Advances in Education 教育进展, 2021, 11(1), 284-288
Published Online January 2021 in Hans. http://www.hanspub.org/journal/ae
https://doi.org/10.12677/ae.2021.111047

Hans汉斯

C程序设计课程教学的探索与实践

张　力，田　琦，范青刚

西安高新技术研究所，陕西　西安
Email: qingyi922@sina.com

收稿日期：2020年12月27日；录用日期：2021年1月20日；发布日期：2021年1月28日

摘　要

“C程序设计”课程是高等学校理工科学生的一门必修课程，这门课的教学目标就是使学生掌握设计程序的思路，学会用C语言编写程序，以解决现实中的实际问题。本文作者根据多年的教学经验，总结了若干行之有效的培养学生编程能力的方法。

关键词

C语言，案例教学法，程序设计

The Exploration and Practice in the Teaching of the “C Programming” Course

Li Zhang, Qi Tian, Qinggang Fan

Xi’an High-Tech Research Institution, Xi’an Shaanxi
Email: qingyi922@sina.com

Received: Dec. 27th, 2020; accepted: Jan. 20th, 2021; published: Jan. 28th, 2021

Abstract

The “C Programming” course is a required course for the Science and Engineering students of colleges and universities, the teaching aims of the course are to teach students to grasp the programming way and learn to program with the C Programming Language to solve practical problems. According to years of teaching experience, the author of this paper proposes effective approaches to cultivate the programming power of students.

文章引用: 张力, 田琦, 范青刚. C 程序设计课程教学的探索与实践[J]. 教育进展, 2021, 11(1): 284-288.
DOI: 10.12677/ae.2021.111047

Keywords

The C Programming Language, Case Teaching Method, Programming Design

1. 引言

C 语言是国内外广泛使用的一种计算机语言。目前，在许多高等院校，C 语言是程序设计类课程的基础，它不仅是计算机专业的一门核心课程，而且也是理工科专业的必修课。学习这门课的目的，就是使学生掌握程序设计的方法，学会使用 C 语言编写程序，以解决现实中的实际问题，也就是说这门课的教学重点是要让学生学会编写程序。所以在教学过程中，教师不仅要使学生掌握 C 语言中的复杂概念和规则，重点还要培养学生的编程能力。本文作者多年来一直从事这门课程的教学工作，对课程的内容有着较深的理解，并且积累了大量的教学经验，下面介绍一下本文作者从多年的教学工作中总结出的几种行之有效的在教学中培养学生编程能力的方法。

2. 采用案例教学法[1]讲解经典案例

在扎实掌握概念、规则的基础上，采用案例教学法，尽可能多的讲解一些典型的案例。学生初学 C 语言，即使掌握了一定的概念和语法规则以后，对于编写程序也是感觉毫无思路，无法下手，这时就需要通过讲解一些典型的案例帮助学生打开思路，教会学生面对一个实际问题时，如何分析问题、如何编程解决问题。培养起学生解决问题的思维方法。具体作法是：在讲解一个问题时，(1) 首先给出这个问题的解决方法和解决步骤，也就是解题思路，(2) 然后再根据这个思路画出框图，(3) 最后再引导学生如何根据框图，利用自己所学的知识一步步编写程序；程序完成后，再进行程序的分析。这样就可以逐步培养起学生解决问题的基本思路。

例如，在讲解循环结构这部分内容时，要讲解一个经典案例[2]，求 1 + 2 + 3 + … + 100，即 $\sum_{n=1}^{100} n$。按照上述作法：

(一) 首先给出解题思路：在处理这个问题时，先分析此题的特点：(1) 这是一个累加的问题，需要重复进行加法运算，将 100 个数相加，像这样的重复性的操作。显然要用循环结构来实现。重复执行循环体 100 次，实现 100 个数的累加。(2) 分析每次所加的数有无规律？发现每次累加的数是有规律的，后一个数是前一个数加 1。因此不需要每次都从键盘临时输入数据，只需在加完上一个数 i 后，使 i 加 1 就可得到下一个数。

(二) 根据解题思路画出此题的流程图，如图 1 所示。

(三) 根据流程图写出程序。

```
#include <stdio.h>
main( )
{
int i=1,sum=0;
while(i<=100)
```

```
{
sum=sum_i;
i++;
}
printf("sum=%d\n",sum);
}
```

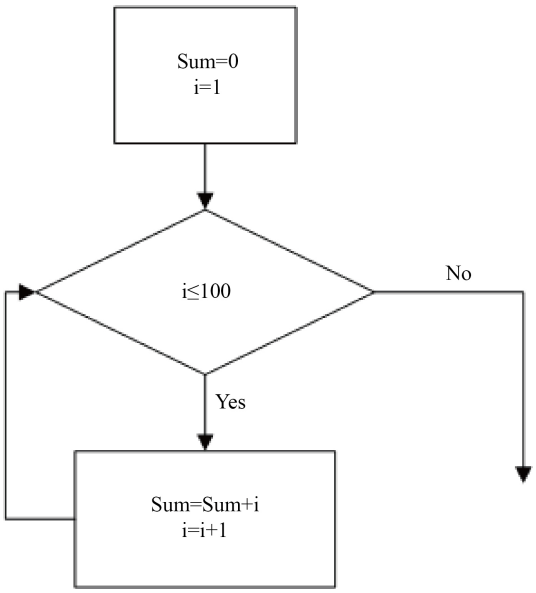

Figure 1. Algorithm flow chart
图 1. 算法流程图

再进行程序的分析，讲出需要注意的问题：(1) 循环体如果包含一个以上的语句，应该用花括弧括起来，作为复合语句出现；(2) 注意累加操作的实现，累加和的初值应该为 0；(3) 注意循环结束条件的设置。

3. 理论讲授和程序演示相结合

理论讲授和程序演示相结合，可以帮助学生更好地理解和掌握所学知识，提高学生的学习兴趣和积极性，活跃课堂气氛。一般来说，在课堂上讲解完一个程序后，学生对于程序的理解还处于比较肤浅的水平，甚至对程序的运行结果还会得出错误的答案，这个时候，现场演示一下程序的运行，通过运行，讲解程序中所包含的理论和知识，会极大地提高学生的学习兴趣，帮助学生深刻理解掌握所讲授的内容，活跃课堂气氛。在演示程序的过程中，还可以人为地在程序中制造一些学生易犯的错误，演示并讲解这种错误带来的后果，从而加深学生对所学内容的理解和掌握。例如，在学指针变量的定义和引用这部分内容时，演示这样一个非法使用指针的程序[3]：

```
#include<stdio.h>
main( )
{   int   *p,a=1;
    float *q;
    *p=5;
```

```
    q=&a;
    printf("%x,%d,%d,%f\n",p,*p,*q,*q);
}
```

运行结果如下：

125，5，0，−NAN (125 是按十六进制形式输出的 p 中值)

通过这个错误的结果，就要向学生说明以下几点：(1) 指向整型变量的指针变量 p 只是被声明，并没有赋初值，因此指针变量 p 是没有值的。这样程序运行完后，p 中存放的值为 125 (十六进制)，这说明 p 指向地址为 125 的存储单元。假如该存储单元中原来存放内存中的有用数据，由于*p = 5 用 5 覆盖该存储单元原来的值，所以得出了错误的结果，有时甚至会导致严重后果。(2) 因为 a 是 int 型，而 q 只能指向 float 型变量，因此 q=&a 是错误的，以%d 和%f 两种形式输出*q 的值时，输出结果都不与 a 的值 1 相同。(3) 编译本程序时，有 4 个警告错误。其中 3 个错误是由于 p 没有确定的指向造成的，而另一个错误是因 q 指向类型不一致的变量引起的。通过这个例子，就可以使学生深刻理解并掌握指针的概念，达到良好的教学效果。再比如，在演示程序时，还可以在程序中制造一些语法错误，通过编译可以显示这些错误，然后引导学生如何根据提示信息找到这些错误，然后对照所学的规则规定改正这些错误，这样逐步提高学生调试程序的能力。

4. 循序渐进的教学方法

采取循序渐进、由简入繁、由易入难的方法，逐步培养学生的编程能力。在学生学习了一些典型例题和解题的方法思路以后，就可以开始要求他们编写程序了。开始让他们从最简单的程序设计入手，熟悉 C 程序的结构和组成，掌握最基本的编程方法，随着教学内容的深入，配合教学内容，逐步增加题目的难度和复杂度，这个阶段就逐步要求学生严格按照编程的三个步骤编写程序，即设计解题步骤 → 画框图 → 编程。又比如，学习函数以后，就要培养学生使用函数编写程序的思路，通过典型案例的讲解，总结出利用函数编写较大程序的方法[3]：(1) 根据题目要求，将程序按功能分成若干个相对独立的模块，每个模块用一个函数实现；(2) 设计主框架，编写主函数，主函数主要做三件事：输入，调用子函数，输出结果。(3) 编写被调函数。首先确定函数首部，这是被调函数和主调函数的接口，也是本函数的入口(接收数据)；然后编写函数体，实现相应的功能。(4) main 函数中的被调函数也可以调用其它函数，因此对于每个被调函数，都要用类似于 main 函数的编写方法逐步细化。这样循序渐进，逐步培养学生的编程能力，同时也可以帮助学生克服畏难心理，在编程的过程中不断获得成就感，从而增强习的兴趣和自信心。

5. 抓好上机实践环节

在学生掌握了一定的理论知识和编程方法后，有一个至关重要的环节就是上机实践。学生在掌握了理论知识后，不但理解不深，而且根本不会编写程序，只有通过上机实践才能加深对理论知识的理解，提高编程能力。而且在程序设计过程中，常常会由于一些细节而导致程序运行错误，有许多编程的细节和技巧，光靠课堂上的讲授是学不到的，必须靠自己多编程，多上机，在上机实践中不断发现问题，积累经验，提高水平。通过上机实践，可以使得学生更好地掌握 C 语言的理论和编程方法，提高学生的动手能力，调试程序的能力和解决问题的能力。开始时，实践的题目不易求多而要求精，每编一个程序，就要求学生通过这个程序更加深刻地理解掌握课堂上所学的 C 语言的概念和规则，以及如何运用这些规则概念编写程序，解决实际问题；上机实践课上，要指导学生在程序出现错误时如何发现问题，调试程序，同时要求学生从程序运行失败中总结经验，踏踏实实地将这个程序调试运行通过，通过编写调试程序，逐步培养起学生的编程能力，这样循序渐进，熟能生巧，上机实践能力切实得到提高。而且实践证

明，通过上机实践，还可以极大提高学生对本门课的学习兴趣和热情。

6. 结语

“C 程序设计”是学生既感兴趣又觉得难以掌握的一门课程，难点就在于编写程序。针对这个难点，在教学中，需要采取灵活多样，循序渐进，理论讲授和上机实践相结合的教学方法。本文作者根据多年教学实践总结出的上述几种方法在教学中取得了良好的教学效果，激发了学生学习这门课程的积极性和兴趣，编程能力得到了极大的提高。

参考文献

[1] 龚绍文. 大学青年教师教学入门[M]. 北京: 北京理工大学出版社, 2008.

[2] 谭浩强. C 程序设计[M]. 北京: 清华大学出版社, 2012.

[3] 崔武子, 等. C 程序设计教程[M]. 北京: 清华大学出版社. 2015.