



21世纪普通高等教育应用型规划教材 · 计算机系列

C/C++

程序设计 综合实践教程

主 编 白剑宇

副主编 唐云廷 徐 斌 刘 锋

C/C++ CHENGXU SHEJI
ZONGHE SHIJIAN
JIAOCHENG



北京邮电大学出版社
www.buptpress.com

21世纪普通高等教育应用型规划教材·计算机系列

C/C++程序设计综合实践教程

主 编 白剑宇

副主编 唐云廷 徐 斌 刘 锋

北京邮电大学出版社

·北京·

内 容 简 介

本书参照项目开发的方法,通过一个具体的学生成绩管理系统实例,介绍了应用程序的总体设计、界面设计和模块设计的一般方法。在整个学习和开发过程中,锻炼和提高学生综合运用 C 语言知识的能力和实际编程能力,帮助学生学习和掌握实际系统开发的一般方法和步骤,并使能够逐步从简单的程序设计过渡到小型系统的开发。本书不仅介绍了用经典的 C 语言开发整个系统的方法,同时介绍了 MFC 程序设计,使学生能够熟悉 Visual C++ 集成开发环境,学习 Windows 编程,激发学生的学习兴趣。

本书可作为高等院校计算机专业、软件工程专业及其他专业“程序设计”课程的课程设计教材和综合训练教材,也可作为大专、职业教育计算机及相关专业学生的教材,还可以作为学习 MFC 的入门教材。

图书在版编目(CIP)数据

C/C++程序设计综合实践教程/白剑宇主编.--北京:北京邮电大学出版社,2011.1

ISBN 978-7-5635-2540-9

I. ①C… II. ①白… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 254809 号

书 名: C/C++程序设计综合实践教程

主 编: 白剑宇

责任编辑: 付兆华

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm×1 092 mm 1/16

印 张: 12.25

字 数: 302 千字

印 数: 1—3 000 册

版 次: 2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷



ISBN 978-7-5635-2540-9

定 价: 23.00

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

C 语言从 1973 年诞生到现在已经快 40 年了,在现在的程序设计教学中,我们经常会碰到这样的疑问,现在还有必要学习 C 语言吗?与其他程序设计语言相比,C 语言无论从开发界面还是学习难度上,都使初学者望而却步,国内也有一些院校取消了 C 语言的课程,然而实际情况如何呢?

据国外 TIOBE 程序设计语言排行榜数据显示,从 2000 年至今,C 语言长期稳居排行榜第二名,其中在 2005 年前后位列第一。这说明 C 语言仍然是目前应用得最广泛的语言之一,仅次于 Java,远高于其他高级语言。从该排行榜还可以看出 2010 年以后 C 语言有超越 Java 的趋势。

其实,C 语言虽然缺乏一些目前时髦的概念,但一旦涉及与底层硬件相关的开发,C 语言与汇编语言仍然是首选的开发工具。尤其是目前大量的与嵌入式系统相关的开发,仍然需要依赖 C 语言。

C 的表达能力强、效率高及对机器的最终控制等特点使得 C 语言具有强大生命力和广阔的应用前景。学习和掌握 C 语言,仍然是当前计算机及相关专业学生所必须具备的基本技能。

当然,也应该正视目前 C 语言教学中面临的一些问题。首先 C 语言中有些知识点较难理解,如指针的理解与应用、函数的构建与调用,以及程序的总体设计等。其次是 C 语言应用程序的界面单调、枯燥,降低了学生的学习兴趣。最后一点是 C 语言的学习对以后的设计、工作联系不大等。

为此,本书努力在 C 语言教学上进行了一些尝试,帮助学生克服学习上的一些难点,激发学生学习 C 语言的兴趣。本书通过一个项目案例,引入软件工程的初步思想,引导学生完整地经历需求分析、总体设计、模块设计、编程实现和测试这些软件工程中的关键步骤,帮助学生建立一个程序设计的大局观,从而扎实地掌握 C 语言的基本编程技能。

此外,本书在满足 C 语言基本结构下,适当添加 MFC GUI 内容,使项目能够以图形化界面的方式呈现,提高学生的学习兴趣,同时让学生开始接触 Windows 编程,与以后的工作、学习衔接。

本书共分 9 章。第 1 章介绍了用 C 语言开发系统的方法和步骤;第 2 章介绍了如何使用 Visual C++ 6.0 和 MFC AppWizard 来编写 Windows 应用程序;第 3~8 章分别介绍了学生成绩管理系统各个功能模块的设计和开发;第 9 章介绍了把应用程序制作成安装程序包的方法。

本书的主要特点如下。

① 实用性强。本书采用项目教学法,围绕一个实际系统的开发展开,并按照实际的项目开发方法和步骤,帮助学生从简单的程序设计过渡到小型系统的开发。



② 专业性强。本书巩固了在“C 程序设计”课程中学过的指针、结构、函数、文件、链表等知识点,并加以灵活运用,使学生具备一定的的高级语言程序设计能力,并为以后的专业课程学习打下良好的基础。

③ 趣味性强。介绍 MFC 程序设计,使学生能够熟悉 Visual C++ 集成开发环境,学习 Windows 编程,良好的界面和简单的操作方法可以激发学生的学习兴趣 and 积极性,并让学生产生成就感,提高自信心。

此外,本书传授知识与传授方法并重,使学生掌握典型问题的分析、程序设计、调试和运行等方法。在熟悉程序设计过程的同时,了解和掌握一些程序设计常用技巧,对大型程序的开发有个初步的了解。

本书是在 C 语言教学上的一种尝试,在编写过程中,由于作者水平有限,书中难免有疏漏之处,恳请广大读者批评指正。

作者

目 录

第 1 章 基于 C 语言的系统开发	1
1.1 系统和开发过程介绍	1
1.1.1 系统设计要求	1
1.1.2 系统功能设计	1
1.1.3 总体设计	2
1.2 系统设计	4
1.2.1 student.h 头文件	4
1.2.2 student.cpp 文件	6
1.2.3 input_disp.cpp 文件	10
1.2.4 delete_record.cpp 文件	13
1.2.5 find_record.cpp 文件	17
1.2.6 save_load.cpp 文件	19
1.2.7 sort.cpp 文件	22
1.3 系统合成与运行	24
1.3.1 多文件程序的编译	24
1.3.2 多文件程序的连接	27
1.3.3 多文件程序的运行	27
1.4 系统测试	27
1.4.1 菜单项、空链表和空文件测试	27
1.4.2 增加记录和显示记录测试	28
1.4.3 保存记录和读取记录测试	29
1.4.4 查询记录和删除记录测试	30
1.4.5 记录排序测试	33
1.4.6 新建班级测试	35
第 2 章 基于 MFC 的系统开发	36
2.1 Windows 编程基础	36
2.1.1 类和对象	36
2.1.2 窗口	37
2.1.3 句柄(Handle)	37
2.1.4 消息与事件驱动	37
2.1.5 工程与资源	38



2.2 Visual C++ 6.0 简介	39
2.2.1 Visual C++ 6.0 工作界面	39
2.2.2 Windows 程序设计	43
2.2.3 使用 MFC AppWizard	43
2.3 基于 MFC 的学生信息管理系统开发	44
2.3.1 应用程序功能与界面	44
2.3.2 创建一个新工程	46
2.3.3 关闭工程和打开一个工程	51
2.4 程序框架和执行流程	51
2.4.1 程序结构	52
2.4.2 代码解释	53
2.4.3 程序执行流程	59
第3章 数据文件的建立与打开	60
3.1 建立数据文件	60
3.1.1 界面设计	60
3.1.2 添加变量	65
3.1.3 添加函数	68
3.1.4 添加函数代码	71
3.1.5 运行工程	75
3.2 打开数据文件	77
3.2.1 添加按钮	77
3.2.2 添加消息响应函数	78
3.2.3 添加代码	80
3.2.4 运行工程	80
3.3 显示数据文件中的记录	81
3.3.1 添加列表控制框	81
3.3.2 定义记录数据类型	86
3.3.3 修改“打开班级”按钮响应函数	88
3.3.4 运行工程	91
第4章 添加与保存学生信息	93
4.1 界面设计	93
4.1.1 数据输入窗口界面设计	93
4.1.2 按钮设计	99
4.2 添加响应函数	101
4.2.1 自定义成员函数	101
4.2.2 为按钮“添加数据”添加消息响应函数	103
4.2.3 为按钮“保存”添加消息响应函数	106



4.3 运行工程	109
4.3.1 添加数据	109
4.3.2 保存数据	111
第5章 修改学生信息	113
5.1 功能需求与模块设计	113
5.1.1 功能需求分析	113
5.1.2 模块设计	113
5.2 界面设计	113
5.3 功能实现	114
5.3.1 操作流程	114
5.3.2 代码实现与解释	114
5.4 运行工程	122
第6章 删除学生信息	125
6.1 功能需求与模块设计	125
6.1.1 功能需求分析	125
6.1.2 子模块设计	125
6.2 界面设计	125
6.3 功能实现	126
6.3.1 操作流程	126
6.3.2 代码实现与解释	126
6.4 运行工程	128
第7章 查询学生信息	130
7.1 功能需求	130
7.2 界面设计	130
7.3 功能实现	131
7.3.1 添加变量	131
7.3.2 代码实现与解释	132
7.4 系统运行	136
第8章 排序	141
8.1 系统需求	141
8.2 界面设计	141
8.3 功能实现	142
8.3.1 代码实现与解释	142
8.3.2 改进	148
8.4 系统运行	151



8.5 添加“退出”按钮	154
第9章 应用程序发布	156
9.1 打包发布前的准备	156
9.1.1 可执行文件	156
9.1.2 自定义版本配置信息	158
9.2 使用 InstallShield for Visual C++ 工具打包发布	160
9.2.1 InstallShield for Visual C++ 6.0 的安装	160
9.2.2 应用程序打包	160
参考文献	185

第 1 章 基于 C 语言的系统开发

本章围绕一个实际的学生信息管理系统的开发,参照项目开发的方法,使学生通过本章的学习,能从简单的程序设计过渡到小型系统的开发;掌握典型问题的分析和程序设计、调试和运行等方法;培养学生分析问题、编程等动手能力,使学生掌握程序设计过程和常用技巧,并了解系统开发的一般方法和步骤;巩固在 C 语言中学过的指针、结构、函数、文件、链表等知识点,并能加以灵活运用。

1.1 系统和开发过程介绍

1.1.1 系统设计要求

学生信息管理系统的界面如图 1-1 所示。这是一个简单的菜单选择界面,从图中可以看出学生信息管理系统应具备增加记录、显示记录、删除记录、查询记录和记录排序等基本功能。

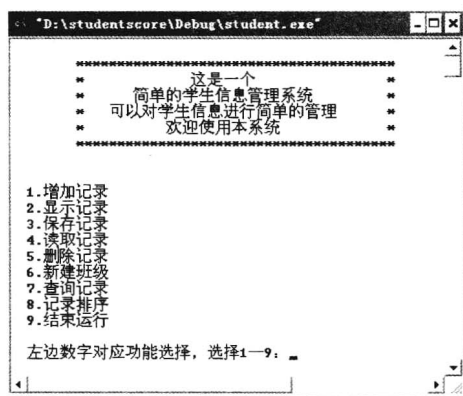


图 1-1 学生信息管理系统界面

学生信息包含学号、姓名、性别、班级、各课成绩、总分和平均分等,可以定义一种新的结构类型,并运用链表来组织数据。

开发本系统将学习和运用指针、结构、函数、文件、链表、宏定义和动态开辟内存等知识点,同时学习和运用数据查询、删除和排序等基本算法。

1.1.2 系统功能设计

系统要求实现的功能很多,学生信息管理系统一般应具备的基本功能如下。

(1) 建立数据文件

- ① 使用默认文件名或指定文件名存储文件;
- ② 如果数据文件存在,则在其后追加记录。



(2) 文件的存取和显示

- ① 可以单独存取文件;
- ② 可以随时显示内存中的全部记录;
- ③ 可以直接存取默认文件或指定文件。

(3) 删除记录

- ① 可以按“姓名”或“学号”方式删除记录并更新链表;
- ② 显示删除记录的信息;
- ③ 显示不能完成记录删除的信息。

(4) 查询记录

- ① 可以按“姓名”或“学号”方式查询记录;
- ② 显示查询记录的信息;
- ③ 显示不能完成记录查询的信息。

(5) 记录排序

- ① 可以按“总分”、“平均分”或“学号”升序或降序排序记录;
- ② 显示排序后的记录。

1.1.3 总体设计

通过简单地对学生信息管理系统进行需求分析,把本系统划分为几个文件,在每个文件内实现若干个能实现具体功能的函数。

1. 数据结构

本系统的数据采用链表来组织。链表的节点设计为结构体,一个结构体(节点)存放一条学生记录。

学生记录结构的定义如下。

```
typedef struct studentscore {  
    char number[20];           //学号  
    char name[20];             //姓名  
    char sex[4];               //性别  
    char classno[10];          //班级  
    double score[NUM_SUBJECT]; //成绩,NUM_SUBJECT 为整型常量,表示课程数  
    double sum;                //总分  
    double average;            //平均分  
    struct studentscore * next; //指针域  
}STUD;
```

这里自定义了一种新的结构类型 STUD,用它可以定义 struct studentscore 类型的变量、数组和指针等。

2. 宏定义

采用链表要比采用结构数组节省空间。使用中根据需要随时动态开辟内存空间,因此动态开辟内存的操作非常频繁,为此,定义带参数的宏定义 ASK,专门用于申请动态内存。

```
#define ASK(p) do{\n    p = (STUD *)malloc(sizeof(STUD));\n
```



```
if(p == NULL) { printf("memory fail"); exit(-1);} \
}while(0)
```

其中行末的符号“\”为宏定义的续行符。该宏定义展开后,先在内存中开辟 struct studentscore 类型结构所需的内存,然后判断动态内存开辟是否成功,如内存开辟成功,则返回该段内存的首地址给参数 p;否则提示内存出错,退出程序。

3. 总体框架和函数调用关系

学生信息管理系统的总体框架可由图 1-2 表示,同时图中表明了各函数间的调用关系。

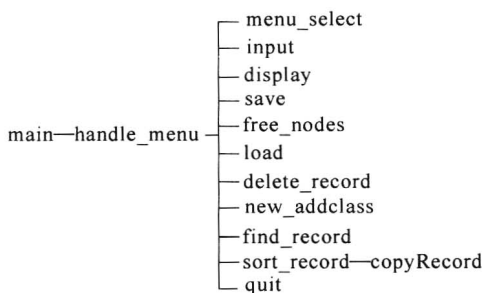


图 1-2 学生信息管理系统总体框架

4. 文件和函数的组成

组成本系统的各个文件和每个文件定义的函数等如表 1-1 所示。

从表 1-1 中可知,学生信息管理系统由 6 个 C 源文件和 1 个头文件组成,共有自定义函数 13 个。每个文件表示一个系统模块,文件中定义的函数就是子模块。

表 1-1 文件及函数组成

源文件	定义函数等	功能	函数原型
student. cpp	main	主函数	void main();
	menu_select	菜单选择	int menu_select(void);
	handle_menu	菜单处理	void handle_menu(STUD *);
	free_nodes	释放内存空间	void free_nodes(STUD *);
	quit	结束运行	void quit(STUD *);
	new_addclass	新建班级	STUD * new_addclass(STUD *);
input_disp. cpp	input	输入信息	STUD * input(STUD *);
	display	显示信息	void display(STUD *);
delete_record. cpp	delete_record	删除信息	STUD * delete_record(STUD *);
find_record. cpp	find_record	查询信息	void find_record(STUD *);
save_load. cpp	save	存储信息	void save(STUD *);
	load	读取文件	STUD * load(void);
sort. cpp	sort_record	记录排序	void sort_record(STUD *);
	copyRecord	复制链表节点	void copyRecord(STUD * ,STUD *);
student. h	ASK 带参数宏定义	宏定义申请内存	
	结构声明	定义数据结构	
	函数包含与声明	函数声明和函数包含	



C 源程序的扩展名应为“.c”，如果在 Visual C++ 中编辑 C 程序，默认的扩展名为“.cpp”，两者皆可使用。

1.2 系统设计

1.2.1 student.h 头文件

在头文件中一般要包含一些相关的标准头文件，常用的标准头文件如下。

- ① ctype.h: 字符处理；
- ② math.h: 与数学处理函数有关的说明与定义；
- ③ stdio.h: 输入输出函数中使用的有关说明和定义；
- ④ string.h: 字符串函数的有关说明和定义；
- ⑤ stdlib.h: 其他常用函数等有关的说明与定义；
- ⑥ time.h: 支持系统时间函数。

在一般系统开发中，将若干个函数组织成文件，又将若干个文件构成系统，并以此来进行编程分工。作为共同开发的系统，为了共享彼此的过程资源（函数），建议将全体函数声明放在一个共用的头文件中。

在头文件中还可以定义外部变量、宏和结构类型等。

1. student.h 文件源代码

```
#ifndef _H_LETTER_H_A
#define _H_LETTER_H_A

#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>

#define ASK(p) do{\
p = (STUD *)malloc(sizeof(STUD));\
    if(p == NULL) { printf("memory fail"); exit(-1); }\
}while(0)

#define LEN sizeof(STUD)
#define NUM_SUBJECT 3

extern int count;
extern int savedTag;

typedef struct studentscore {
```



```

char number[20];           //学号
char name[20];             //姓名
char sex[4];               //性别
char classno[10];          //班级
double score[NUM_SUBJECT]; //成绩
double sum;                //总分
double average;            //平均分
struct studentscore * next;
}STUD;

STUD * input(STUD * );
void display(STUD * );
void save(STUD * );
STUD * load(void);
int menu_select(void);
void handle_menu(STUD * );
STUD * delete_record(STUD * );
void free_nodes(STUD * );
void quit(STUD * );
STUD * new_addclass(STUD * );
void find_record(STUD * );
void copyRecord(STUD * ,STUD * );
void sort_record(STUD * );

```

```
#endif
```

2. 代码解释

在上述头文件中,包含了4个标准头文件,13个自定义函数的声明,2个符号常量 LEN 和 NUM_SUBJECT 的宏定义,1个带参数的宏定义 ASK 及2个外部变量 count 和 saved-Tag 的声明。

此外,在头文件中采用了条件编译,通过条件编译能保证头文件中的结构、变量和函数只声明或定义一次,不会重复声明和定义,这一方法也称为“头文件卫士”。

```
#ifndef _H_LETTER_H_A
```

```
.....(条件编译内容)
```

```
#endif
```

上述两个语句就是条件编译,意思是:如果没有定义“_H_LETTER_H_A”这一宏名,则执行条件编译内容。

在系统具体运行过程中,第一次包含这一头文件时,由于没有定义宏“_H_LETTER_H_A”,则执行条件编译内容。条件编译内容的第一句将定义宏“_H_LETTER_H_A”。以后若再次包含这一头文件,由于已经定义了宏“_H_LETTER_H_A”,将不再执行条件编译内



容,这就保证了头文件中的结构、变量和函数只声明或定义一次。

1.2.2 student.cpp 文件

student.cpp 文件是主文件,在这个文件里定义了主函数 main。系统的使用界面,即菜单设计和选择都在这个文件中实现。

1. student.cpp 文件源代码

```
#include "student.h"

int count;                                //声明全局记录计数变量
int savedTag = 0;                         //记录是否已保存过数据
void main()
{
    printf("\n");
    printf("\t ***** \n");
    printf("\t *           这是一个           * \n");
    printf("\t *      简单的学生信息管理系统      * \n");
    printf("\t *      可以对学生信息进行简单的管理      * \n");
    printf("\t *      欢迎使用本系统           * \n");
    printf("\t ***** \n");
    printf("\n");

    count = 0;
    handle_menu(NULL);
}

//菜单处理函数
void handle_menu(STUD * top)
{
    for(;;){
        switch(menu_select())
        {
            case 1:
                top = input(top);
                break;
            case 2:
                display(top);
                break;
            case 3:
                save(top);
                break;
```



```
        case 4:
            free_nodes(top);
            top = load();
            break;
        case 5:
            top = delete_record(top);
            break;
        case 6:
            top = new_addclass(top);
            break;
        case 7:
            find_record(top);
            break;
        case 8:
            sort_record(top);
            break;
        case 9:
            quit(top);
            break;
    }
}

//菜单选择函数 menu_selected
int menu_select()
{
    char s[2];
    int cn = 0;
    printf("\n");
    printf(" 1. 增加记录\n");
    printf(" 2. 显示记录\n");
    printf(" 3. 保存记录\n");
    printf(" 4. 读取记录\n");
    printf(" 5. 删除记录\n");
    printf(" 6. 新建班级\n");
    printf(" 7. 查询记录\n");
    printf(" 8. 记录排序\n");
    printf(" 9. 结束运行\n");
    printf("\n 左边数字对应功能选择,选择 1—9:");
```



```
    for(;;)
    {
        gets(s);
        cn = atoi(s);
        if(cn<1||cn>9) printf("\n 输入错误,请重选择 1—9:");
        else break;
    }
    return cn;
}

/ *****
* 释放以 top 为头的链表的内存空间
* 参数 top:当前班级学生信息链表的表头
***** /
void free_nodes(STUD * top)
{
    STUD * p = top;
    STUD * old = p;
    while(p) //未到表头
    {
        old = p;
        p = p->next;
        free(old);
    }
    count = 0;
}

/ *****
* 结束运行,退出
* 参数 top:当前班级学生信息链表的表头
***** /

void quit(STUD * top)
{
    char str[10];
    if(savedTag == 1)//已改动,未保存
    {
        printf("通讯录已改动,是否保存? (Y/N)");
```