Creative Education Studies 创新教育研究, 2019, 7(4), 498-503
Published Online August 2019 in Hans. http://www.hanspub.org/journal/ces
https://doi.org/10.12677/ces.2019.74085

Hans汉斯

The Case Design Method in C Language Programming Teaching

Xue Feng

School of Computer, Beijing Information Science & Technology University, Beijing
Email: fengxue@bistu.edu.cn

Received: Aug. 1st, 2019; accepted: Aug. 15th, 2019; published: Aug. 22nd, 2019

Abstract

C Language Programming Course is an introductory course for computer majors. How to enable students to comprehensively understand all knowledge points and improve their programming ability effectively is the key problem to be explored in this course. This paper designs a set of examples to link up the knowledge points of the course from shallow to deep and step by step, so that students can really feel the gradual application of these knowledge points in the examples. Compared with the knowledge system of this course, the examples designed in this paper basically cover all the core knowledge points. Through the practice test of classroom, the application of this set of examples help to improve the teaching effect.

Keywords

C Language, Examples, Knowledge System

C语言程序设计教学中的实例设计方法

冯　雪

北京信息科技大学计算机学院，北京
Email: fengxue@bistu.edu.cn

收稿日期：2019年8月1日；录用日期：2019年8月15日；发布日期：2019年8月22日

摘　要

C语言程序设计课程是计算机专业的程序入门课，如何让学生融会贯通地理解各个知识点，并有效提高编程能力，是该课程需要探索的关键问题。本文设计了一套实例，将课程的各个知识点由浅入深地、循序渐进地衔接起来，让学生可以真正感受到这些知识点在实例中的逐步应用。通过与课程知识体系对比，本文设计的实例基本覆盖了所有核心知识点。通过课堂实践检验，该套实例的应用有助于提升教学效果。

文章引用: 冯雪. C 语言程序设计教学中的实例设计方法[J]. 创新教育研究, 2019, 7(4): 498-503.
DOI: 10.12677/ces.2019.74085

关键词

C语言，实例，知识体系

1. 引言

C 语言程序设计课程是计算机专业的基础课程之一，也是我院开设的八大核心课程之一，是计算机专业学生学习的第一门编程类课程。在谭浩强老师出版的《C 程序设计(第五版)》教材[1]中，主要介绍了顺序结构、选择结构、循环结构、数组、函数、指针和文件等方面的内容[1]。针对各个知识点，教材中设计了一些例题来帮助学生理解。但是，如何对这些内容融会贯通，将各个知识点有效结合，并能够提高学生的编程能力，是本课程一直探索的关键。

经过多年的实践验证，好的实例是学生能够更好地学习 C 语言课程的关键。因此，如何设计一套实例，将课程的各个知识点由浅入深地衔接起来，让学生可以真正感受到这些知识点在实例中的逐步应用，是本文的主要研究内容。通过课堂实践检验，该实例的应用有助于提升教学效果。文献[2]也提出一套案例教学方法，但本文中的实例设计方法更强调知识点逐步结合的过程，并且可以覆盖更广泛的知识点。

2. C 语言课程实例设计方法

2.1. C 语言课程知识体系

本课程的知识点主要包括顺序结构的程序设计、选择结构的程序设计、循环结构的程序设计、函数、指针、结构体、文件等方面的知识，目标是使学生能够熟练地阅读和运用结构化程序设计方法来设计、编写、调试和运行 C 语言程序。具体知识点描述如表 1 所示，其中 3~10 是讲授的重点。

Table 1. The knowledge system of C language programming course
表 1. C 语言程序设计课程知识体系

序号	教学内容	基本要求
1	C 语言概述	了解 C 语言的历史和特点，初步掌握 C 语言程序的构成。
2	算法	了解算法的基本概念和表示方法；掌握什么是结构化程序设计方法
3	顺序结构程序设计	**重点掌握**编写简单 C 程序的方法，熟悉不同数据类型的输入和输出。
4	选择结构程序设计	掌握关系运算符、逻辑运算符及其表达式的使用；**重点掌握**选择结构的程序设计思想。
5	循环结构程序设计	**重点掌握**循环结构的程序设计思想。
6	利用数组处理批量数据	**重点掌握**一维数组、二维数组的定义和引用方法，能够编写使用了数组的程序。
7	用函数实现模块化程序设计	熟练掌握函数的定义方法和调用方法，**重点掌握**基于函数的模块化程序设计方法。
8	指针的定义与使用	掌握指针的基本概念和运算以及空间的动态分配思想； **重点掌握**指针作为函数参数的使用方法以及利用指针操作数组的方法。
9	结构体的定义与使用	**重点掌握**结构体的定义、使用以及结构体指针和链表的使用方法。
10	文件的使用	**重点掌握**文件的打开、关闭和读写等方法。

2.2. C 语言课程实例设计方法

本文提示的实例设计方法，秉持着两点原则：1) 强调知识点逐步结合的过程，从简单实例开始，随着学生不断深入的学习过程，逐步扩展实例；2) 尽可能覆盖更多的知识点。实例设计过程如下：

1) 设计一个实例，输入一个年份和月份，显示该月有多少天。这个实例主要用到选择结构，1，3，5，7，8，10，12 这些月份是 31 天，4，6，9，11 是 30 天，2 月份正常是 28 天，若是闰年，则是 29 天。该实例的代码如下：

```
#include<stdio.h>
int main()
{
    int year, month, days;
    scanf("%d %d", &year, &month);
    if(month==1||3||5||7||8||10||12)
        days=31;
    else if(month==4||6||9||11)
        days=30;
    else if((year%4==0&&year%100!=0)||(year%400==0))
        days=29;
    else
        days=28;
    printf("%d\n", days);
    return 0;
}
```

2) 在该实例的基础上进行扩展，输入一个年、月、日，计算该天是一年中的第多少天，这样就融入了数组和循环结构两项知识点。设计一个数组 m，里面存放 12 个月的天数。通过循环来计算该天是一年中的第多少天。扩展后的代码如下：

```
#include<stdio.h>
int main()
{
    int year, month, day,days=0,i;
    int m[12]={31,28,31,30,31,30,31,31,30,31,30,31}; //数组的定义
    scanf("%d %d %d", &year, &month, &day);
    if((year%4==0&&year%100!=0)||(year%400==0))
        m[1]=29;
    for(i=0;i<month-1;i++)   //使用循环和数组
        days=days+m[i];
    days=days+day;
    printf("%d\n", days);
    return 0;
}
```

3) 继续扩展该实例，引入基于函数的模块化程序设计方法，将闰年的判定方法写成一个函数，天数的计算则用另一个函数来描述。扩展后的代码如下：

```
#include<stdio.h>
int main()
{
    int year, month, day,days;
    scanf("%d %d %d", &year, &month, &day);
    days=compute_days(year,month,day);
    printf("%d\n", days);
    return 0;
}
int judge_leap_year(int year)     //闰年判定函数
{
    if((year%4==0&&year%100!=0)||(year%400==0))
        return 1;
    else
        return 0;
}
int compute_days(int year, int month, int day)     //天数计算函数
{
    int m[12]={31,28,31,30,31,30,31,31,30,31,30,31},i,days=0;
    if (judge_leap_year(year)==1)
        m[1]=29;
    for(i=0;i<month-1;i++)
        days=days+m[i];
    days=days+day;
    return days;
}
```

4) 继续扩展该实例，融入指针知识点，通过指针变量 pt 来指向数组 m 的首元素，并使用指针来操作数组元素，扩展后的代码如下：

```
#include<stdio.h>
int main()
{
    int year, month, day,days;
    scanf("%d %d %d", &year, &month, &day);
    days=compute_days(year,month,day);
    printf("%d\n", days);
    return 0;
}
```

```
int judge_leap_year(int year)     //闰年判定函数
{
    if((year%4==0&&year%100!=0)||(year%400==0))
        return 1;
    else
        return 0;
}
int compute_days(int year, int month, int day)     //天数计算函数
{
    int m[12]={31,28,31,30,31,30,31,31,30,31,30,31},i,days=0;
    /* 下面将使用指针来操作数组 m 中的元素 */
    int *pt=m;
    if (judge_leap_year(year)==1)
        *(pt+1)=29;
    for(i=0;i<month-1;i++)
        days=days+*(pt+i);
    days=days+day;
    return days;
}
```

5) 继续扩展该实例，融入结构体知识点，将年、月、日封装成一个结构体类型。在上述实例基础上，定义一个叫做 date 的结构体，该结构中包含年、月、日三个成员，扩展后的代码如下：

```
#include<stdio.h>
/* 结构体定义 */
struct date
{
    int year;
    int month;
    int day;
};
int main()
{
    date d1;   //结构体变量定义
    scanf("%d %d %d", &d1.year, &d1.month, &d1.day); //给结构体成员输入值
    days=compute_days(d1); //函数实参为结构体变量
    printf("%d\n", days);
    return 0;
}
int judge_leap_year(int year)     //闰年判定函数
{
```

```
    if((year%4==0&&year%100!=0)||(year%400==0))
        return 1;
    else
        return 0;
}
int compute_days(date d)    //天数计算函数，其中形参为结构体变量
{
    int m[12]={31,28,31,30,31,30,31,31,30,31,30,31},i,days=0;
    int *pt=m;
    if (judge_leap_year(d.year)==1)
        *(pt+1)=29;
    for(i=0;i<d.month-1;i++)
        days=days+*(pt+i);
    days=days+d.day;
    return days;
}
```

6) 继续扩展该实例，融入文件知识点。在上述实例的基础上，将一组日期信息存入到文件中，并从文件中读取日期信息，并根据日期信息，计算该天是一年中的第多少天。其中，需要增加日期存储到文件的函数和从文件读取日期的函数，并设计菜单界面，让用户选择相应的操作(1、存储日期；2、读取日期；3、天数计算)。扩展后的代码，本文不详细给出。从实例设计角度，扩展后的实例融入了选择结构、循环结构、数组、函数、指针、结构体和文件等知识点。通过与 2.1 节中的课程知识体系对比，该实例基本覆盖了所有核心知识点。

经过几轮授课，在针对各个知识点学习的基础上，结合本文提出的循序渐进扩展式实例，可以让学生感受到各个知识点的综合使用方法，更重要的是，可以让学生感受到这些知识点逐步结合的过程。实际授课表明，这种循序渐进扩展式实例更有助于学生去学习 C 语言，学习效果更佳。

3. 结束语

好的实例是学生能够更好地学习 C 语言课程的关键，2017 年起，在北京信息科技大学计算机学院的学生中应用本文所设计的一套实例，结合教材的使用，可以进一步提升学生的学习效果，加强学生对知识点的理解和融会贯通。今后，将进一步提升实例的综合性、趣味性，从而提高学生的学习兴趣。

基金项目

北京信息科技大学教改项目(2018JGYB13)。

参考文献

[1] 谭浩强. C 程序设计[M]. 第五版. 北京: 清华大学出版社, 2017.

[2] 杨帅, 薛永江. 以培养编程能力为导向的 C 程序课程设计教学[J]. 教育进展, 2018, 8(4): 407-411.

知网检索的两种方式：

1. 打开知网首页：http://cnki.net/，点击页面中“外文资源总库 CNKI SCHOLAR”，跳转至：http://scholar.cnki.net/new，搜索框内直接输入文章标题，即可查询；
   或点击“高级检索”，下拉列表框选择：[ISSN]，输入期刊 ISSN：2331-799X，即可查询。
2. 通过知网首页 http://cnki.net/顶部“旧版入口”进入知网旧版：http://www.cnki.net/old/，左侧选择“国际文献总库”进入，搜索框直接输入文章标题，即可查询。

投稿请点击：http://www.hanspub.org/Submission.aspx
期刊邮箱：ces@hanspub.org