富的头文件,包括 `stdio.h`、`math.h`、`graphics.h` 等。





6. C 语言是一种编译型的程序设计语言。一个 C 程序要经过编辑、编译、链接和运行四个步骤,才能得到程序的执行结果。

7. 程序设计语言有一套语法规则以判断一个程序是否正确构建。编译器首先检查程序的语法是否正确,若违反了语法规则,编译器将显示出错信息。由于违反了语法规则而导致的错误称为语法错误(syntax error)。若由于逻辑上的错误导致程序不能正确地解决问题,这种类型的错误称为逻辑错误。找出并修正逻辑错误的过程称为调试。

8. 编程时,既要遵守规范,又可以有个人的独到之处。良好的编程习惯有助于对程序本身的理解。

1.1.2 例题分析

1. 计算机能直接识别的语言是()。

- A) 机器语言 B) 汇编语言 C) 高级语言 D) 面向对象语言

分析:现代计算机硬件只能直接识别二进制代码的程序,而在四个选项中只有机器语言程序是二进制代码,所以 A 选项是正确的。汇编语言必须经过汇编翻译成二进制的目标程序才能被执行。高级语言和面向对象语言必须经过编译、链接才能得到可执行文件。

答案:A。

2. 在 C 语言源程序中,以下叙述正确的是()。

- A) main 函数必须位于文件的开头
B) 每行只能写一条语句
C) 程序中的一个语句可以写成多行
D) 每个语句的最后必须有点号

分析:C 程序的书写很灵活,一行可以写多个语句,一个语句可以写成多行(如 if 语句、for 语句等),但要求每个语句的最后必须有分号。C 程序只执行 main 函数,其他函数都是由 main 函数直接或间接调用。但 main 函数不一定要位于文件的开头。

答案:C。

3. 结构化程序设计过程的基本思路不包括()。

- A) 最优化 B) 自顶而下,逐步细化
C) 模块化设计 D) 结构化编码

分析:结构化化程序设计的基本思路是,把一个复杂问题的求解过程分阶段进行,每个阶段所处理的问题都控制在人们容易理解和处理的范围内。具体地说,采用如下方法保证得到结构化的程序:自顶向下,逐步细化;模块化设计;结构化编码。

答案:A。

4. 良好的编程习惯有助于对程序本身的理解,在编写 C 程序时,应注意避免()。

- A) 程序代码中穿插一些必要的注释
B) 标识符命名字符数越少越好
C) 程序语句书写格式规范,对于嵌套语句使用缩进



D)以模块化方式考虑程序结构,以函数形式书写较复杂的程序

分析:A、C和D所描述的都是良好的编程习惯,而在标识符命名时应做到“见名知义”,以便程序理解和阅读。

答案:B。

1.1.3 C语言学习:从入门到精通

1. 什么是C语言

(1) Turbo C语言的发展

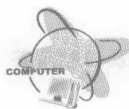
Turbo C是美国 Borland 公司的产品,Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件,如 Turbo BASIC、Turbo Pascal、Turbo Prolog,这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品,其中使用了全然一新的集成开发环境,使用了一系列下拉式菜单,将文本编辑、程序编译、链接以及程序运行一体化,大大方便了程序的开发。1988 年,Borland 公司又推出 Turbo C 1.5 版本,增加了图形库和文本窗口函数库等,而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能,并可以在 Tiny 模式下直接生成 .COM(数据、代码、堆栈处在同一 64 k 内存中)文件,还可对数学协处理器(支持 8087/80287/80387 等)进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++,它继承并发展了 Turbo C 2.0 的集成开发环境,包含面向对象的基本思想和设计方法。

(2) Turbo C 2.0 集成软件简介

Turbo C 2.0 集成软件可以直接从网上下载,只需将包含 Turbo C 2.0 集成开发环境的整个 TC 目录的内容复制到计算机中就可以了。下面对 Turbo C 2.0 的主要文件作一简单介绍:

TC. EXE	集成开发环境的可执行程序
TCINST. EXE	集成开发环境的配置设置程序
TCHELP. TCH	帮助文件
THELP. COM	读取 TCHHELP. TCH 的驻留程序
README. COM	关于 Turbo C 的信息文件
TCCONFIG. EXE	配置文件转换程序
MAKE. EXE	项目管理工具
TCC. EXE	命令行编译
TLINK. EXE	Turbo C 系列连接器
TLIB. EXE	Turbo C 系列库管理工具
C0?. OBJ	不同模式启动代码
C?. LIB	不同模式运行库
GRAPHICS. LIB	图形库
*. H	Turbo C 头文件(一般在 INCLUDE 文件夹中)
*. BGI	不同显示器图形驱动程序(一般在 BGI 文件夹中)



*.C

Turbo C 应用程序(源文件)

其中,上面的“?”可以是以下六种情况,分别表示 Turbo C 的六种运行模式。

T	Tiny(微型模式)
S	Small(小模式)
C	Compact(紧凑模式)
M	Medium(中型模式)
L	Large(大模式)
H	Huge(巨大模式)

(3)C 语言的特点

C 语言发展如此迅速,成为最受欢迎的语言之一,主要因为它具有强大的功能。许多著名的系统软件,如 dBASEIII Plus、dBASE IV 都是用 C 语言编写的。而且 C 语言还能和许多其他语言和软件结合使用,如汇编、MATLAB 等,例如 PC-DOS、WordStar 就是这种结合的产物。

C 语言具有下列特点:

①C 语言是中级语言

它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。

②C 语言是结构化语言

结构化语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可方便地互相调用,并具有多种循环、条件语句,控制程序流向,从而使程序完全结构化。

③C 语言功能齐全

C 语言具有多种数据类型,并引入了指针概念,可使程序效率更高。另外 C 语言也具有强大的图形功能,支持多种显示器和驱动器,而且计算功能、逻辑判断功能也比较强大,可以方便地实现决策目的。

④C 语言适用范围大

C 语言还有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX,也适用于多种机型。

2. C 语言能做什么,怎么学习 C 语言

在刚开始学习语言时,可以先知其然而不知其所以然,能够依样画葫芦就可以了。但是很多学生在刚开始学习 C 语言时就经常问“C 语言能做什么”。以下是关于这个问题的简单并不全面的回答,读者可以从 Internet 上查找到更多更全面的介绍和解释。

- 可以解决一些常见的计算问题,可以编写小型 MIS 系统,如图书管理系统、考勤管理系统等;
- 可以编写操作系统,比如我们用的 Windows、Linux 的底层代码,大部分都是用 C/C++ 语言编写的;
- 可以写游戏,很受热捧的“魔兽世界”用的语言就是 C/C++;
- C 语言还广泛应用于嵌入式开发、单片机等。

在 C 语言的学习期间必须勤动手、多实践,知道 C 语言的基本数据类型、基本结构的同



时,一定要结合例题、习题学会使用这些知识解决简单问题,然后逐渐将应用范围扩大,因为学习是分阶段的。

(1)初步学习阶段。可以用C语言来描述和解决一些简单的问题,如求 $n!$ 、求一元二次方程的根、求100~200之间的所有素数、求出Fabonacci序列的前40个数或对班级同学的姓名进行排序等。

这个阶段的学习主要是熟悉C语言的基本语法语义特性,熟练地在C语言IDE集成环境中进行编辑、编译、链接并运行程序,能够理解书本中的实例程序并参照编写程序解决一些常见的数学问题。

(2)全面掌握阶段。在基本掌握了C语言的主要知识点,能熟练地运用C语言时,便可以用C语言编写小型的应用程序了,如班级的同学录、寝室内务管理系统或银行取款机模拟系统等。但大家最感兴趣的可能还是能够编写一些小型游戏,从简单的“贪吃蛇”到较复杂的“扫地雷”再到自己设计的战略游戏等,都可以尝试着编写。有兴趣再加上一定的想象力,一定能编写出自己满意的游戏软件。

另外,在这个阶段,还可以尝试着用C语言来对计算机硬件进行控制和操作。也可以利用C语言来进行一些复杂的基于数据结构的算法研究,如我们所熟知的压缩软件WinRAR,在学习它的编码协议时,便可以用C语言来进行研究和验证。

(3)深入学习阶段。这个阶段可着眼于将C语言应用于各种社会实践,并注重与其他领域其他学科的知识结合起来。例如,信号采集程序和音频解压缩算法都可以用C语言来实现。而PIC单片机编程、DSP编程所采用的语言也都从原来的汇编语言逐渐转向了C语言。

实际上,由于C语言的功能强大、可移植性好以及对硬件控制能力高这些特点,C语言不但可用来编写应用软件,也可用来编写系统软件。

总之,不论一个C程序员处于哪个学习阶段,都可以用C语言来解决一些实际问题,关键在于怎样去发现并用C语言来解决这些问题,并积极思考将解决问题的方法与C语言的学习紧密地结合起来。注意平时的一点一滴,包括书本知识和上机实践。在这个过程中,逐渐形成个性化的编程风格和编程思想。

3. 培养优秀的C程序员

(1)掌握语言本质

在初学C语言时,可能会对有些问题理解不透,或者表达方式与以往数学学习中不同(如运算符等),这就要求初学者不气馁,遇到不明白的地方多问多想,鼓足勇气进行学习,待学完后面章节的知识,前面的问题也就迎刃而解了。

学不好C程序设计课程的同学大多是因为一开始遇到困难就放弃。在学习C语言的过程中,始终要记住“曙光在前头”和“千金难买回头看”。“千金难买回头看”是学习知识的重要方法,就是说,学习后面的知识,不要忘了回头弄清前面遗留下的问题和加深理解前面的知识,这是非常重要的。学习C语言就是要经过几个反复,才能前后贯穿,积累应该掌握的C语言知识。不要一遇到些许困难就轻言放弃。

那么,我们如何学好“C语言程序设计”呢?

①理解C语言的数据类型





一门编程语言课的基础知识实际上无非就包括这门语言所提供的数据类型以及程序结构。数据类型决定了程序将如何存储数据、处理数据(包括输入、处理和输出)。对于一个基本整型(int)变量,它在微型计算机内存中将占用两个存储单元(16位),以补码的方式来保存变量的值,可以用“%d”的格式进行输入输出,可进行算术运算、关系运算。如果是一个字符(char)变量呢?字符变量只占用一个字节的存储空间,保存字符的ASCII码,通常用“%c”的格式进行输入输出,也可进行算术运算、关系运算。但如果是一个结构体变量呢?请考虑。

②学好C语言的运算符和运算顺序

这是学好“C语言程序设计”的基础。C语言的运算非常灵活,功能十分丰富,运算种类远多于其他程序设计语言。在表达式方面较其他程序语言更为简洁,如自加、自减、逗号运算和三目运算使表达式更为简单,但初学者往往会觉得这种表达式难读,关键原因就是运算符和运算顺序理解不透不全。当多种不同运算组成一个运算表达式,即一个表达式中出现多种运算符时,运算的优先顺序和结合规则显得十分重要。在学习中,只要我们对此合理地进行分类,找出它们与数学中所学到运算之间的不同点之后,记住这些运算也就不困难了,有些运算符在理解后更会牢记心中,将来用起来得心应手。

要明确运算符按优先级不同分类。C语言运算符可分为15种优先级,从高到低,优先级为1~15,除第2、3级和第14级为从右至左结合外,其他都是从左至右结合,它决定同级运算符的运算顺序。

③学好C语言的四种程序结构

● 顺序结构

顺序结构的程序设计是最简单的,只要按照解决问题的顺序写出相应的语句就行,它的执行顺序是自上而下,依次执行。

例如: $a=3, b=5$,现交换 $a、b$ 的值,这个问题就好像交换两个杯子里的水,当然要用到第三个杯子,假如第三个杯子是 c ,那么正确的程序为:

```
c=a;a=b;b=c;
```

执行结果是 $a=5, b=c=3$ 。

如果改变其顺序,写成:

```
a=b;b=c;c=a;
```

则执行结果就变成 $a=b=c=5$,不能达到预期的目的,初学者应避免犯这种错误。

顺序结构可以独立使用构成一个简单的完整程序,常见的输入、计算、输出“三部曲”的程序就是顺序结构。例如计算圆的面积,其程序的语句顺序就是输入圆的半径 r ,计算 $s=3.14159 * r * r$,输出圆的面积 s 。

不过大多数情况下顺序结构都是作为程序的一部分,与其他结构一起构成一个复杂的程序,如分支结构中的复合语句、循环结构中的循环体等。

● 分支结构

顺序结构的程序虽然能解决计算、输出等问题,但不能做判断再选择。对于要先做判断再选择的问题就要使用分支结构。分支结构的执行是依据一定的条件选择执行路径,而不是严格按照语句出现的物理顺序。分支结构的程序设计方法的关键在于构造合适的分支条件和分



析程序流程,根据不同的程序流程选择适当的分支语句。分支结构适合于带有逻辑或关系比较等条件判断的计算,设计这类程序时往往都要先绘制程序流程图,然后根据程序流程写出源程序,这样做把程序设计分析与语言分开,使得问题简单化,易于理解。程序流程图是根据问题分析所绘制的程序执行流程图。

学习分支结构不要被分支嵌套所迷惑,只要正确绘制出流程图,弄清各分支所要实现的功能,嵌套结构也就不难了。嵌套只不过是分支中又包括分支语句而已,不是新知识,但要耐心地理解清楚,按缩进方式编写、理解程序。

● 循环结构

循环结构可以减少源程序重复书写的工作量,用来描述重复执行某段算法的问题,这是程序设计中最能发挥计算机特长的程序结构。C语言中提供四种循环,即 goto 循环、while 循环、do-while 循环和 for 循环。四种循环可以用来处理同一问题,一般情况下它们可以互相替换,但不提倡用 goto 循环,因为强制改变程序的执行顺序经常会给程序的运行带来不可预料的错误。我们主要学习 while、do-while、for 三种循环。

学习常用的三种循环结构的重点在于弄清它们相同与不同之处,以便在不同场合下使用。这就要清楚三种循环的格式和执行顺序,将每种循环的流程图理解透彻后就会明白如何替换使用,如把 while 循环的例题,用 for 语句重新编写一个程序,这样能更好地理解它们的作用。特别要注意在循环体内应包含趋于结束的语句(即循环变量值的改变),否则就可能成为一个死循环。

在学完这三个循环后,应明确它们的异同点:用 while 和 do-while 循环时,循环变量的初始化操作应在循环体之前,而 for 循环一般在语句的“表达式 1”中进行;while 循环和 for 循环都是先判断条件,后执行循环体,而 do-while 循环是先执行循环体后判断条件,也就是说 do-while 的循环体最少会被执行一次,而 while 和 for 循环的循环体则可能一次都不执行。另外,还应注意的是这三种循环都可以用 break 语句跳出循环,用 continue 语句结束本次循环,而 goto 语句与 if 构成的循环,是不能用 break 和 continue 语句进行控制的。

顺序结构、分支结构和循环结构并不是彼此孤立的,在循环中可以有分支、顺序结构,分支中也可以有循环、顺序结构。其实不管哪种结构,均可广义地把它看成一个语句。在实际编程过程中常将这三种结构相互结合以实现各种算法,设计出相应程序,但是要解决的问题较大时,编写出的程序就往往很长,结构重复多,造成可读性差,难以理解,解决这个问题的方法是将 C 程序设计成模块化结构,用多个函数一起来实现。

● 模块化程序结构

C 语言的模块化程序结构用函数来实现,即将复杂的 C 程序分为若干模块,每个模块都编写成一个 C 函数,然后通过主函数调用函数及函数调用函数来实现大型问题的 C 程序编写,因此一个 C 程序可以看成是由一个主函数(main 函数)和若干个子函数组成的,即一个 C 程序 = 1 个 main 函数 + 若干个自定义函数。

④ 掌握一些简单的算法

编程的一大部分工作其实就是分析问题找到解决问题的方法,再以相应的编程语言写出代码。这就要求掌握算法,如两个数的交换、三个数的比较、选择法排序和冒泡法排序,要求我

