

# C++程序设计实验报告

题目名称: 课程平时成绩管理系统

姓 名: 孙重豪

学 号: 8207191520

专 业: 自动化与电气类

班 级: 1915 班

编写日期: 2020 年 7 月 15 日

# 目录

一. 课程设计要求.....	3
1. 程序要求.....	3
2. 内容要求.....	3
二. 课程设计目的.....	4
三. 课程设计内容.....	4
1. 程序设计说明.....	4
(1) 密码管理菜单.....	4
(2) 课程信息管理菜单.....	5
(3) 具体课程信息管理菜单.....	5
(4) 学生成绩信息管理菜单.....	5
2. 程序流程图.....	5
3. 程序功能模块.....	7
(1) 密码管理菜单.....	7
(2) 课程信息管理菜单.....	7
(3) 具体课程信息管理菜单.....	8
(4) 学生成绩信息管理菜单.....	8
四. 程序设计代码.....	9
1. 头文件(Student.h).....	9
2. 函数源文件(Student.cpp).....	10
3. 主函数源文件(main.cpp).....	22
五. 程序设计结果及分析.....	32
六. 收获与感想.....	38

# 一. 课程设计要求

## 1. 程序要求

本设计采用教师指导、学生自学和独立编程操作方式，先由学生查阅与本设计有关的书籍和资料，在两周时间内，完成课程设计内容中的一个管理系统，具体要求如下：

1. 独立思考：独立完成课程设计中各任务的设计和调试。
2. 数据管理：管理系统中的数据可用数据库的形式保存，也可用普通文件（文本文件或二进制文件）的形式保存。
3. 面向对象编程要求：要求使用类和对象进行程序设计，并能够合理设计类（包括对不同类的设计以及同一类的成员的设计），要求在程序中体现出函数重载、运算符重载、类的继承与派生等语法。
4. 尽量避免使用全局变量
5. 用户管理功能：所设计的系统要求能用于多个用户，每个用户管理自己的数据和登录密码。
6. 程序要求：界面美观、可操作性强、稳定性好。

## 2. 内容要求

### 选题十：课程平时成绩管理系统

#### 1. 系统要求

[1]编写一个学生平时成绩管理系统，对课程信息、学生信息和每个学生的平时成绩，以及平时成绩的考核方式进行管理。

[2]课程包括以下信息：课程编号、课程名称、任课老师等，上课学生的信息包括：学生学号，姓名，班级；学生的平时成绩至少由3项构成（如实验成绩、考勤成绩、作业成绩、讨论成绩、报告成绩等等）。不同课程平时成绩构成不同，老师可以对平时成绩构成和学生的平时成绩进行管理。

#### 2. 基本管理功能

#### 3. 课程管理，包括：

添加：添加课程信息。

显示：显示课程信息。考核方式管理：增加考核方式，修改考核方式等

#### 4. 学生信息管理，包括：

添加：添加一个课程的学生；

修改：修改一个学生的信息；

查询：查询学生的信息

#### 5. 平时成绩管理：

添加：添加某一个学生的平时成绩；

修改：修改某项平时成绩；

显示：显示所有学生平时成绩。

查询：查询某个学生的平时成绩

删除和恢复：删除和恢复某个学生的平时成绩

#### 6. 统计功能：根据需要设计合理的统计功能，比如：每个分数段的学生数，

作业分数低于多少的学生，考勤情况差的学生等。

7. 退出功能：要求点击退出，可以退出系统。

8. 其他要求及说明：

[1] 要求每个课程的学生数不少于 30 人。

[2] 假定平时成绩满分为 100 分，总平时成绩由多项合成。每一项平时成绩满分也为 100 分。

## 二. 课程设计目的

本设计是课程——《面向对象编程(C++)》的一个关键实践环节。它是根据教学计划的要求，在教师的指导下，对学生实施程序设计训练的必要过程，是对前期课堂学习内容的综合应用及其效果的检验和提高。其目的在于培养学生综合运用理论知识来分析和解决实际问题的能力、以及严谨的科学态度和良好的程序设计习惯。在课程设计中，要求学生能够遵循软件开发过程的基本规范，运用面向对象程序设计的方法，按照选定的课程设计题目要求，独自地完成设计、编写、调试和测试应用程序及编写文档的任务。

## 三. 课程设计内容

### 1. 程序设计说明

总体结构我采取模块化方式进行程序设计，要求程序的功能设计、数据结构设计以及整体结构设计合理。也可根据自己对题目的理解增加新的功能模块。

系统以文本菜单界面的方式工作，界面整洁，引导性比较好，易上手。系统整体以四个菜单为框架，层层嵌套，层层深入，通过四个 switch 语句嵌套实现，菜单中的每一个选项背后有一个函数，实现一个特定的功能。

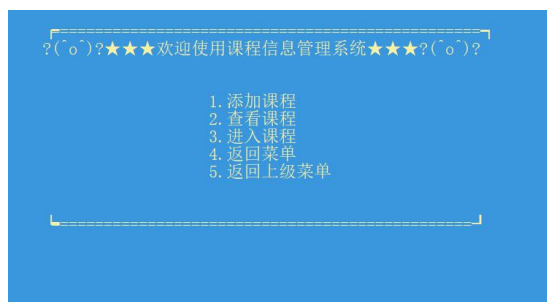
四个界面如下：

(1) 密码管理菜单



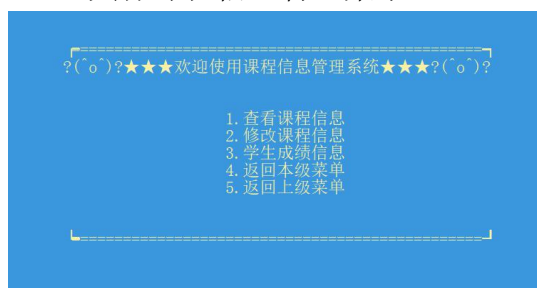
主要功能是登录、注册、修改、删除密码。

## (2) 课程信息管理菜单



主要功能是添加、查看、进入课程。

## (3) 具体课程信息管理菜单



主要功能是查看、修改具体课程信息，操作该课程学生信息。

## (4) 学生成绩信息管理菜单



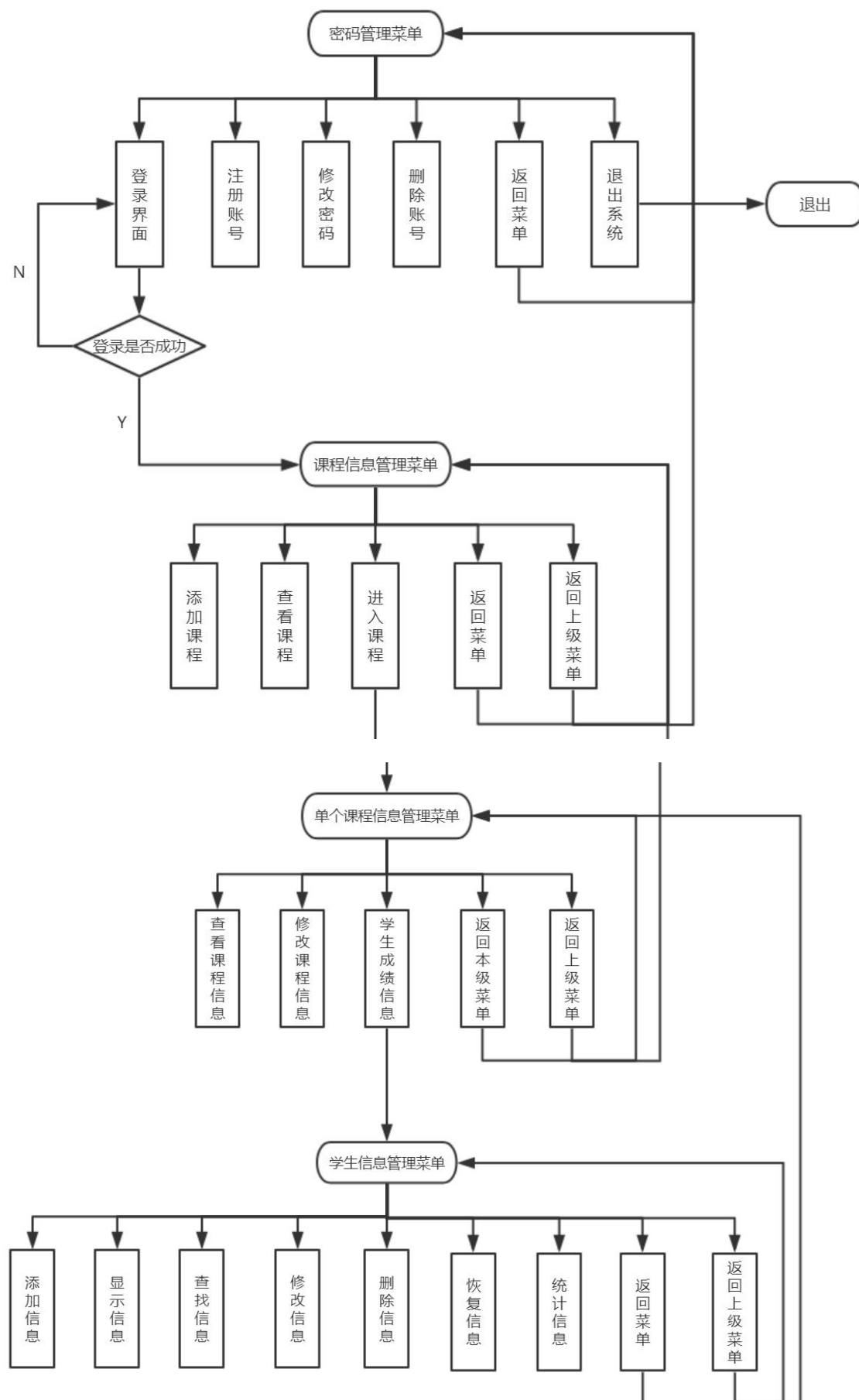
主要功能是添加、显示、查找、修改、删除、恢复、统计学生信息。

## 2. 程序流程图

整体流程如下：

运行程序后，首先出现密码管理菜单，若初始没有账号可以先进行注册，如果已有账号可以进行密码修改和账号删除，以及登录，选择登录进入登录界面，密码验证成功后进入课程信息管理菜单，可以添加课程和查看课程，选择进入课程可以进入单个课程信息管理菜单，可以查看和修改该课程信息，选择学生成绩信息可以进入学生信息管理菜单，可以选择添加、显示、查找、修改、删除、恢复、统计学生成绩信息。在进入每个子菜单后可以选择返回菜单返回本级菜单，选择返回上级菜单可以跳出本级菜单，直到返回第一级菜单，选择退出系统即可退出。

流程图如下：



### 3. 程序功能模块

#### (1) 密码管理菜单

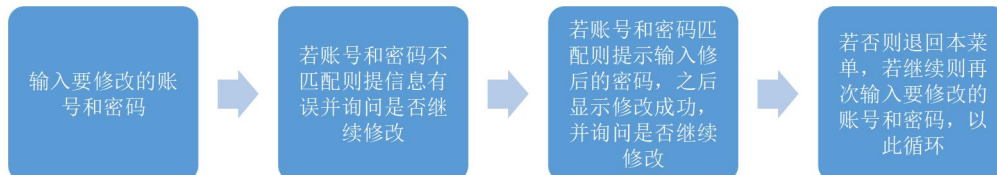
##### ① 登录



##### ② 注册密码 (调用 ADD() 函数)



##### ③ 修改密码 (调用 ALTER() 函数)



##### ④ 删除账号 (调用 DEL() 函数)



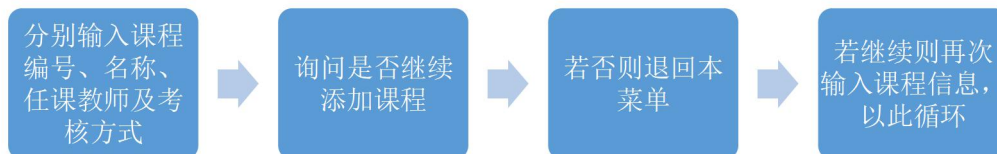
##### ⑤ 返回本菜单

##### ⑥ 退出系统



#### (2) 课程信息管理菜单

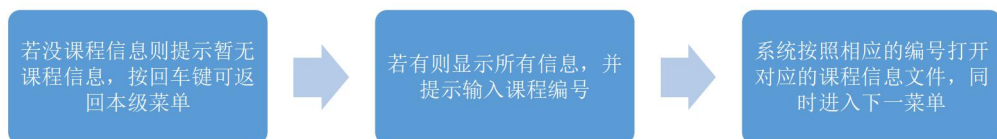
##### ① 添加课程 (调用 Add() 函数, 函数内部调用 course 类的 set() 函数)



##### ② 查看课程 (调用 Show() 函数)



##### ③ 进入课程 (调用 Show() 函数显示课程信息)

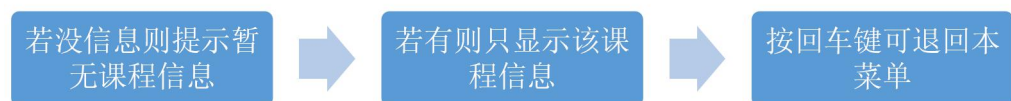


④返回本菜单

⑤返回上级菜单

(3) 具体课程信息管理菜单

①查看课程(调用 Show() 函数)



②修改课程(调用 Alter() 函数)



③学生成绩信息

直接显示下一菜单内容

④返回本菜单

⑤返回上级菜单

(4) 学生成绩信息管理菜单

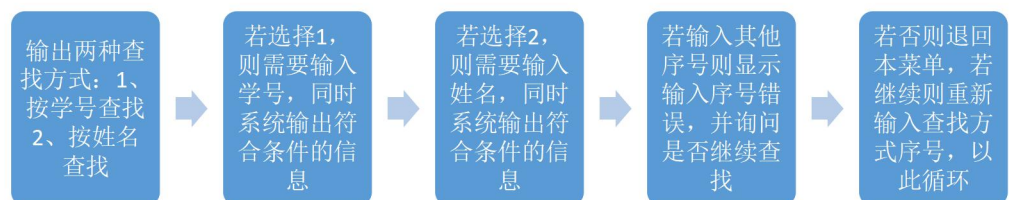
①添加信息(调用 add() 函数)



②显示信息(调用 show() 函数、函数内部调用 student 类的 show() 函数)



③查找信息(调用 search() 函数)



④修改信息(调用 alter() 函数)

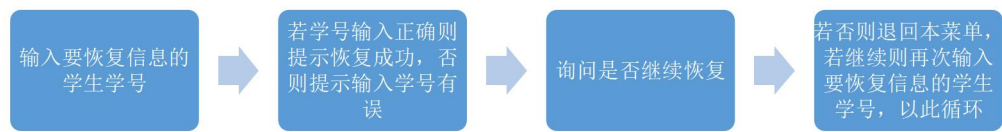


⑤删除信息(调用 del() 函数)





#### ⑥恢复信息(调用 regain() 函数)



#### ⑦统计信息(调用 data() 函数)

输出所有的统计信息，按回车键可退回本菜单

#### ⑧返回本菜单

#### ⑨返回上级菜单

## 四. 程序设计代码

### 1. 头文件(Student.h)

```
#pragma once
#include<iostream>
#include<string>
using namespace std;

class pass
{
public:
    string user, passWord, filecourse, filecache;
    void Set();
    friend void ADD(pass pa[], int& p);
    friend void ALTER(pass pa[], int p);
    friend void DEL(pass pa[], int& p);
};

class student
{
public:
    int num;//学号
    string name, clas;//姓名, 班级
    float score1, score2, score3;//三门平时成绩
    void set1();//输入学号
    void show();//输出信息
```

```

};
class course
{
public:
    int Num;
    string Name, teacher, exam[3], file;
    void set();
    friend void Add(course co[], int& s); //添加课程信息
    friend void Show(course co[], int s); //显示全部课程信息
    friend void Show(course co); //显示当前课程信息
    friend void Alter(course& co); //修改当前课程信息
};

void add(student st[], int& t, course co); //输入学生成绩
void show(student st[], int t, course co); //显示全部学生成绩
void search(student st[], int t, course co); //查询学生信息
void alter(student st[], int t, course co); //修改学生信息
void del(student st[], student ca[], int& t, int& n, course co); //删除学生信息
void data(student st[], int t, course co); //统计成绩信息
void regain(student st[], student ca[], int& t, int& n); //恢复学生信息

```

## 2. 函数源文件(Student.cpp)

```

#include<iostream>
#include<string>
#include "Student.h"
#include <iomanip>
#include<stdlib.h>
using namespace std;

void student::set1()
{
    cin >> num>> name>> clas>> score1>> score2 >> score3;
}

void student::show()
{
    cout << num << setw(8) << name << setw(20) << clas << setw(10) << score1
        << setw(10) << score2 << setw(10) << score3 << endl;
}

```

```

void add(student st[], int &t, course co)
{
    system("cls");
    int k = t;
    string str;
    while (true)
    {
        cout << "请依次输入以下信息" << endl;
        cout << "学号" << setw(5) << "姓名" << setw(5) << "班级" << setw(7)
            << co.exam[0] << setw(5) << co.exam[1] << setw(5) << co.exam[2] << endl;
        st[k].set1();
        k++;
        t = k;
        cout << "是否继续输入(Y or N)?" << endl;
        cin >> str;
        if (str == "y" || str == "Y")
        {
            continue;
        }
        else
        {
            break;
        }
    }
}

void show(student st[], int t, course co)
{
    system("cls");
    if (t == 0)
        cout << "暂无学生信息" << endl;
    else
    {
        cout << "学号" << setw(10) << "姓名" << setw(14) << "班级" << setw(17)
            << co.exam[0] << setw(11) << co.exam[1] << setw(10) << co.exam[2] << endl;
        for (int k = 0; k < t; k++)
        {
            st[k].show();
        }
    }
}

void search(student st[], int t, course co)

```

```

{
    system("cls");
    int ch, number;
    string N;
    static char T = 'y';
    cout << "1. 按学号查找" << endl;
    cout << "2. 按姓名查找" << endl;
    while ((T == 'y') || (T == 'Y'))
    {
        cout << "请输入查找方式序号: ";
        cin >> ch;
        if (ch == 1)
        {
            cout << "请输入要查找学生的学号: ";
            cin >> number;
            for (int i = 0; i < t; i++)
            {
                if (number == st[i].num)
                {
                    cout << "学号" << setw(10) << "姓名" << setw(14) << "班级" << setw(17)
                        << co.exam[0] << setw(11) << co.exam[1] << setw(10) << co.exam[2]
<< endl;

                    st[i].show();
                }
            }
        }
        else if (ch == 2)
        {
            cout << "请输入要查找学生的姓名: ";
            cin >> N;
            for (int i = 0; i < t; i++)
            {
                if (N.compare(st[i].name) == 0)
                {
                    cout << "学号" << setw(10) << "姓名" << setw(14) << "班级" << setw(17)
                        << co.exam[0] << setw(11) << co.exam[1] << setw(10) << co.exam[2]
<< endl;

                    st[i].show();
                }
            }
        }
        else
            cout << "输入序号错误!!!" << endl;
        cout << "是否继续查找 (Y or N)?" << endl;
    }
}

```

```

        cin >> T;
    }
}

void alter(student st[], int t, course co)
{
    system("cls");
    int number;
    static char T = 'y';
    while ((T == 'y') || (T == 'Y'))
    {
        cout << "请输入要修改信息的学生学号: ";
        cin >> number;
        for (int i = 0; i < t; i++)
        {
            if (number == st[i].num)
            {
                cout << "要修改的学生信息如下" << endl;
                cout << "学号" << setw(10) << "姓名" << setw(14) << "班级" << setw(17)
                    << co.exam[0] << setw(11) << co.exam[1] << setw(10) << co.exam[2] <<
endl;

                st[i].show();
                cout << "请依次输入修改后的学生信息: " << endl;
                cout << "学号" << setw(5) << "姓名" << setw(5) << "班级" << setw(7)
                    << co.exam[0] << setw(5) << co.exam[1] << setw(5) << co.exam[2] <<
endl;

                st[i].set1();
            }
        }

        cout << "是否继续修改 (Y or N)?" << endl;
        cin >> T;
    }
}

void del(student st[], student ca[], int &t, int& n, course co)
{
    system("cls");
    int id;
    string str; //用来判断是否继续输入的字符串
    while (true)
    {
        cout << "请输入要删除学生的学号: ";
        cin >> id;
    }
}

```

```

int flag = 0; //用来统计删除后剩余的数组元素个数
student a[30];
    for (int i = 0; i < t; i++)
    {
        if (st[i].num != id)
        {
            a[flag] = st[i];
            flag++;
        }
        else
        {
            ca[n] = st[i];
            n++;
        }
    }
    if (flag < t)
    {
        t = flag;
        for (int k = 0; k < t; k++)
        {
            st[k] = a[k];
        }
        cout << id << "号学生信息删除成功!" << endl;
    }
    else
    {
        cout << "输入学号有误! \n";
    }
    cout << "是否继续删除(Y or N)?";
    cin >> str;
    if (str == "y" || str == "Y")
    {
        continue;
    }
    else
    {
        break;
    }
}

}

void data(student st[], int t, course co)
{
    system("cls");

```

```

float sum;
int x = 0, y = 0, z = 0;
for (int b = 0; b < t; b++)
{
    sum = 0.3 * st[b].score1 + 0.3 * st[b].score2 + 0.4 * st[b].score3;
    if (sum >= 80)
        x++;
    else if (sum >= 60)
        y++;
    else
        z++;
}

cout << "总平时成绩高于 80 分的有" << x << "人" << endl;
cout << "总平时成绩 60-80 分的有" << y << "人" << endl;
cout << "总平时成绩低于 60 分的有" << z << "人" << endl;
cout << co.exam[0] << "不及格的学生信息如下" << endl;
cout << "学号" << setw(10) << "姓名" << setw(14) << "班级" << setw(17)
    << co.exam[0] << setw(11) << co.exam[1] << setw(10) << co.exam[2] << endl;
for (int b = 0; b < t; b++)
{
    if(st[b].score1<60)
        st[b].show();
}

cout << co.exam[1] << "不及格的学生信息如下" << endl;
cout << "学号" << setw(10) << "姓名" << setw(14) << "班级" << setw(17)
    << co.exam[0] << setw(11) << co.exam[1] << setw(10) << co.exam[2] << endl;
for (int b = 0; b < t; b++)
{
    if (st[b].score2 < 60)
        st[b].show();
}

cout << co.exam[2] << "不及格的学生信息如下" << endl;
cout << "学号" << setw(10) << "姓名" << setw(14) << "班级" << setw(17)
    << co.exam[0] << setw(11) << co.exam[1] << setw(10) << co.exam[2] << endl;
for (int b = 0; b < t; b++)
{
    if (st[b].score3 < 60)
        st[b].show();
}
}

void regain(student st[], student ca[], int& t, int& n)
{
    system("cls");

```

```

int id,b,flag;
int c;
string str;
student a[30];
while (true)
{
    cout << "请输入要恢复学生的学号： ";
    cin >> id;
    flag = 0;
    c = n;
    for (b = 0;b < c;b++)
    {
        if (ca[b].num == id)
        {
            st[t] = ca[b];
            t++;
            n--;
        }
        else
        {
            a[flag] = ca[b];
            flag++;
        }
    }
    for (int d = 0;d < n;d++)
    {
        ca[d] = a[d];
    }
    if (flag == c)
        cout << "输入的学号不存在！ " << endl;
    else
        cout << "恢复成功！ " << endl;
    cout << "是否继续恢复(Y or N)?";
    cin >> str;
    if (str == "y" || str == "Y")
    {
        continue;
    }
    else
    {
        break;
    }
}
}

```



```

void ADD(pass pa[], int& p)
{
    system("cls");
    string str;
    int k = p;
    while (true)
    {
        pa[k].Set();
        k++;
        p = k;
        cout << "是否继续输入(Y or N)?" << endl;
        cin >> str;
        if (str == "y" || str == "Y")
        {
            continue;
        }
        else
        {
            break;
        }
    }
}

void ALTER(pass pa[], int p)
{
    system("cls");
    string str, u, pw;
    while (true)
    {
        int flag = 0;
        cout << "输入要修改的账户和密码" << endl;
        cout << "账号: ";
        cin >> u;
        cout << "密码: ";
        cin >> pw;
        for (int r = 0; r < p; r++)
        {
            if ((u == pa[r].user) && (pw == pa[r].passWord))
            {
                cout << "请输入修改后的账户和密码" << endl;
                cout << "账号: ";
                cin >> pa[r].user;
                cout << "密码: ";
            }
        }
    }
}

```

```

        cin >> pa[r].passWord;
        cout << "修改成功!" << endl;
    }
    else
        flag++;
}
if (flag == p)
{
    cout << "输入信息有误!" << endl;
}
cout << "是否继续修改(Y or N)?" << endl;
cin >> str;
if (str == "y" || str == "Y")
{
    continue;
}
else
{
    break;
}
}
}

```

```

void DEL(pass pa[], int& p)
{
    system("cls");
    string str,u,pw;
    int k;
    pass a[10];
    while (true)
    {
        int flag = 0;
        cout << "输入要删除的账户和密码" << endl;
        cout << "账号: ";
        cin >> u;
        cout << "密码: ";
        cin >> pw;
        for (int r = 0;r < p;r++)
        {
            if ((u != pa[r].user) || (pw != pa[r].passWord))
            {
                a[flag] = pa[r];
                flag++;
            }
        }
    }
}

```

```

    }
    if (flag == (p - 1))
    {
        p = flag;
        for (int k = 0; k < p; k++)
        {
            pa[k] = a[k];
        }
        cout << "账户" << u << "删除成功!" << endl;
    }
    else
    {
        cout << "输入信息有误!" << endl;
    }
    cout << "是否继续删除(Y or N)?" << endl;
    cin >> str;
    if (str == "y" || str == "Y")
    {
        continue;
    }
    else
    {
        break;
    }
}
}

```

```

void Add(course co[], int &s)
{
    system("cls");
    string str;
    int k = s;
    while (true)
    {
        co[k].set();
        k++;
        s = k;
        cout << "是否继续输入(Y or N)?" << endl;
        cin >> str;
        if (str == "y" || str == "Y")
        {
            continue;
        }
        else

```

```

        {
            break;
        }
    }
}

void Show(course co[], int s)
{
    system("cls");
    if (s == 0)
        cout << "暂无课程信息" << endl;
    else
    {
        for (int k = 0; k < s; k++)
        {
            cout << "课程编号: " << co[k].Num << " 课程名称: " << co[k].Name << " 任课
老师: " << co[k].teacher << " 考核方式:";
            for (int b = 0; b < 3; b++)
            {
                cout << co[k].exam[b];
                cout << " ";
            }
            cout << "信息保存位置: " << co[k].file << endl;
        }
    }
}

void Show(course co)
{
    system("cls");
    cout << "课程编号: " << co.Num << " 课程名称: " << co.Name << " 任课老师: " << co.teacher
<< " 考核方式:";
    for (int b = 0; b < 3; b++)
    {
        cout << co.exam[b];
        cout << " ";
    }
    cout << "信息保存位置: " << co.file << endl;
}

void Alter(course &co)
{
    system("cls");

```

```

    cout << co.Name << "的考核方式为: ";
    for (int b = 0;b < 3;b++)
    {
        cout << co.exam[b];
        cout << " ";
    }
    cout << "\n 输入修改后的考核方式" << endl;
    for (int b = 0;b < 3;b++)
    {
        cin >> co.exam[b];
    }
    cout << "考核方式修改成功!" << endl;
}

void course::set()
{
    cout << "课程编号: ";
    cin >> Num;
    cout << "课程名称: ";
    cin >> Name;
    cout << "任课老师: ";
    cin >> teacher;
    cout << "考核方式: ";
    for (int b = 0;b < 3;b++)
    {
        cin >> exam[b];
        cout << "\t ";
    }
    cout << "\n 信息保存位置: ";
    cin >> file;
}

void pass::Set()
{
    cout << "账号: ";
    cin >> user;
    cout << "密码: ";
    cin >> passWord;
    cout << "课程保存地址: ";
    cin >> filecourse;
    cout << "预恢复信息保存位置: ";
    cin >> filecache;
}

```

### 3. 主函数源文件(main.cpp)

[illegible]



```
welcome();  
while (choice != 6)  
{  
  
    welcome();  
    cout << "\n 请输入>>";  
    cin >> choice;  
    e = getchar();  
    switch (choice)  
    {  
        case 1:  
            {  
  
                if (currentN == 0)  
                {  
  
                    cout << "当前没有用户，请先注册！" << endl;  
                }  
                else  
                {  
  
                    string username;  
                    string password;  
                    string strr;  
                    char g;  
                    char a[20];  
                    while (true)  
                    {  
  
                        system("cls");  
                        cout << "\n\n\t\t\t\t\t\t\t┐\n";  
=====┘ \n";  
                        cout << "\t\t\t\t\t\t\t?(^o^)?★★★登录界面★★★?(^o^)?\n\n";  
                        cout << "\t\t\t\t\t\t\t用户名：";  
                        getline(cin, username, '\n');  
                        cout << "\t\t\t\t\t\t\t密码：";  
                        int z;  
                        for (z = 0;z < 20;z++)  
                        {  
  
                            a[z] = _getch();  
                            if (a[z] == 13) //响应回车  
                            {  
  
                                break;  
                            }  
                        }  
                        else if (a[z] == '\b') //响应退格键  
                        {  
  
                            if (z == 0) //如果当前已经没有字符，则直接跳出  
                            {  
  
                                continue;
```



```

    }
    //屏幕上删掉的字符填成空格，并回退光标
    cout << "\b";
    cout << " ";
    cout << "\b";
    //将数组响应位置填成 0
    z--;
    a[z] = 0;
    //将 i 往前置一位，因为接下来会执行 for 循环的 i++
    z--;
    //清空输入缓冲，避免回车等其他字符捣乱
    cin.clear();
}
else
{
    cout << "*";
}
}
cout << endl;
//a 字符串结尾，并赋给 password
a[z] = 0;
password = a;
F = 0;
for (int r = 0; r < currentN; r++)
{
    if ((username == Pass[r].user) && (password ==
Pass[r].passWord))
    {
        q = r;
    }
    else
        F++;
}
if (F==(currentN-1))
{
    cout << "登录成功!" << endl;

    int choice1, choice2, choice3;
    ifstream filein(Pass[q].filecourse, ios::binary);
    if (filein)
    {
        filein.read((char*)&currentN1, sizeof(currentN1));
        for (int i = 0; i < currentN1; i++)
        {

```

```

        filein.read((char*)&cou[i], sizeof(cou[i]));
    }
}
filein.close();
welcome1();
choice1 = 1;
while (choice1 != 5)
{
    welcome1();
    cout << "\n 请输入>>";
    cin >> choice1;
    switch (choice1)
    {
    case 1:
        Add(cou, currentN1);
        break;
    case 2:
        Show(cou, currentN1);
        system("pause");
        break;
    case 3:
    {
        Show(cou, currentN1);
        if (currentN1 == 0)
        {
            cout << "按 4 返回菜单!" << endl;
            break;
        }
    }
    else
    {
        cout << "请输入课程编号: ";
        cin >> x;
        x--;
        welcome2();
        choice2 = 1;
        while (choice2 != 5)
        {
            welcome2();
            cout << "\n 请输入>>";
            cin >> choice2;
            if (choice2 == 1)
            {
                Show(cou[x]);
                system("pause");
            }
        }
    }
}
}

```

```

    }
    else if (choice2 == 2)
    {
        Alter(cou[x]);
    }
    else if (choice2 == 3)
    {

        ifstream infile(cou[x].file,

ios::binary);

        if (infile)
        {
            infile.read((char*)&currentN2,

sizeof(currentN2));

            for (int i = 0; i < currentN2; i++)
            {
                infile.read((char*)&stu[i],

sizeof(stu[i]));

            }
        }
        infile.close();
        ifstream infile1(Pass[q].filecache,

ios::binary);

        if (infile1)
        {
            infile1.read((char*)&currentN3,

sizeof(currentN3));

            for (int i = 0; i < currentN3; i++)
            {
                infile1.read((char*)&cache[i],

sizeof(cache[i]));

            }
        }
        infile1.close();
        welcome3();
        choice3 = 1;
        while (choice3 != 9)
        {
            welcome3();
            cout << "\n 请输入>>";
            cin >> choice3;
            if (choice3 == 1)
            {
                add(stu, currentN2, cou[x]);
            }
        }
    }
}

```

```

        cou[x]);

currentN3, cou[x]);

currentN3);

        }
        else if (choice3 == 2)
        {
            show(stu, currentN2, cou[x]);
            system("pause");
        }
        else if (choice3 == 3)
        {
            search(stu, currentN2,

        }
        else if (choice3 == 4)
        {
            alter(stu, currentN2, cou[x]);
        }
        else if (choice3 == 5)
        {
            del(stu, cache, currentN2,

        }
        else if (choice3 == 6)
        {
            regain(stu, cache, currentN2,

        }
        else if (choice3 == 7)
        {
            data(stu, currentN2, cou[x]);
            system("pause");
        }
        else if (choice3 == 8)
        {
            welcome3();
        }
        else if (choice3 == 9)
        {
            welcome2();
        }
        else
        {
            cout << "\t\t任意键返回本菜单

            \n";

            welcome3();
        }
    }
}

```



```

        {
            break;
        }
    }
}
system("pause");
break;
}
case 2:
    ADD(Pass, currentN);
    break;
case 3:
    ALTER(Pass, currentN);
    break;
case 4:
    DEL(Pass, currentN);
    break;
case 5:
    welcome();
    break;
case 6:
    {
        cout << "退出成功! " << endl;
        break;
    }
default:
    cout << "\t\t 任意键返回本菜单\n";
    welcome();
}
}

/*往文件写入数据*/
ofstream outfile1(Pass[q].filecache, ios::binary);
if (!outfile1)
{
    cout << "恢复文件不能被打开! " << endl;
    return 0;
}
outfile1.write((char*)&currentN3, sizeof(currentN3));
for (int i = 0; i < currentN3; i++)
{
    outfile1.write((char*)&cache[i], sizeof(cache[i]));
}
outfile1.close();

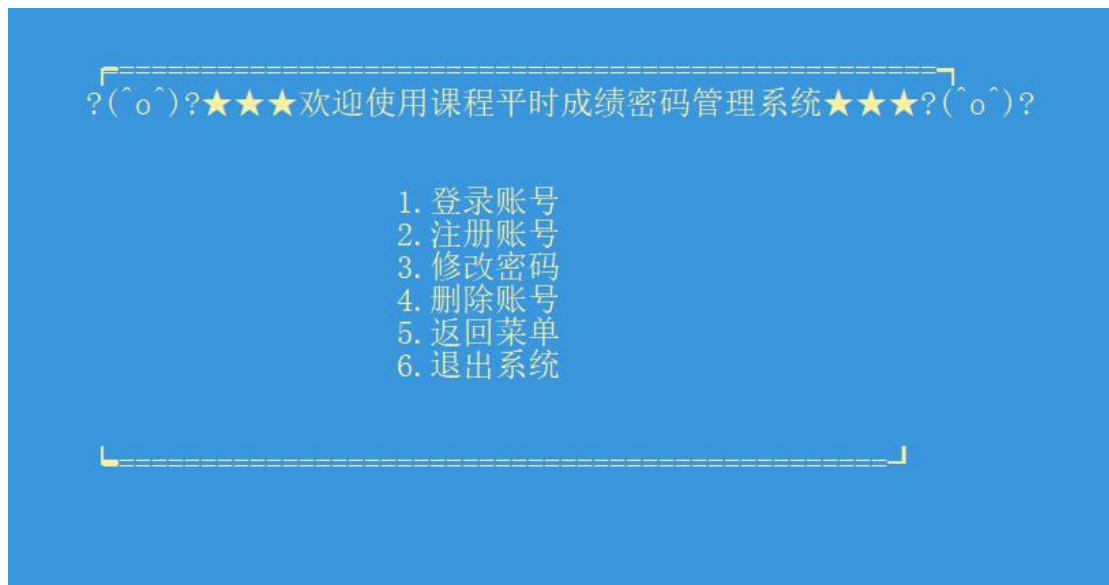
```

```

ofstream outfile(cou[x].file, ios::binary);
if (!outfile)
{
    cout << "文件不能被打开!" << endl;
    return 0;
}
outfile.write((char*)&currentN2, sizeof(currentN2));
for (int i = 0; i < currentN2; i++)
{
    outfile.write((char*)&stu[i], sizeof(stu[i]));
}
outfile.close();
ofstream fileout(Pass[q].filecourse, ios::binary);
if (!fileout)
{
    cout << "文件不能被打开!" << endl;
    return 0;
}
fileout.write((char*)&currentN1, sizeof(currentN1));
for (int i = 0; i < currentN1; i++)
{
    fileout.write((char*)&cou[i], sizeof(cou[i]));
}
fileout.close();
ofstream Fileout("password.dat", ios::binary);
if (!Fileout)
{
    cout << "文件不能被打开!" << endl;
    return 0;
}
Fileout.write((char*)&currentN, sizeof(currentN));
for (int i = 0; i < currentN; i++)
{
    Fileout.write((char*)&Pass[i], sizeof(Pass[i]));
}
Fileout.close();
return 0;
}

```

## 五. 程序设计结果及分析





C:\Windows\system32\cmd.exe

输入要修改的账户和密码

账号: szh

密码: 12345

请输入修改后的账户和密码

账号: SZH

密码: 1234

修改成功!

是否继续修改(Y or N)?

n

C:\Windows\system32\cmd.exe

输入要删除的账户和密码

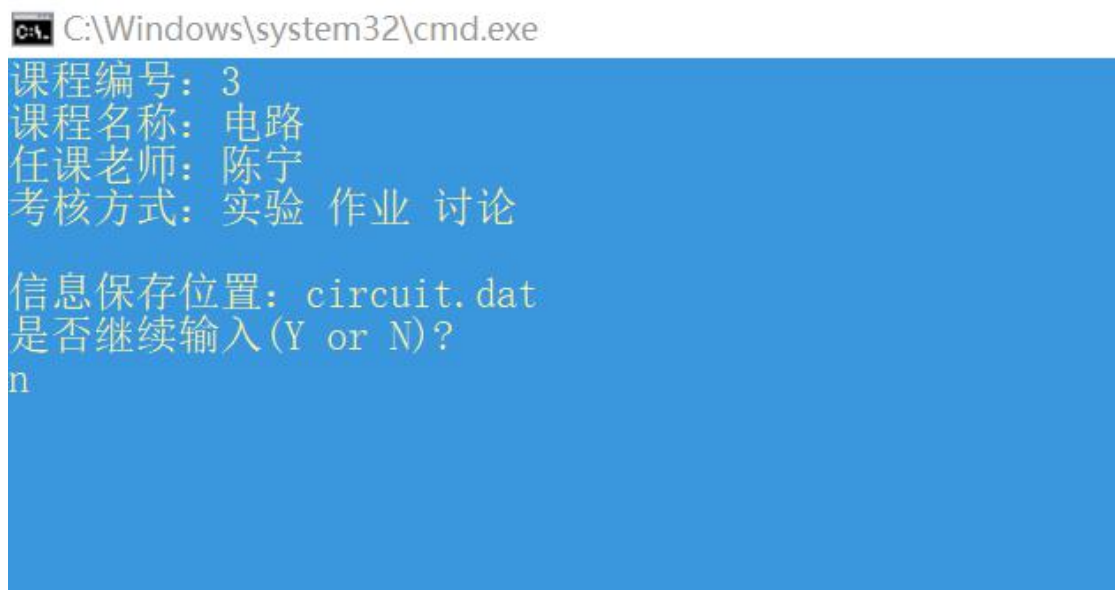
账号: SZH

密码: 1234

账户SZH删除成功!

是否继续删除(Y or N)?

N



C:\Windows\system32\cmd.exe

课程编号: 1 课程名称: 高数 任课老师: 张齐 考核方式: 作业 考勤 讨论 信息保存位置: math.dat  
课程编号: 2 课程名称: 大物 任课老师: 彭政 考核方式: 实验 作业 考勤 信息保存位置: physics.dat  
课程编号: 3 课程名称: 电路 任课老师: 陈宁 考核方式: 实验 作业 讨论 信息保存位置: circuit.dat  
请按任意键继续. . .

C:\Windows\system32\cmd.exe

课程编号: 1 课程名称: 高数 任课老师: 张齐 考核方式: 作业 考勤 讨论 信息保存位置: math.dat  
课程编号: 2 课程名称: 大物 任课老师: 彭政 考核方式: 实验 作业 考勤 信息保存位置: physics.dat  
课程编号: 3 课程名称: 电路 任课老师: 陈宁 考核方式: 实验 作业 讨论 信息保存位置: circuit.dat  
请输入课程编号: 1

┌───┐  
?(^o^)?★★★★欢迎使用课程信息管理系统★★★★?(^o^)?

1. 查看课程信息
2. 修改课程信息
3. 学生成绩信息
4. 返回本级菜单
5. 返回上级菜单

└───┘

C:\Windows\system32\cmd.exe

高数的考核方式为: 讨论 作业 考勤  
输入修改后的考核方式  
作业 考勤 讨论

?(^o^)?★★★★欢迎使用学生信息管理系统★★★★?(^o^)?

1. 添加信息
2. 显示信息
3. 查找信息
4. 修改信息
5. 删除信息
6. 恢复信息
7. 统计信息
8. 返回菜单
9. 退出系统

C:\Windows\system32\cmd.exe

请依次输入以下信息  
学号 姓名 班级 作业 考勤 讨论  
190102 李四 自动化1901 70 80 76  
是否继续输入(Y or N)?  
n

C:\Windows\system32\cmd.exe

学号	姓名	班级	作业	考勤	讨论
190101	张三	自动化1901	80	70	90
190102	李四	自动化1901	70	80	76
190103	王五	自动化1901	80	77	78
190104	赵六	自动化1901	85	60	75
190105	陈七	自动化1901	50	78	83
190106	张三三	自动化1901	98	90	93
190107	李四四	自动化1901	88	90	59
190108	王五五	自动化1901	75	90	84
190109	赵六六	自动化1901	60	50	70
190110	陈七七	自动化1901	80	70	50

请按任意键继续. . .

C:\Windows\system32\cmd.exe

1. 按学号查找
2. 按姓名查找

请输入查找方式序号: 1

请输入要查找学生的学号:

学号	姓名	班级
190101	张三	自动化1901

作业
80

考勤
70

讨论
90

是否继续查找 (Y or N)?

Y

请输入查找方式序号: 2

请输入要查找学生的姓名:

学号	姓名	班级
190103	王五	自动化1901

作业
80

考勤
77

讨论
78

是否继续查找 (Y or N)?

n

C:\Windows\system32\cmd.exe

请输入要修改信息的学生学号: 190110

要修改的学生信息如下

学号	姓名	班级
190110	陈七七	自动化1901

作业
80

考勤
70

讨论
50

请依次输入修改后的学生信息:

学号	姓名	班级	作业	考勤	讨论
190110	陈七七	自动化1901	70	80	50

是否继续修改 (Y or N)?

C:\Windows\system32\cmd.exe

请输入要删除学生的学号: 190110

190110号学生信息删除成功!

是否继续删除 (Y or N)?N



C:\Windows\system32\cmd.exe

请输入要恢复学生的学号: 190110  
恢复成功!  
是否继续恢复(Y or N)?N

C:\Windows\system32\cmd.exe

总平时成绩高于80分的有3人  
总平时成绩60-80分的有7人  
总平时成绩低于60分的有0人  
作业不及格的学生信息如下:

学号	姓名	班级	作业	考勤	讨论
190105	陈七	自动化1901	50	78	83

考勤不及格的学生信息如下

学号	姓名	班级	作业	考勤	讨论
190109	赵六六	自动化1901	60	50	70

讨论不及格的学生信息如下

学号	姓名	班级	作业	考勤	讨论
190107	李四四	自动化1901	88	90	59
190110	陈七七	自动化1901	80	70	50

请按任意键继续. . .

course1	2020/7/16 20:39	DAT 文件	1 KB
csch1	2020/7/16 20:39	DAT 文件	1 KB
math	2020/7/16 20:39	DAT 文件	1 KB
password	2020/7/16 20:39	DAT 文件	1 KB
physics	2020/7/16 15:04	DAT 文件	1 KB

## 六. 收获与感想

通过这次的 C++语言程序设计, 我获得了很多经验与知识, 虽然在敲代码的过程中比较劳累, 但是遇到一些不会的问题咨询他人后, 自己成功的敲出这个程序的时候, 内心还是有成就感的。在这里, 十分感谢老师的授课与指导, 并且经过此次课程设计的训练, 自己逻辑思维也加强了许多, 对 C++的知识有了更深入的理解, 敲代码的能力得以提升, 特别是整个程序的函数写完后, 怎么调用更符合用户的需求、一次进入系统和之后进入系统界面的设计真的用心思考了, 相对于 C 语言程序设计, 自己的自学能力得到了提高, 文件操作也掌握的非常好, 这也是大学主要培养的能力。