DOI:10.13718/j.cnki.xsxb.2023.05.015

面向数据结构的“C 程序设计”教学改进策略①

李向华[1], 王震[2], 高超[1]

1. 西北工业大学 光电与智能研究院,西安 710000;2. 西北工业大学 网络空间安全学院,西安 710000

摘要:文章分析了 C 语言和数据结构两门课程之间的关系. C 语言学习的效果直接影响学生对数据结构课程的学习兴趣和掌握程度,因此,在学习 C 语言的过程中,有必要考虑后续数据结构如何“无痛”开展,做好两门课程的衔接. 针对当前 C 语言教学内容中存在的问题,提出了相应的解决办法,不仅让学生有重点地学好 C 语言,为学好数据结构做铺垫,同时,也可以培养学生的逻辑思维能力、算法设计能力和创新能力.

关 键 词:C 语言;数据结构;逻辑思维;算法设计;创新

中图分类号:TP399 **文献标志码**:A **文章编号**:1000-5471(2023)05-0111-05

C Programming Teaching Methods Oriented Data Structure

LI Xianghua[1], WANG Zhen[2], GAO Chao[1]

1. *School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an 710000, China*;
2. *School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710000, China*

Abstract: This paper analyzed the relationship between C Programming and Data Structure. Effects from learning C Programming directly affect students' interest in Data Structure. It also affects whether they have the ability to apply data structure. Therefore, it is necessary to consider how to connect the two courses in order to learn data structure easily and smoothly. This paper provides some methods to solve the problems existing in the current C Programming teaching. Students not only can learn C Programming well with the methods, but also can make good groundwork for Data Structure. Further, Students' abilities, such as logical thinking, algorithm designing and innovation, will be improved.

Key words: C programming; data structure; logical thinking; algorithm designing; innovation

瑞士计算机科学家尼古拉斯·沃斯(Niklaus Wirth)提出的影响深远的“算法+数据结构=程序”(Algorithm+Data=Programs)[1]公式,成为计算机科学特别是软件科学的指导思想,同时这个公式也说明了程序设计语言和数据结构这两门课程之间的关系[2].

C 程序设计和数据结构是计算机相关专业两门非常重要的基础必修课. 利用 C 语言实现的数据结构必然以 C 程序设计为先导课程,利用 C 语言具体表达数据在计算机中的存储和算法实现. 学生对数据结构相关知识的学习,不仅理解了数据结构、算法和程序之间的关系,也是对 C 语言知识的巩固和深化,使学生

① 收稿日期:2022-11-22
基金项目:陕西省高等教育学会高等教育科学研究项目(XGH21036);西北工业大学教育教学改革研究项目(2023JGY59);西北工业大学高等教育研究基金项目(GJJJM202203).
作者简介:李向华,副教授,博士,主要从事多模态认知计算、群体智能决策等方向的研究.

具有解决实际问题的能力. 所以,C程序设计和数据结构关系紧密,在一定程度上影响学生对计算机学科的学习兴趣. 然而,笔者在翻阅大量的C程序设计相关教材时发现,C语言课程过多注重其语法体系,授课教师也基本按照教材的流程进行讲解,因此,很多学生学习过C语言后,不了解其实际应用效果,也不能顺畅地编写程序. 在学习数据结构课程过程中,无论算法设计还是程序书写上,尤其涉及到指针部分,学生都觉得晦涩难懂,不会灵活应用,上机实践时更是无从下手. 这些现象说明教师在讲授C语言和学生在学习C语言的过程中没能结合适应数据结构的方法.

基于此,有必要在C语言的教学中,为数据结构课程的开展做些铺垫,让学生在一种轻松、有兴趣的氛围中学习,让数据结构变成相对容易学习的学科. 同时,通过C语言的学习,培养学生的逻辑思维能力、算法设计能力和创新能力[3-5].

1 C语言与数据结构的关系

C语言课程的内容主要围绕结构化程序设计展开,即顺序、选择、循环三个基本结构[6-7],为了能实现这三种结构,前期要学习数据类型、输入输出语句等,为了处理稍复杂的问题,后期在三种结构的基础上学习函数、数组、指针、结构体、文件等. C语言课程的主要内容如图1所示,其中箭头表示授课的先后顺序. 它与数据结构的关系如图2所示.

根据图2可以看出,数据结构课程主要讲授两大类数据在计算机中的存储和实现,即线性数据(线性表、栈、队列、串等)和非线性数据(树、图等)[8-9],主要操作是数据的增、删、改、查. 其中线性存储结构是非线性结构的基础,而线性结构的基础又是C语言中的数组、指针、函数和结构体,故而应在学习C语言的过程中把这几部分知识重点讲解,让学生多加练习,学精学透. 如此,学生们在进行数据结构的学习后,会顺利过度,无缝衔接,也会进一步提升学生们的学习兴趣和解决实际问题的能力. 因此,教师在讲授C语言的过程中,必须有所侧重、有的放矢,为学习数据结构做好准备.

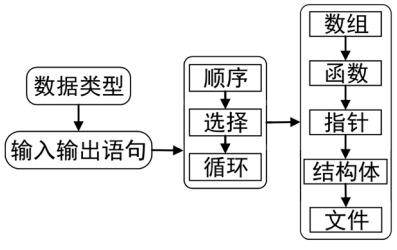

图1 C程序设计主要内容

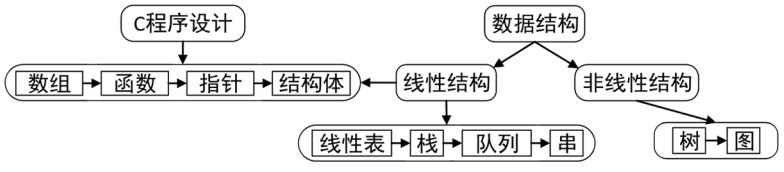

图2 C语言相关内容与数据结构之间的关系

2 C语言教学中存在的问题

从如何学好数据结构这门课程来看,目前的C语言教学过程中存在如下主要问题:

(1) 总体来看,过于注重语法体系,实验课内容浅显. C语言是学生接触到的第一门程序设计语言,而学生的学习精力大多用在简单程序的编写上. 教学过程中发现,在循环结构学习之前,绝大部分同学都能跟上节奏,然而当循环、函数、数组、指针、结构体等内容展开后,开始陆续有学生掉队. 这说明我们在平时的授课和实验过程中,对相对难点投入的时间和深度不够,使得学生在处理复杂问题时缺乏经验和能力. 而数据结构的学习又恰恰以数组、函数、指针和结构体为基础,这个环节的薄弱导致了学习数据结构不能得心应手.

（2）从具体内容来看，存在如下问题：

①从 C 语言到数据结构命名形式的跳变. 在 C 语言的教学过程，虽然讲授了变量、函数名等的命名规则，如做到见名知意、驼峰命名法等，但因为教材中的例题简单，采用的命名法也相对简易，如变量名和数组名用 a、b、c 等，函数名使用 f1、f2 等. 虽然对初学者来说，这样的命名简单、易写，但是过渡到数据结构中，却非常不适应，仿佛学的是两种语言. 如数据结构中初始化线性表的函数名 InitList_Sq，建立链表的函数名为 CreateList_L，图的顶点数目常量 MAX_VERTEX_NUM 等. 在数据结构中的变量和函数名虽然稍显复杂，但真正做到了见名知意，增加了程序的可读性.

②指针内容讲授过于泛化. 众所周知，指针是 C 语言的核心和精华，指针的使用除了可以使程序简洁、紧凑和高效外，有些情况非用指针不可，比如数据结构中链表、树、图的构建. 在有些无法用值传递完成的问题中，可以用指针来完成，如在子函数中完成两个数的交换. 然而，在 C 语言的教学中，根据教材内容讲授往往浅显且宽泛，并无针对性地重点讲指针的具体应用，给学生的感受是指针可用可不用，指针麻烦难懂，可被替代，完全没有体会出其精髓所在，过度到数据结构时，老师又花费大量时间帮助学生理解指针的用法和妙处，增加了数据结构学习的难度.

③结构体部分的讲解过于简略. 在 C 语言的学习过程中，由于课本例题、习题以及实验内容偏简单，似乎不用太多结构体的知识，加之课时有限，很多老师就把该部分作为略讲和选讲的内容，以至给学生带来结构体不重要的错觉. 然而结构体在数据结构的学习过程中，贯穿始终，有时候还层层嵌套，比如图的邻接表的存储表示[9]，如图 3 所示.

图的实现属于数据结构中偏后部分的知识，但仍有部分同学不能很好地理解这种嵌套的结构体，都是源于 C 语言基础薄弱.

```
//------------图的邻接表存储表示---------
#define MAX_VERTEX_NUM     20
Typedef struct ArcNode{
    int                adjvex;                //该弧所指向的顶点的位置
    struct ArcNode     *nextarc;              //指向下一条弧的指针
    InfoType           *info;                 //该弧相关信息的指针
}ArcNode;

Typedef struct VNode{
    VertexType         data;                  //顶点信息
    ArcNode            *firstarc;             //指向第一条依附该顶点的弧的指针
}VNode, AdjList[MAX_VERTEX_NUM];

Typedef struct {
    AdjList             vertices;
    int                vexnum, arcnum;        //图的当前顶点数和弧数
    int                kind;                  //图的种类标志
}ALGraph;
```

图 3　结构体在数据结构中重要性示例

④算法设计思想薄弱. 在 C 语言的学习过程中，由于内容比较浅显，例题、练习题和实验内容都比较简单，所以同学们算法设计思想淡薄. 虽然学习了流程图，但几乎在整个 C 语言的学习过程中没有用到，致使学生们养成了遇到问题没有过多的思考和设计就直接编码的习惯，结果形成了编了删、删了再编的低效率学习，也使得同学们在学习数据结构的过程中，由于问题的复杂，而变得束手无策，算法设计不成体系.

3　以数据结构为导向开展 C 程序设计

为了解决上述在 C 语言教学中存在的问题，现提出以下对应整改措施.

（1）总体上侧重实际应用. 由于以往 C 语言的教学过程中，程序相对简单，所以学生在编写过程中会存在这样的困惑，编写这样的程序有什么用？有些老师还会花费大量时间去讲解在实际应用中使用率不高

的语法点，导致学生的学习热情不高. 所以，我们应该以解决实际问题为出发点，以问题为导向开展C语言的教学. 比如讲解变量的数据类型时，不用面面俱到，只讲最常用和实用的int、float、double、char的基础知识；讲解自增、自减运算符时，只介绍i++(i--)和++i(--i)的区别即可，其余知识请学生自学；在讲解输入输出语句时，也只讲解最为常用的几对输入输出语句即可，如printf和scanf、putchar和getchar、puts和gets等，在实际应用中再次强化. 这样，我们利用较少的时间学习最实用的这部分基础知识，结余出更多时间去学习后面的循环、数组、函数和指针等难点部分，为数据结构的学习打好基础.

(2) 具体内容来看，措施如下：

①养成良好的编程风格. 既然C语言是学生接触到的第一门编程语言，那它也担负着学生习惯养成的重大使命. 这个习惯主要是指算法设计和编写可读性强的程序. 面对复杂问题时，我们要培养学生化繁为简、有步骤各个击破的思想，即利用函数解决一个个小问题，每个函数只实现一个子功能. 由于函数的高内聚性，设计它就变得容易了.

编写可读性强的程序则要求学生们做到以下几点[10]：a. 注意程序的视觉组织，采用阶梯缩进形式使程序结构清晰明显；b. 变量、函数的命名做到见名知意；c. 给每个函数加序言性注释，重要的变量、语句等加解释性注释.

例如图4所示的代码[11]，虽然功能简单，但可读性强. 这样的编程风格，更加有利于学生们过渡到数据结构的学习中.

```
//以下是根据收入、支出计算盈亏的小程序
#include <math.h>
#include <stdio.h>
void main(void){
    double   income, expenses, savings, deficit, interest;

    printf("Enter your income and expenses:\n");
    scanf("%lf%lf", &income, &expenses);
    printf("\n\n");

    if(income > expenses){
        savings = income- expenses; //计算结余
        printf("\nYou are saving money. Your savings for this month are: $%8.2f\n", savings);
    }
    else{
         deficit = expenses – income;//计算亏空
         printf("\nYou are running a deficit. Your deficit for this month is: $%8.2f\n", deficit);
    }
    interest = (deficit > 0.0) ? (0.05* deficit : (0.0);
    printf("\nThe interest you owe on your debt is $%.2f\n", interest);
}
```

图4 程序可读性举例

②指针部分重点讲解与数据结构结合强的知识点. 指针是C语言中的重点和难点，然而在C语言的教学过程中，过多地注重了指针的语法，而非应用. 在C语言的大部分教材中，有一节详细地讲述了一维数组、二维数组与指针的关系，以及指针数组和数组指针，给学生的体会是指针并无太大用处，是否使用指针都可以实现数组的操作. 为避免学生们对指针认知上的偏差，除了指针的基础知识外，我们应该花大量篇幅去学习必须使用指针的情况、指针与函数的关系、动态内存分配中涉及到的指针知识以及结构体与指针. 站在利用数据结构解决实际问题的角度，指针贯穿始终，从顺序结构的链表到层次结构的树和网状结构的图，没有指针难以实现. 以链表为例，链表节点的定义包括了指针与结构体的关系；链表的创建除了指针与结构体的关系，还包括了指针与动态分配内存的关系；链表的删除包括了指针与函数的关系等. 总之，指针的重要作用无可取代，所以我们应该重点讲授与数据结构存储相关的指针知识点，可适当渗透数据结构相关知识，引起学生兴趣与重视.

③结合数据结构的顺序表学习数组和结构体. 数据结构中的顺序表本质结构是一维数组，但通过动态申请获得空间，利用结构体整合数组，同学们会觉得晦涩难懂，这都源于 C 语言学习过程中老师未曾从数据结构角度进行讲解. 所以，当学生们学完基本的数组基础知识后，可以适当扩充，数组不仅可以静态产生，也可以动态申请，而且动态申请的空间弥补了静态不能动态增长的弊端. 数组的学习也可以引出结构体相关知识，例如有些实际问题不能预估具体数据量，故会定义较大数组存储少于数组长度的数据，使得数组中实际数据个数与该数组息息相关，这样我们就可以把数组和它的实际长度封装在一起，利用结构体实现. 通常情况下，在 C 语言的教材中，动态申请空间的两个函数 malloc 和 remalloc 以及结构体都是放在最后面学的，讲解也比较粗略，学生们也没有应用场景，自然体会不到它们的真正作用.

经过把顺序表和数组学习的结合，让同学们深切感受到了结构体和动态申请空间函数的实际应用，做到了有的放矢，能够学以致用，更能提高学习兴趣. 同时，也可以把书上的例题和习题按动态申请数组空间和利用结构体封装的思想再练习一遍. 这样，学生们学习数据结构时候就会衔接顺利，再学习栈、队列的时候也更得心应手.

④强化算法设计思想. 如果我们在学习和讲授 C 语言的过程中，注重实际应用且执行上述措施，学生潜移默化中就有了算法设计的思想. 在讲解具体题目时，除了自然语言描述算法思想外，教师再配以流程图、伪码等描述算法的工具，学生慢慢就会养成先思考、再编程的习惯.

4　总结

鉴于学生们对学习数据结构的迷茫和畏难心里，我们需要从根本上解决问题，即把其基础——C 语言程序设计学透、学牢. 故本文针对 C 语言教学中存在的主要问题加以分析并给出了相应的解决措施. 此外，本人通过走访、调研学生心声发现，绝大多数学生希望这两门课程由一名教师授课，形成两学期课程的连续性.

本文所阐述的内容不是在 C 程序设计过程中简单地添加数据结构知识，而是以数据结构为导向的深入学习 C 语言知识，并渗透数据结构算法的思想，同时培养学生的逻辑思维能力和创新能力. 故对学生的考核应以实际动手能力为主，合理选用教材. 本人下一步的主要工作就是重新编写融入数据结构思想的 C 程序设计教材.

参考文献：

[1] Algorithms＋Data Structures＝Programs：[M]. Upper Saddle River，NJ：Prentice Hall PTR.
[2] 赵榆琴，杨健，张晓玲，等. 程序设计与数据结构“双向重构”教学衔接法探讨 [J]. 计算机教育，2018(8)：151-155.
[3] 丁智国，彭浩，王峰. 协同创新模式下程序设计综合训练课程改革 [J]. 中国信息技术教育，2022(14)：87-89.
[4] 胡丽娜，褚洪波，孟宪伟. 程序设计类课程的实践与创新能力的培养研究 [J]. 物联网技术，2022，12(9)：144-146.
[5] 宛南，张义，杨利. 基于程序设计能力培养的数据结构课程教学改革与实践 [J]. 黄山学院学报，2020，22(5)：112-114.
[6] 布莱恩. W. 克泥汗，丹尼斯. M. 里奇. C 程序设计语言 [M]. 陈宝文，李志，译. 北京：机械工业出版社，2019.
[7] 谭浩强. C 程序设计教程 [M]. 3 版. 北京：清华大学出版社，2018.
[8] 马克. 艾伦. 维斯. 数据结构与算法分析 C 语言描述 [M]. 冯舜玺，译. 北京：机械工业出版社，2019.
[9] 李筠，姜学军. 数据结构：微课版 [M]. 3 版. 北京：清华大学出版社，2021.
[10] 张海藩，牟永敏. 软件工程导论学习辅导 [M]. 6 版. 北京：清华大学出版社，2013.
[11] TAN H H，D’ORAZIO T B，OR S H，et al. C Programming a Q & A Approach [M]. New York：The McGraw-Hill Companies.

责任编辑　胡杨