

课程设计报告

课程名称：计算机程序设计基础 2

| | |
|----|----------------------------------|
| 班级 | 无 08 |
| 学号 | 2020010841 |
| 姓名 | 李煜彤 |
| 邮箱 | liyutong20@mails.tsinghua.edu.cn |

目录

| | |
|-------------------|----|
| 一、系统需求分析 | 3 |
| 二、系统总体设计 | 6 |
| 三、系统详细设计 | 8 |
| 四、系统调试 | 12 |
| 五、测试结果与分析 | 13 |
| 六、总结 | 17 |
| 七、附录一：完成记录 | 19 |
| 八、附录二：源程序清单 | 20 |
| 九、附录三：评分表 | 46 |

一、系统需求分析

学生成绩管理系统可以录入、查询、修改学生的成绩，它包括：信息条目序号、学生姓名、课程名称、课程学分数、学生该课程的成绩（百分制成绩，等级制成绩，绩点）。学生成绩管理系统能够提供以下功能：

- 1) **录入**学生的成绩信息：从键盘输入数据，依次输入：学生姓名、学生参加课程、该课程学分、百分制成绩，系统会自动匹配序号，并给出其百分制成绩对应的等级制成绩和绩点，其对应规则如下：

| 等级制成绩 | 绩点 | 对应百分制成绩范围 |
|-------|-----|-----------|
| A+ | 4.0 | 100 |
| A | | 95~99 |
| A- | | 90~94 |
| B+ | 3.6 | 85~89 |
| B | 3.3 | 80~84 |
| B- | 3.0 | 77~79 |
| C+ | 2.6 | 73~76 |
| C | 2.3 | 70~72 |
| C- | 2.0 | 67~69 |
| D+ | 1.6 | 63~66 |
| D | 1.3 | 60~62 |
| F | 0 | 0~59 |

输入要求：姓名 课程名称 学分 百分制成绩

李煜彤 英语阅读写作 A 2 92

假设已经有 10 个信息条目，那么录入的信息条目为：

11 李煜彤 英语阅读写作 A 2 92 A- 4.0

- 2) **查询**某条成绩信息：输入信息条目序号，可以显示对应的信息条目。

输入要求：

11

输出结果：

| 序号 | 姓名 | 课程名称 | 学分 | 百分制成绩 | 等级制成绩 | 绩点 |
|----|-----|----------|----|-------|-------|-----|
| 11 | 李煜彤 | 英语阅读写作 A | 2 | 92 | A- | 4.0 |

- 3) **修改**某条成绩信息：可以对对应序号的信息条目进行修改（可修改姓名、课程名称或百分制成绩），然后显示完整的修改记录。其中，等级制成绩和绩点会随着百分制成绩的修改而修改。

输入：

5

输出：

5 李煜彤 电子电路与系统基础 (1) 2 79 B- 3.0

请重新输入该条目的学生姓名、课程名称、学分与百分制成绩，用空格隔开：

输入：

李煜彤 电子电路与系统基础 (1) 2 80

输出：

修改成功！

| 序号 | 姓名 | 课程名称 | 学分 | 百分制成绩 | 等级制成绩 | 绩点 |
|----|-----|---------------|----|-------|-------|-----|
| 5 | 李煜彤 | 电子电路与系统基础 (1) | 2 | 80 | B | 3.3 |

- 4) **查询**某个学生的全部成绩信息：输入学生的名字，可以查询该学生已录入的课程及成绩，同时计算该学生的均绩（GPA）。输出的信息条目按照序号排序。

其中，GPA 计算公式为：
$$GPA = \frac{\sum \text{课程学分} * \text{绩点}}{\sum \text{课程学分}}$$

输入要求：

李煜彤

输出结果：

| 序号 | 姓名 | 课程名称 | 学分 | 百分制成绩 | 等级制成绩 | 绩点 |
|----|-----|---------------|----|-------|-------|-----|
| 5 | 李煜彤 | 电子电路与系统基础 (1) | 2 | 79 | B- | 3.0 |
| 11 | 李煜彤 | 英语阅读写作 A | 2 | 92 | A- | 4.0 |

输入：

5

输出：

5 李煜彤 电子电路与系统基础 (1) 2 79 B- 3.0

请重新输入该条目的学生姓名、课程名称、学分与百分制成绩，用空格隔开：

输入：

李煜彤 电子电路与系统基础 (1) 2 80

输出：

修改成功！

| 序号 | 姓名 | 课程名称 | 学分 | 百分制成绩 | 等级制成绩 | 绩点 |
|----|-----|---------------|----|-------|-------|-----|
| 5 | 李煜彤 | 电子电路与系统基础 (1) | 2 | 80 | B | 3.3 |

- 5) 统计某门课程的全部成绩信息，查询结果按照序号升序排序，同时计算学生的平均百分制成绩。

输入：

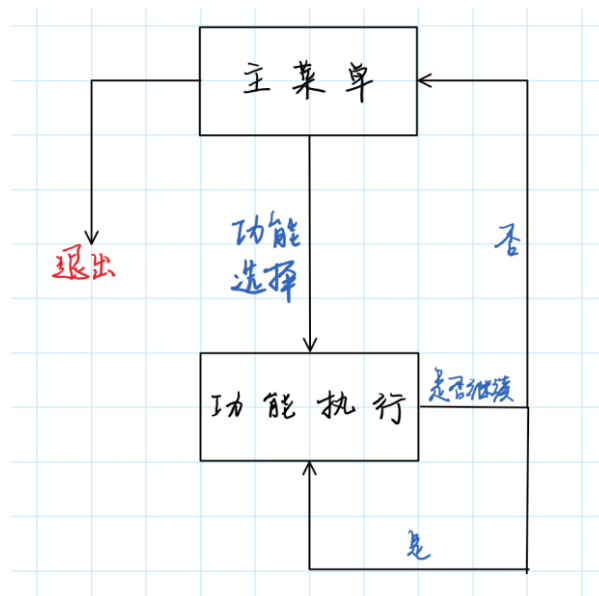
英语阅读写作 A

输出：

| 序号 | 姓名 | 课程名称 | 学分 | 百分制成绩 | 等级制成绩 | 绩点 |
|------------|-----|----------|----|-------|-------|-----|
| 10 | 余欣然 | 英语阅读写作 A | 2 | 100 | A+ | 4.0 |
| 11 | 李煜彤 | 英语阅读写作 A | 2 | 92 | A- | 4.0 |
| 平均百分制成绩：96 | | | | | | |

- 6) 系统以菜单方式工作，即用户可以自由选择所要执行的功能。

系统流程图：



使用者可以通过以上功能录入信息、修改信息、查询信息、整理统计出所了解的信息。除了要实现上述的基本功能之外，本系统还应该与用户形成互动，增加体验感。学生成绩管理系统有广大的用户群，是学校管理的必要工具，既能查看学生的情况，又能查看课程的情况。总之，该系统可以满足用户需求，实现对其管理的学生的成绩管理工作。

注：本人写了两个版本，原因会在后续“总结”说明。其中，在新版本中，输入时系统会依次进行提示，而不是一起输入。

二、系统总体设计

学生成绩管理系统的主要组成部分为信息条目，依次在文件中呈现。每个信息条目包含七个项目，分别为：序号、学生姓名、课程名称、课程学分数、学生该课程的百分制成绩、等级制成绩、绩点。其中，等级制成绩和绩点由百分制成绩决定。

例如：

11 李煜彤 英语阅读写作 A 2 92 A- 4.0

学生成绩管理系统以菜单的形式工作。在总菜单中，用户可以选择执行功能或退出管理系统。在执行完某个功能后，将会提示是继续执行该功能还是返回主菜单。

学生成绩管理系统包含五个主要功能，分别是：录入学生的成绩信息、查询某条成绩信息、修改某条成绩信息、查询某个学生的全部成绩信息、统计某门课程的全部成绩信息。

- **录入学生的成绩信息**

用户在系统的提示下，通过键盘进行信息的录入。

完成该功能后，系统会提示是否继续录入，用户可以选择继续录入或返回主菜单。

系统也会检查录入信息条目的百分制成绩是否在 1~100 的区间内，如不在，将会记录保留这条记录，但绩点会显示 0.0。如用户希望修改，可以通过返回主菜单-修改功能进行修改。

- **查询某条成绩信息**

用户在系统的提示下，通过键盘输入需要查询的信息条目序号，系统会检索文件并将该条目打印出来。

系统会检查录入信息条目的序号是否存在。若不存在，则会提示用户检查序号；若存在，则输出该信息条目。系统会提示是否继续查询，用户可以选择继续查询或返回主菜单。

- **修改某条成绩信息**

用户在系统的提示下，通过键盘输入需要修改的信息条目序号，系统会检索文件并将该条目打印出来。系统会提示是否修改，若用户选择“否”，则返回主菜单；若用户选择“是”，则会提示用户按照格式（见“一、系统需求分析”）输入，系统会检查录入信息的格式是否正确，若不正确，则会提示用户按照格式输入，并重新输入，若正确，则完成该功能。系统会提示是否继

续修改，用户可以选择继续修改或返回主菜单。

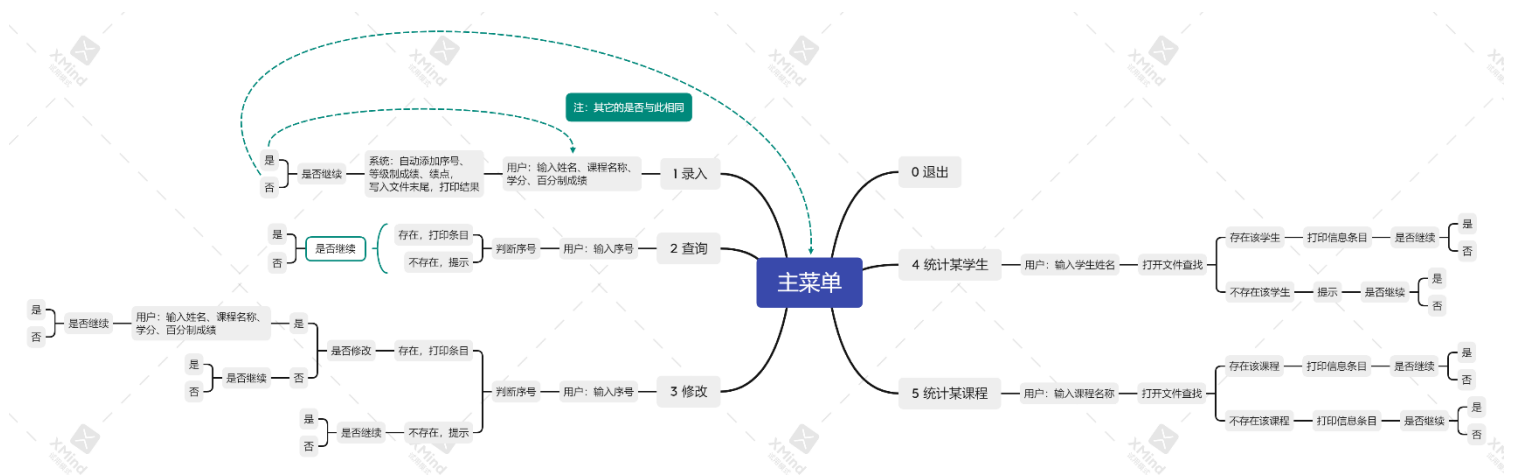
- 统计某个学生的全部成绩信息

用户在系统的提示下，通过键盘输入需要查询的学生姓名。系统会检查该学生是否存在。若不存在，则会提示用户检查学生姓名；若存在，则系统会检索文件，输出该学生的全部成绩信息及均绩。系统会提示是否继续查询，用户可以选择继续查询或返回主菜单。

- 统计某门课程的全部成绩信息

用户在系统的提示下，通过键盘输入需要查询的课程。系统会检查该课程是否存在。若不存在，则会提示用户检查课程名称；若存在，则系统会检索文件，输出该课程的全部成绩信息及平均百分制成绩。系统会提示是否继续查询，用户可以选择继续查询或返回主菜单。

系统功能模块图（输入输出的详细内容见“一、系统需求分析”）



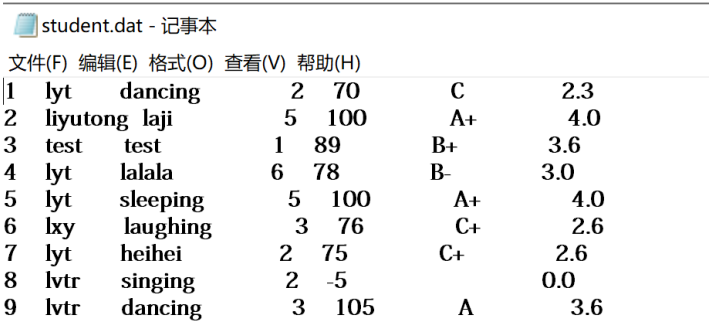
三、系统详细设计

- 数据文件

学生管理系统的文件默认为“student.dat”。其存放数据的形式为：

| 序号 | 姓名 | 课程名称 | 学分 | 百分制成绩 | 等级制成绩 | 绩点 |
|----|-----|----------|----|-------|-------|-----|
| 11 | 李煜彤 | 英语阅读写作 A | 2 | 92 | A- | 4.0 |

即如下图所示：



| student.dat - 记事本 | | | | | | |
|-------------------------------|----------|----------|---|-----|----|-----|
| 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H) | | | | | | |
| 1 | lyt | dancing | 2 | 70 | C | 2.3 |
| 2 | liyutong | laji | 5 | 100 | A+ | 4.0 |
| 3 | test | test | 1 | 89 | B+ | 3.6 |
| 4 | lyt | lalala | 6 | 78 | B- | 3.0 |
| 5 | lyt | sleeping | 5 | 100 | A+ | 4.0 |
| 6 | lxy | laughing | 3 | 76 | C+ | 2.6 |
| 7 | lyt | heihei | 2 | 75 | C+ | 2.6 |
| 8 | lvtr | singing | 2 | -5 | | 0.0 |
| 9 | lvtr | dancing | 3 | 105 | A | 3.6 |

使用者可以通过直接打开该文档查看已经录入的信息情况。

需要说明的是，每个信息条目的序号是系统自动分配的。当文件中不存在信息条目时，录入时会自动将其序号分配为 1；若文件中已经存在信息条目，则录入新的成绩信息时，会自动将其序号分配为文件最后一个信息条目序号+1。也就是说，只要使用者不擅自打开该文件进行修改，那么其序号排列是自然而规范的。功能 2 和功能 3，即查看和修改的功能，都是以序号为查找对象进行的。

- 界面设计

该成绩管理系统为我们课程一直使用的控制台应用程序，依靠打印一些字符并对其进行排列来提高界面的美观性。

主菜单界面：



退出时界面：



并且在系统提示语句前也会有空行，避免看串行，提高清晰性。

除了美观方面，系统在与用户交互的每一步都会有所提示，以提高其人性化。

- **类的层次设计**

成绩管理系统共有五个类，分别为：

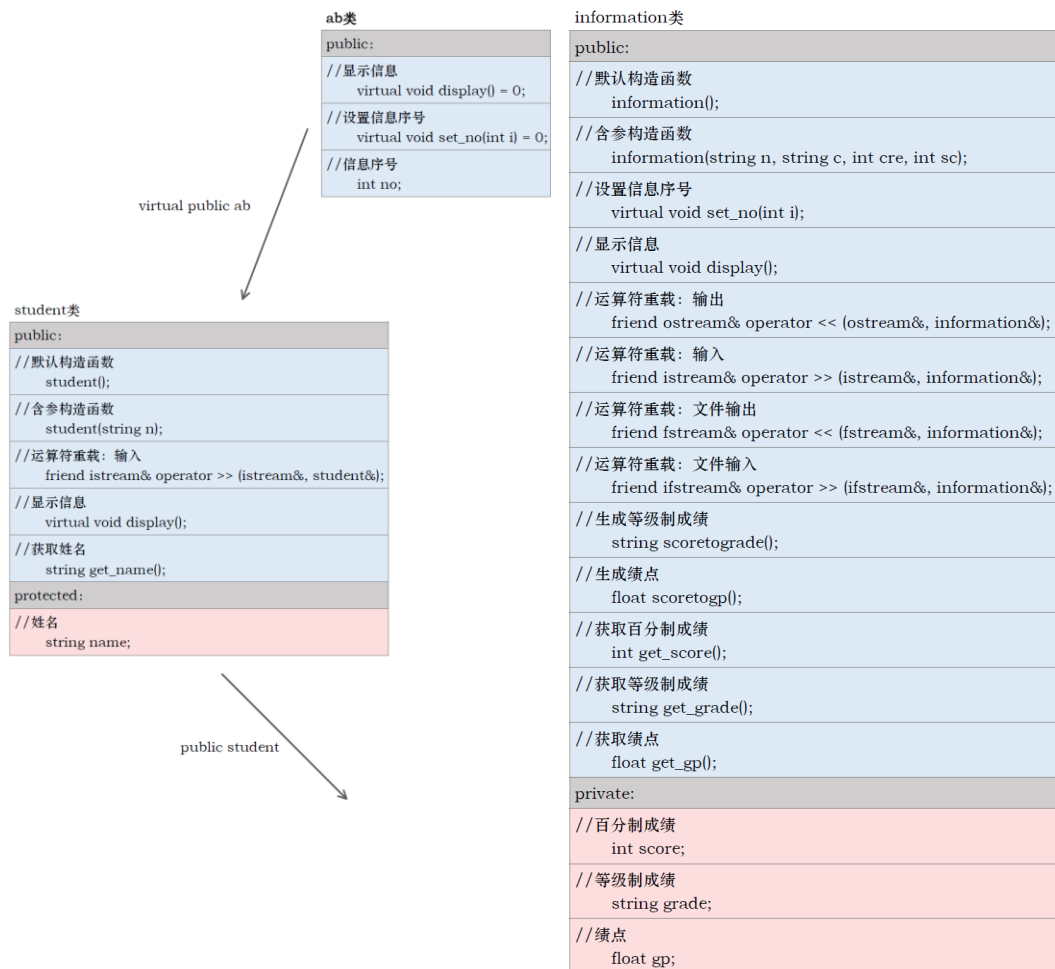
1. StudentManager 类，负责整个系统的运行
2. ab 类
3. student 类
4. curriculum 类
5. information 类

其层次图（包括其属性与行为）如下：

| | |
|----|--|
| 行为 | |
| 属性 | |

StudentManager类

| |
|---|
| public: |
| //构造函数 StudentManager(); |
| //展示主菜单：运算符重载 friend ostream& operator << (ostream&, StudentManager&); |
| //0 退出系统 void Exit_System(); |
| //1 录入学生成绩 void Add_Info(); |
| //2 查询某条成绩信息 void Search_Info(); |
| //3 修改某条成绩信息 void Modify_Info(); |
| //4 统计某个学生的全部成绩信息 void Count_Stu(); |
| //5 统计某门课程的全部成绩信息 void Count_Cur(); |
| //打印表头 void print_title(); |
| //从文件中得到已有个数 int learn_no(); |
| //保存文件 void save(); |
| //析构函数 ~StudentManager(); |
| //学生个数 int Student_Num; |
| //学生数组的指针 information * Student_Array; |



• 各功能模块实现

- ✓ Main 函数主要有两部分：①实例化 StudentManager，同时会完成打开文件、读取数据的操作；②循环：循环是整个程序的大架构，也就是让用户在主菜单和功能间往复循环，直到用户选择退出系统。
- ✓ 用户在选择完功能后，main 函数会进入一个小的循环，也就是在执行完依次该功能后，询问用户是否继续，如此实现批量操作。
- ✓ 每个功能下，只需调用 StudentManager 的一个成员函数。每个成员函数内会进行功能的实现。
- ✓ 具体能够实现的功能详见“二、系统总体设计”，具体的实现代码见“附录二：源程序清单”。

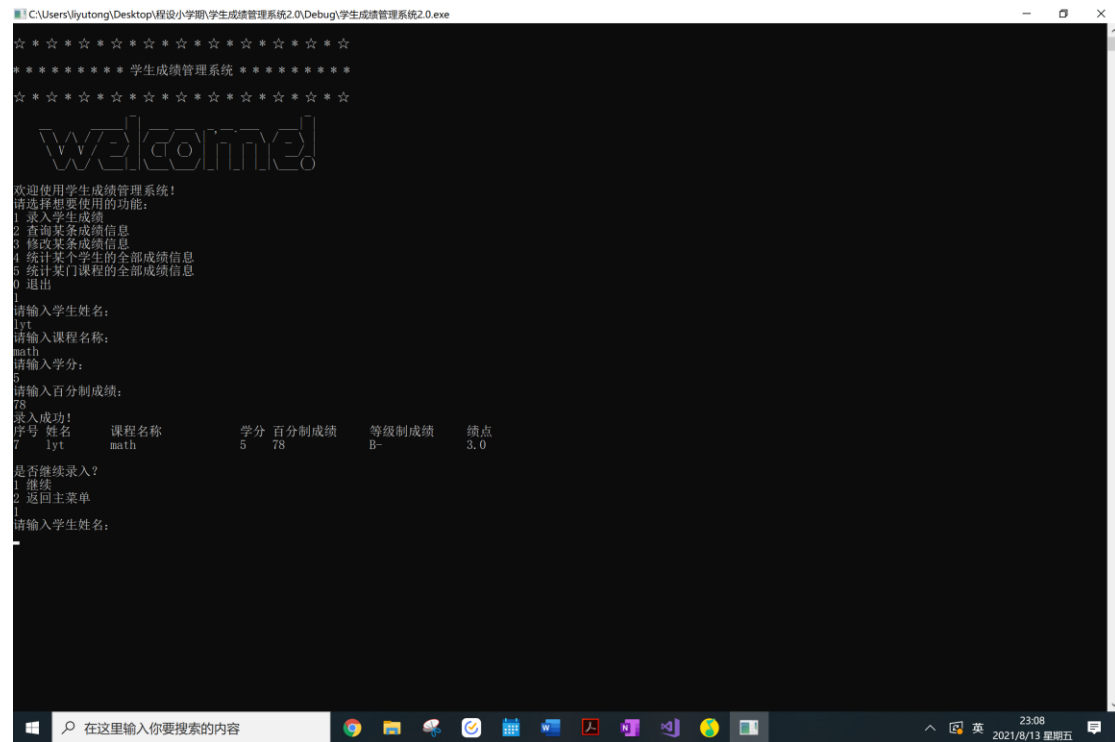
四、系统调试

- ✓ 异常： 写入访问权限冲突，XXX 是 nullptr
没有给 char* 分配空间，决定把 student 类里的字符串指针都变成字符数组。（是否动态分配空间是更优解？）
- ✓ warning C4305: 从 “double” 到 “float” 截断
把 return 的常量，如 3.6，写为 3.6f
- ✓ 错误 LNK2005 "int researchflag" (?researchflag@@3HA) 已经在 function.obj 中定义
原因：在头文件中含有全局变量的定义，cpp 文件中遇到又要定义一遍，所以是重定义
改正：在一个 cpp 文件中定义：int researchflag，在另一个 cpp 文件中使用：extern int researchflag
- ✓ 读取位置 0x00000000 时发生访问冲突
原因：操作了空指针

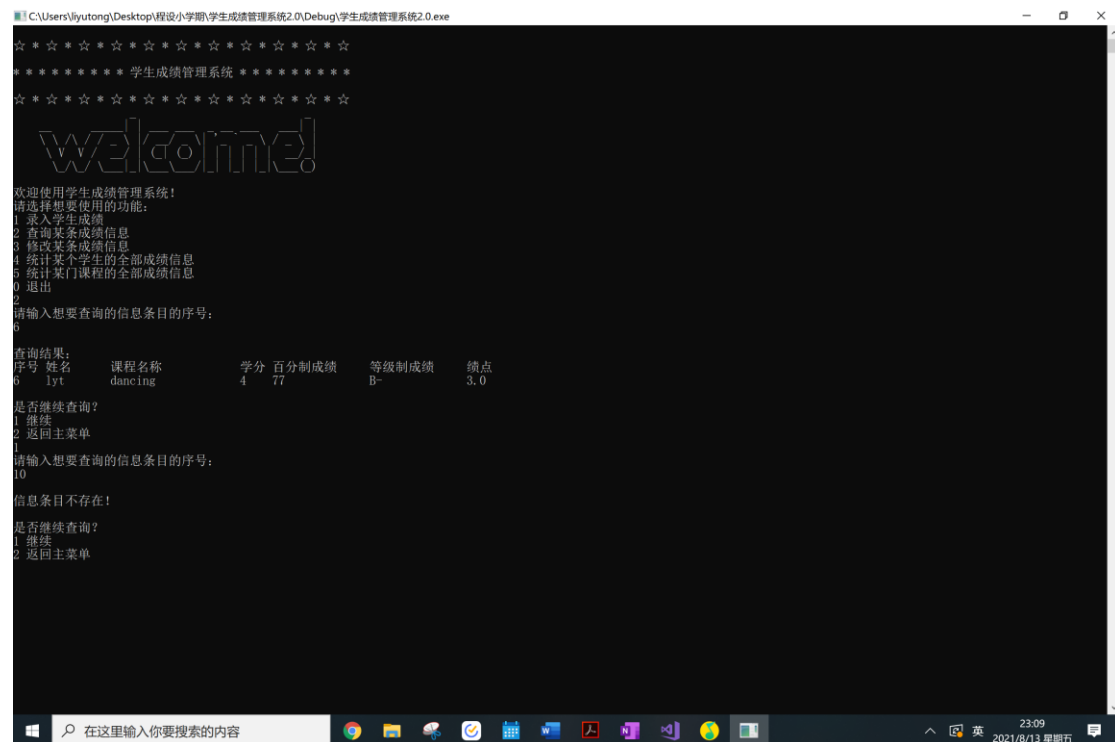
以上为一些 bug 节选，有一些比较小的很快就改正，并没有留下什么印象。更有一些不知所云的 bug，它说的是一回事，实际上可能问题出在远远的另一处。虽然碰见 bug 的时候甚是烦恼，但看到 bug 解决的一瞬间还是能感到发自内心的一种淳朴的快乐。

五、测试结果与分析

功能 1：录入成绩



功能 2：查询成绩



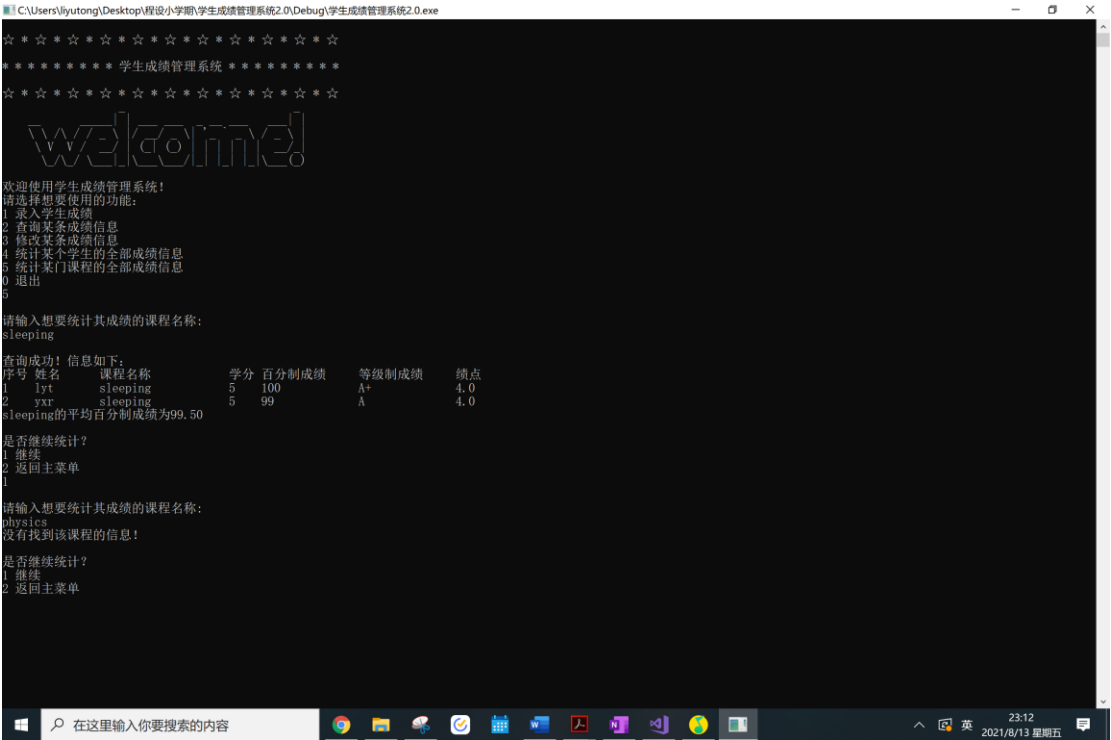
功能 3 修改成绩

```
C:\Users\liyutong\Desktop\课程设计小学期\学生成绩管理系统2.0\Debug\学生成绩管理系统2.0.exe
1 录入学生成绩
2 查询某条成绩信息
3 修改某条成绩信息
4 统计某个学生的全部成绩信息
5 统计某门课程的全部成绩信息
0 退出
3
请输入想要修改的信息条目的序号:
1
查询结果:
序号 姓名 课程名称 学分 百分制成绩 等级制成绩 绩点
1 lyt sleeping 5 100 A+ 4.0
是否修改?
1 是
2 否
2
是否继续修改?
1 继续
2 返回主菜单
1
请输入想要修改的信息条目的序号:
5
查询结果:
序号 姓名 课程名称 学分 百分制成绩 等级制成绩 绩点
5 yxr eating 5 33 F 0.0
是否修改?
1 是
2 否
1
请输入学生姓名:
yxr
请输入课程名称:
drinking
请输入学分:
5
请输入百分制成绩:
88
修改成功! 修改结果:
序号 姓名 课程名称 学分 百分制成绩 等级制成绩 绩点
5 yxr drinking 5 88 B+ 3.6
是否继续修改?
1 继续
2 返回主菜单
```

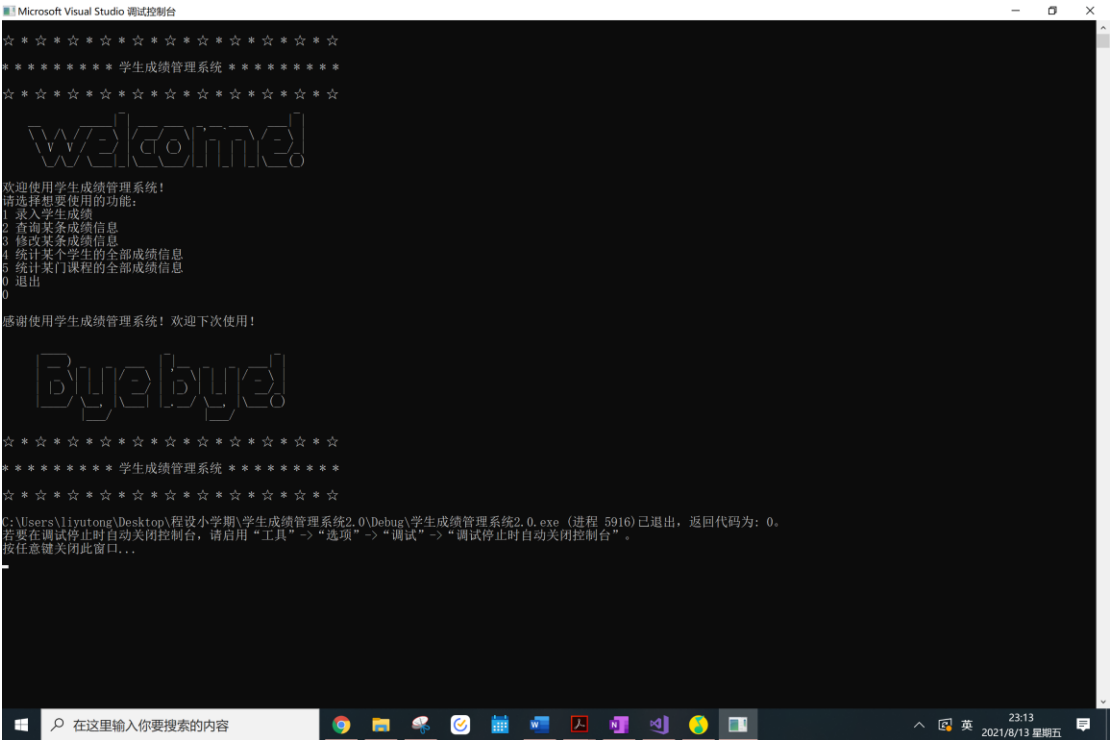
功能 4 统计学生成绩 (包括回答继续时输入不合法)

```
C:\Users\liyutong\Desktop\课程设计小学期\学生成绩管理系统2.0\Debug\学生成绩管理系统2.0.exe
*****
***** 学生成绩管理系统 *****
*****
*****
welcome!
欢迎使用学生成绩管理系统!
请选择想要使用的功能:
1 录入学生成绩
2 查询某条成绩信息
3 修改某条成绩信息
4 统计某个学生的全部成绩信息
5 统计某门课程的全部成绩信息
0 退出
4
请输入想要统计其成绩的学生姓名:
lyt
查询成功! 信息如下:
序号 姓名 课程名称 学分 百分制成绩 等级制成绩 绩点
1 lyt sleeping 5 100 A+ 4.0
4 lyt eating 6 80 B 3.3
6 lyt dancing 4 77 B- 3.0
7 lyt math 5 78 B- 3.0
lyt的gpa为3.34
是否继续统计?
1 继续
2 返回主菜单
1
请输入想要统计其成绩的学生姓名:
11lyt
没有找到该学生的信息!
是否继续统计?
1 继续
2 返回主菜单
2
输入不正确! 请输入1或2!
是否继续统计?
1 继续
2 返回主菜单
```

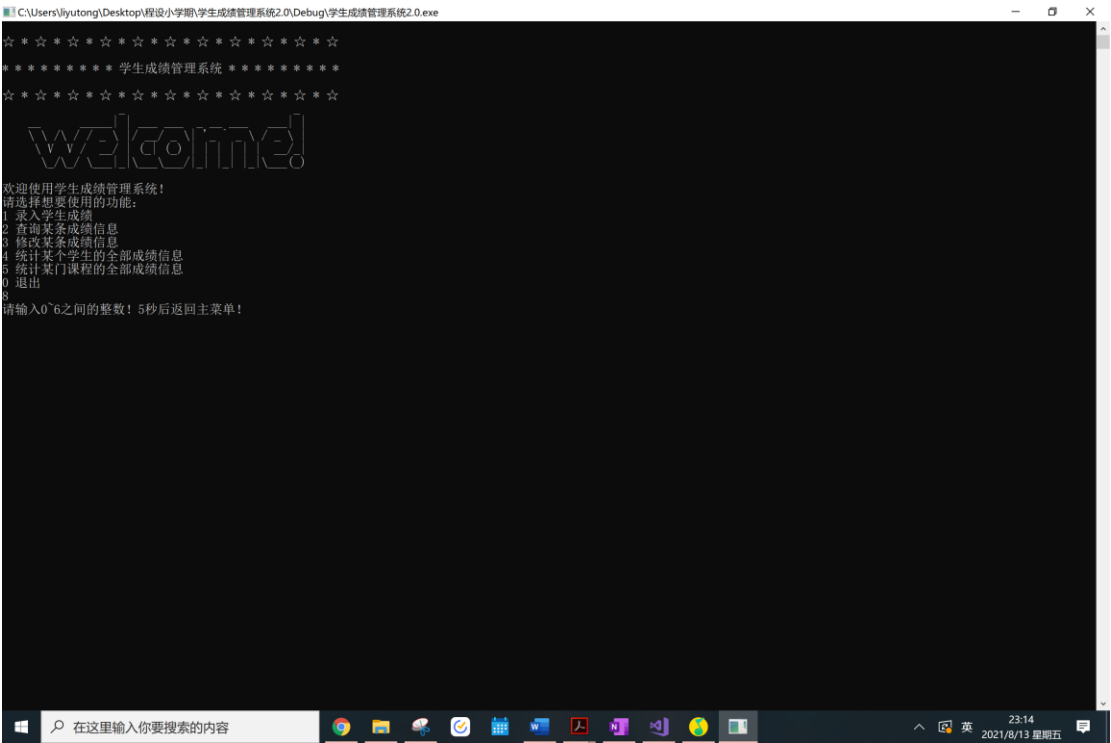
功能 5 统计课程成绩



功能 0 退出系统

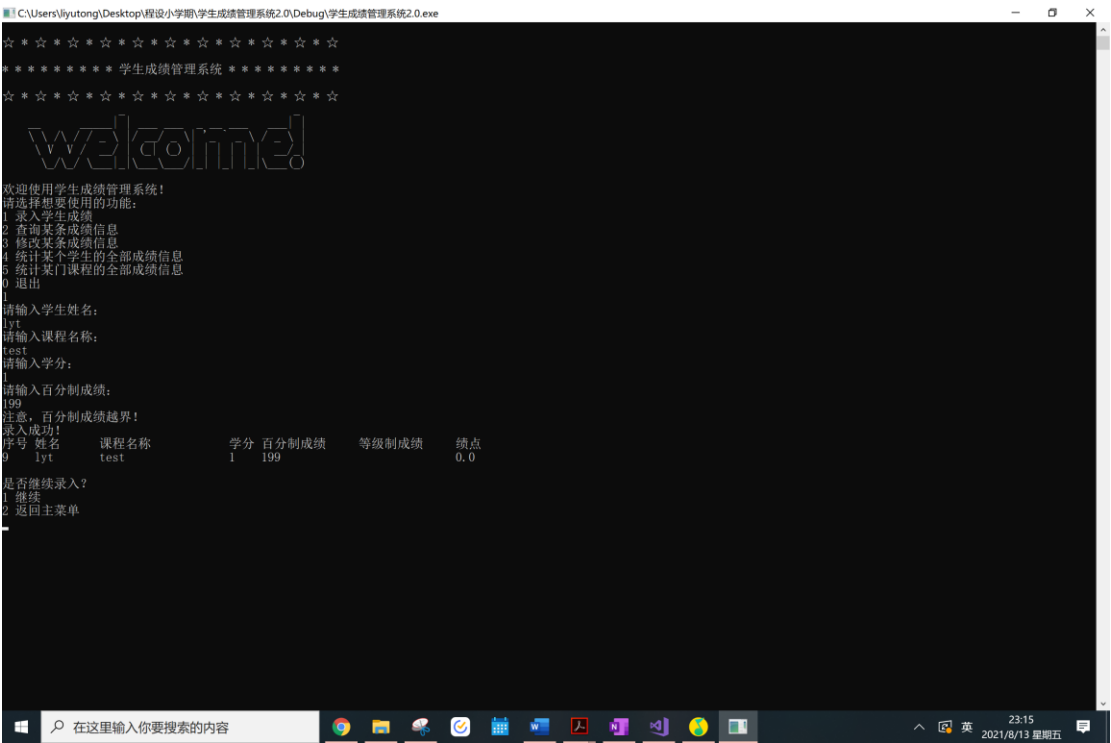


当主菜单输入不合法时:



5 秒后返回主菜单

录入成绩时，会有一个百分制成绩越界判断



六、总结

在第一遍写该程序时，我认为一个 student 类就可以完成这个作业的要求，不知道继承派生、虚函数抽象类之类的东西引入该程序的必要，左想右想想不太懂，就决定先用这么一个类写写看。写的时候就感觉有点在“凑”，一点一点碎片慢慢拼凑出一个程序，行内人大概叫“封装性”不好。

写完了之后，高兴归高兴，觉得至少有四个类的要求实在是难以达到，便找同学吐槽。这位同学怀疑我没有领会面向对象的编程思想，建议我去网上再学学。我并非谦虚，但也知道他大约说得对。后来到网上找到了 c++ 的讲解视频，不得不说非常清晰易懂，仿佛让我找到了中学上课的感觉，当时觉得许多匪夷所思的知识都有了灵性。顿时觉得脸红，当时上课不好好听，恐怕错过了许多，毕竟那种填鸭式进度下，因并非放弃听课的理由，腐朽的脑子也不懂得开发自学的新道路。并且也发现，xx 管理系统并非一家之创，更仿佛是 c++ 新手的必经之路，看了别人的代码，才能够为自己“锦”上添花。

于是开始写第二版。相比第一版有了许多的改进：

- ✓ 增加了类的数目，大大改善封装性

首先是 StudentManager 类，把这整个系统的操控封装成一个类是写第一版的我不曾想过的。这个类仿佛一只手，在背后操控着程序。这样就摆脱了零碎冗长的 main 函数。

其次有了类的层次，从抽象类 ab 派生出 student 类和 curriculum 类，information 类继承了 student 类和 curriculum 类。这样设计非说我是为了完成作业，也行。但是这样设计表示了一种关系：每条 information 都需要一个 student 和一个 curriculum 来决定。一个 student 会上许多 curriculum，一个 curriculum 也会有很多的 student，但是特定的 student 与一个特定的 curriculum 对应，才能产生一个确定的成绩。不知道如此想法是否是合格的。

- ✓ 文件交互的改善：利用数组

在原有程序中，每次进行一个功能都要打开一次程序，不太清晰。新版本中，最开始会将文件中已有数据读入数组中，之后只需要对数组进行操作，最后保存到文件中即可。

- ✓ 注释的丰富

在很多地方配备了很多详尽的注释，后续查看清晰，方便做工作。

- ✓ 一些其它的收获

比如，明白了 h 和 cpp 文件的关系，一些更规范的操作（初始化），main 函数中使用 switch 而不是一大堆的 if else 等等。

尽管有很多改进，但还是有很多不足的点，值得我反思、改进：

- ✓ 是否能保证，用户无论在哪个环节输入什么系统都不会退出，即异常处理的完备性。
- ✓ 对 const 理解不好，整个程序没有用到。

写完第二版我才感觉自己的 c++ 水平有些提升——第一个版本甚至不如上学期期末的水平。我觉得我还有一个收获比较宝贵——并不是那位同学说的面向对象的思想（这个我也有提升，但没那么醍醐灌顶）——编程不用那么简洁。之前的我写代码时，能不要的函数之类的都不要（甚至注释…?），但是我现在觉得不是。有的操作，虽然一时看去多余，但可能是更加规范的操作。

七、附录一：完成记录

说明：在小学期第四周，也就是上机时间，由于我有别的安排已经离开北京，该大作业完成情况较其余同学滞后。关于小学期上机请假，我于 7 月 19 日晚上向黄永峰老师提交邮件，但一直没有收到回复。

- ✓ 8/1 着手开始大作业，完成系统需求分析与系统总体设计，发现 c++ 知识遗忘得相当干净。
- ✓ 8/2 温习一半课件，写下一些代码。
- ✓ 8/3 温习完课件。完成大框架构建（主菜单），基本完成两个功能。
- ✓ 8/4 基本完成所有功能，优化主菜单界面。
- ✓ 8/5 完善主菜单及所有功能，更进本文档中的系统需求分析与系统总体设计。
- ✓ 8/6 完成一点系统详细设计，受到同学指点，明白自己未能领会 oop 精髓，决定先暂缓进度，多多学习，回过头来继续完善作业
- ✓ 8/10~8/11 学习了一点 c++ 的知识
- ✓ 8/12~8/13 完成“学生成绩管理系统 2.0”及该作业文档的编写

八、附录二：源程序清单

```
//ab.h
#pragma once
#include <iostream>
#include <string>
#include <iomanip>
#include <string>
#include <fstream>
using namespace std;

//抽象类（不需要 cpp 文件来实现）
class ab
{
public:
    //显示信息
    virtual void display() = 0;
    //设置信息序号
    virtual void set_no(int i) = 0;
    //信息序号
    int no;
};

//curriculum.h
#pragma once
#include "ab.h"

//课程类
class curriculum :virtual public ab
{
public:
    //默认构造函数
    curriculum();
    //含参构造函数
```

```

    curriculum(string cur,int c);
    //运算符重载: 输入
    friend istream& operator >> (istream&, curriculum&);
    //显示信息
    virtual void display();
    //获取课程名称
    string get_curname();
    //获取学分
    int get_credit();

protected:
    //课程名称
    string curname;
    //学分
    int credit;
};

```

```

//student.h
#pragma once
#include "ab.h"

//学生类
class student :virtual public ab
{
public:
    //默认构造函数
    student();
    //含参构造函数
    student(string n);
    //运算符重载: 输入
    friend istream& operator >> (istream&, student&);
    //显示信息
    virtual void display();
    //获取姓名
    string get_name();

```

```
protected:
```

```
    //姓名
```

```
    string name;
```

```
};
```

```
//information.h
```

```
#pragma once
```

```
#include"student.h"
```

```
#include"curriculum.h"
```

```
//信息类
```

```
class information :public student, public curriculum
```

```
{
```

```
public:
```

```
    //默认构造函数
```

```
    information();
```

```
    //含参构造函数
```

```
    information(string n, string c, int cre, int sc);
```

```
    //设置信息序号
```

```
    virtual void set_no(int i);
```

```
    //显示信息
```

```
    virtual void display();
```

```
    //运算符重载：输出
```

```
    friend ostream& operator << (ostream&, information&);
```

```
    //运算符重载：输入
```

```
    friend istream& operator >> (istream&, information&);
```

```
    //运算符重载：文件输出
```

```
    friend fstream& operator << (fstream&, information&);
```

```
    //运算符重载：文件输入
```

```
    friend ifstream& operator >> (ifstream&, information&);
```

```
    //生成等级制成绩
```

```
    string scoretograde();
```

```
    //生成绩点
```

```

float scoretoggp();
//获取百分制成绩
int get_score();
//获取等级制成绩
string get_grade();
//获取绩点
float get_gp();

private:
    //百分制成绩
    int score;
    //等级制成绩
    string grade;
    //绩点
    float gp;

};

//studentmanager.h
#pragma once
#include "information.h"

class StudentManager
{
public:
    //构造函数
    StudentManager();

    //展示主菜单：运算符重载
    friend ostream& operator << (ostream&, StudentManager&);

    //0 退出系统
    void Exit_System();

    //1 录入学生成绩

```

```
void Add_Info();

//2 查询某条成绩信息
void Search_Info();

//3 修改某条成绩信息
void Modify_Info();

//4 统计某个学生的全部成绩信息
void Count_Stu();

//5 统计某门课程的全部成绩信息
void Count_Cur();

//打印表头
void print_title();

//从文件中得到已有个数
int learn_no();

//保存文件
void save();

//析构函数
~StudentManager();

//学生个数
int Student_Num;

//学生数组的指针
information * Student_Array;

};
```



```
//student.cpp
#include "student.h"

//默认构造函数
student::student()
{

}

//含参构造函数
student::student(string n)
{
    name = n;
}

//运算符重载：输入
istream& operator >> (istream& input, student& s)
{
    string n;
    cout << "请输入学生姓名： " << endl;
    cin >> n;
    s.name = n;
    return input;
}

//显示信息
void student::display()
{
    cout << setiosflags(ios::left) << setw(10) << name;
}

//获取姓名
string student::get_name()
{
    return name;
}
```

```
}
```

```
//curriculum.cpp
```

```
#include"curriculum.h"
```

```
curriculum::curriculum()
```

```
{
```

```
}
```

```
curriculum::curriculum(string cur,int c)
```

```
{
```

```
    curname = cur;
```

```
    credit = c;
```

```
}
```

```
//运算符重载：输入
```

```
istream& operator >> (istream& input, curriculum& c)
```

```
{
```

```
    string n;
```

```
    int cre;
```

```
    cout << "请输入课程名称： " << endl;
```

```
    cin >> n;
```

```
    cout << "请输入学分： " << endl;
```

```
    cin >> cre;
```

```
    c.curname = n;
```

```
    c.credit = cre;
```

```
    return input;
```

```
}
```

```
//显示信息
```

```
void curriculum::display()
```

```
{
```

```
    cout << setw(20) << curname << setw(5) << credit;
```

```
}
```

```
//获取课程名称
string curriculum::get_curname()
{
    return curname;
}
```

```
//获取学分
int curriculum::get_credit()
{
    return credit;
}
```

```
//information.cpp
#include"information.h"
```

```
//默认构造函数
information::information()
{

}
```

```
//含参构造函数
information::information(string n, string c, int cre, int
sc):student(n),curriculum(c,cre)
{
    //learn_no();
    score = sc;
    scoretograde();
    gp = scoretogp();
}
```

```
//设置序号
void information::set_no(int i)
{
```

```

        no = i;
    }

//显示信息
void information::display()
{
    cout << setiosflags(ios::left) << setw(5) << no << setw(10) << name
score    << setw(20) << curname << setw(5) << credit << setw(15) <<
        << setw(15) << grade << setw(5) << setiosflags(ios::fixed)
        << setprecision(1) << gp << resetiosflags(ios::fixed) << endl;
}

//运算符重载：输出
ostream& operator << (ostream& output, information& s)
{
    output << setiosflags(ios::left) << setw(5) << s.no << setw(10) <<
s.name    << setw(20) << s.curname << setw(5) << s.credit << setw(15) <<
s.score    << setw(15) << s.grade << setw(5) << setiosflags(ios::fixed)
        << setprecision(1) << s.gp << resetiosflags(ios::fixed) << endl;
    return output;
}

//运算符重载：输入
istream& operator >> (istream& input, information& s)
{
    cout << "请输入学生姓名： " << endl;
    cin >> s.name;
    cout << "请输入课程名称： " << endl;
    cin >> s.curname;
    cout << "请输入学分： " << endl;
    cin >> s.credit;
    cout << "请输入百分制成绩： " << endl;

```

```

        cin >> s.score;
        s.scoretograd();
        s.gp = s.scoretogp();

        return input;
    }

```

//运算符重载：文件输出

```

fstream& operator << (fstream& output, information& s)
{
    output << setiosflags(ios::left) << setw(5) << s.no << setw(10) <<
s.name
    << setw(20) << s.curname << setw(5) << s.credit << setw(15) <<
s.score
    << setw(15) << s.grade << setw(5) << setiosflags(ios::fixed)
    << setprecision(1) << s.gp << resetiosflags(ios::fixed) << endl;
    return output;
}

```

//运算符重载：文件输入

```

ifstream& operator >> (ifstream& input, information& s)
{
    input >> s.no >> s.name >> s.curname >> s.credit >> s.score >>
s.grade >> s.gp;
    return input;
}

```

//生成等级制成绩

```

string information::scoretograd()
{
    if (score > 100)
        cout << "注意，百分制成绩越界！ " << endl;
    else if (score == 100)
        grade = "A+";
    else if (score >= 95)
        grade = "A";
}

```

```

else if (score >= 90)
    grade = "A-";
else if (score >= 85)
    grade = "B+";
else if (score >= 80)
    grade = "B";
else if (score >= 77)
    grade = "B-";
else if (score >= 73)
    grade = "C+";
else if (score >= 70)
    grade = "C";
else if (score >= 67)
    grade = "C-";
else if (score >= 63)
    grade = "D+";
else if (score >= 60)
    grade = "D";
else if (score >= 0)
    grade = "F";
else
{
    cout << "注意，百分制成绩越界！ " << endl;
    grade = "N/A";
}
return grade;
}

```

//生成绩点

```

float information::scoretogp()
{
    if (score > 100)
        return 0.0f;
    else if (score >= 90)
        return 4.0f;
}

```

```
        else if (score >= 85)
            return 3.6f;
        else if (score >= 80)
            return 3.3f;
        else if (score >= 77)
            return 3.0f;
        else if (score >= 73)
            return 2.6f;
        else if (score >= 70)
            return 2.3f;
        else if (score >= 67)
            return 2.0f;
        else if (score >= 63)
            return 1.6f;
        else if (score >= 60)
            return 1.3f;
        else
            return 0.0f;
    }
```

//获取百分制成绩

```
int information::get_score()
{
    return score;
}
```

//获取等级制成绩

```
string information::get_grade()
{
    return grade;
}
```

//获取绩点

```
float information::get_gp()
{
```

```

        return gp;
    }

//studentmanager.cpp
#include "studentmanager.h"
#include "information.h"

//构造函数
StudentManager::StudentManager()
{
    //文件不存在
    ifstream ifs;
    ifs.open("student.dat", ios::in);
    if (!ifs.is_open())
    {
        cout << "文件不存在! " << endl;
        //初始化学生个数
        Student_Num = 0;
        //初始化学生数组
        Student_Array = NULL;
        ifs.close();
        return;
    }

    //文件存在，数据为空
    char ch;
    ifs >> ch; //读走一个字符
    if (ifs.eof()) //若到结尾
    {
        cout << "文件为空! " << endl;
        //初始化学生个数
        Student_Num = 0;
        //初始化学生数组
        Student_Array = NULL;
        ifs.close();
    }
}

```


[illegible]


```

//将原空间数据拷贝到新空间
if (Student_Array != NULL)
{
    for (int i = 0;i < Student_Num - 1;i++)
    {
        newspace[i] = Student_Array[i];
    }
}

//录入数据
cin >> newspace[Student_Num - 1];
newspace[Student_Num - 1].set_no(Student_Num);

//释放原有空间
delete[]Student_Array;

//更新指向
Student_Array = newspace;

//录入成功提示
cout << "录入成功! " << endl;
print_title();
cout << Student_Array[Student_Num-1];

//保存到文件
save();
}

//2 查询某条成绩信息
void StudentManager::Search_Info()
{
    int goal;
    cout << "请输入想要查询的信息条目的序号: " << endl;
    cin >> goal;
}

```

```

        if (goal<0 || goal>Student_Num)
            cout << endl << "信息条目不存在! " << endl;
        else
        {
            cout << endl << "查询结果: " << endl;
            print_title();
            cout << Student_Array[goal - 1];
        }
    }

//3 修改某条成绩信息
void StudentManager::Modify_Info()
{
    int goal,choice=0;
    cout << "请输入想要修改的信息条目的序号: " << endl;
    cin >> goal;
    if (goal<0 || goal>Student_Num)
        cout << endl << "信息条目不存在! " << endl;
    else
    {
        cout << endl << "查询结果: " << endl;
        print_title();
        cout << Student_Array[goal - 1];

        while (choice != 1 && choice != 2)
        {
            cout << endl << "是否修改? " << endl << "1 是" << endl << "2
否" << endl;
            cin >> choice;
            if (choice == 1)
            {
                cin >> Student_Array[goal - 1];
                cout << endl << "修改成功! 修改结果: " << endl;
                print_title();
                cout << Student_Array[goal - 1];
            }
        }
    }
}

```

```

        save();
    }
    else if (choice != 2)
    {
        cout << "输入不正确! 请输入 1 或 2! " << endl;
    }
}
}
}
}
}

```

//4 统计某个学生的全部成绩信息

```

void StudentManager::Count_Stu()
{
    int creditsum = 0;
    float gpsum = 0;
    float gpa;
    string goal;
    int i, nameflag = 1;

    cout << endl << "请输入想要统计其成绩的学生姓名:" << endl;
    cin >> goal;

    for (i = 0; i < Student_Num; i++)
    {
        if (Student_Array[i].get_name() == goal)
        {
            if (nameflag)
            {
                nameflag = 0;
                cout << endl << "查询成功! 信息如下: " << endl;
                print_title();
            }
            creditsum = creditsum + Student_Array[i].get_credit();
            gpsum = gpsum + Student_Array[i].get_gp();

```

```

        cout << Student_Array[i];
    }
}
if (nameflag)
{
    cout << "没有找到该学生的信息! " << endl;
}
else
{
    gpa = gpsum / (float)creditsum;
    cout << goal << "的 gpa 为" << setiosflags(ios::fixed) <<
    setprecision(2) << gpa << endl;
}
}

```

//5 统计某门课程的全部成绩信息

```

void StudentManager::Count_Cur()
{
    int sum = 0;
    int scoresum = 0;
    float ave;
    string goal;
    int i, nameflag = 1;

    cout << endl << "请输入想要统计其成绩的课程名称:" << endl;
    cin >> goal;

    for (i = 0; i < Student_Num; i++)
    {
        if (Student_Array[i].get_curname() == goal)
        {
            if (nameflag)
            {
                nameflag = 0;
                cout << endl << "查询成功! 信息如下: " << endl;
            }
        }
    }
}

```

```

        print_title();
    }
    sum++;
    scoresum = scoresum + Student_Array[i].get_score();
    cout << Student_Array[i];
}
}
if (nameflag)
{
    cout << "没有找到该课程的信息! " << endl;
}
else
{
    ave = (float)scoresum / (float)sum;
    cout << goal << "的平均百分制成绩为" << setiosflags(ios::fixed) <<
    setprecision(2) << ave << endl;
}
}

```

//打印表头

```

void StudentManager::print_title()
{
    cout << setiosflags(ios::left) << setw(5) << "序号" << setw(10) << "姓名"
    << setw(20) << "课程名称" << setw(5) << "学分" << setw(15) << "百分制成绩"
    << setw(15) << "等级制成绩" << setw(5) << setiosflags(ios::fixed)
    << setprecision(1) << "绩点" << resetiosflags(ios::fixed) << endl;
}

```

//从文件中得到已有个数

```

int StudentManager::learn_no()
{
    int num;
    ifstream infile1("student.dat", ios::in);           //读取信息条目的序号
    if (!infile1)

```

```

{
    cerr << "open error!" << endl;
    exit(1);
}
char ch;
infile1 >> ch;//读走一个字符
if (infile1.eof())//若到结尾
    num = 0;
else
{
    infile1.seekg(-77, ios::end);
    infile1 >> num;
}
infile1.close();

return num;
}

//保存文件
void StudentManager::save()
{
    ofstream outfile;

    //打开文件
    outfile.open("student.dat", ios::out);
    //判断文件是否打开成功
    if (!outfile)
    {
        cerr << "open error!?" << endl;
        exit(1);
    }

    //将数据写入文件中
    for (int i=0;i < Student_Num;i++)
    {

```



```

        outfile << Student_Array[i];
    }

    //关闭文件
    outfile.close();
}

StudentManager::~StudentManager()
{
    //手动释放堆区数据
    if (Student_Array != NULL)
    {
        delete[] Student_Array;
        Student_Array = NULL;
    }
}

//学生成绩管理系统 2.0.cpp
#include "studentmanager.h"
#include <ctime>
using namespace std;

int main()
{
    //实例化
    StudentManager sm;
    int OPTION = 0, option = 0;
    while (true)
    {
        system("cls");
        cout << sm; //显示主菜单
        cin >> OPTION; //得到用户的选项
        switch (OPTION)
        {

```

```

case 0://0 退出系统
    sm.Exit_System();
    break;
case 1://1 录入学生成绩
    while (true)
    {
        option = 0;

        sm.Add_Info();

        while (option != 1 && option != 2)
        {
            cout << endl << "是否继续录入? " << endl << "1 继续"
<< endl << "2 返回主菜单" << endl;
            cin >> option;
            if (option != 1 && option != 2)
                cout << "输入不正确! 请输入 1 或 2! " << endl;
        }
        if (option == 1)
            continue;
        if (option == 2)
            break;
    }

    break;
case 2://2 查询某条成绩信息
    while (true)
    {
        option = 0;

        sm.Search_Info();

        while (option != 1 && option != 2)
        {
            cout << endl << "是否继续查询? " << endl << "1 继续"
<< endl << "2 返回主菜单" << endl;

```

```

        cin >> option;
        if (option != 1 && option != 2)
            cout << "输入不正确! 请输入 1 或 2! " << endl;
    }
    if (option == 1)
        continue;
    if (option == 2)
        break;
}
break;
case 3://3 修改某条成绩信息
    while (true)
    {
        option = 0;

        sm.Modify_Info();

        while (option != 1 && option != 2)
        {
            cout << endl << "是否继续修改? " << endl << "1 继续"
<< endl << "2 返回主菜单" << endl;
            cin >> option;
            if (option != 1 && option != 2)
                cout << "输入不正确! 请输入 1 或 2! " << endl;
        }
        if (option == 1)
            continue;
        if (option == 2)
            break;
    }
    break;
case 4://4 统计某个学生的全部成绩信息
    while (true)
    {
        option = 0;

```

```

        sm.Count_Stu();

        while (option != 1 && option != 2)
        {
            cout << endl << "是否继续统计? " << endl << "1 继续"
<< endl << "2 返回主菜单" << endl;
            cin >> option;
            if (option != 1 && option != 2)
                cout << "输入不正确! 请输入 1 或 2! " << endl;
        }
        if (option == 1)
            continue;
        if (option == 2)
            break;
    }
    break;
case 5://5 统计某门课程的全部成绩信息
    while (true)
    {
        option = 0;

        sm.Count_Cur();

        while (option != 1 && option != 2)
        {
            cout << endl << "是否继续统计? " << endl << "1 继续"
<< endl << "2 返回主菜单" << endl;
            cin >> option;
            if (option != 1 && option != 2)
                cout << "输入不正确! 请输入 1 或 2! " << endl;
        }
        if (option == 1)
            continue;
        if (option == 2)
            break;
    }
}

```

```
        break;
default:
{
    cout << "请输入 0~6 之间的整数! 5 秒后返回主菜单! " << endl;
    double pretime = clock();
    while (true)
    {
        if (clock() - pretime > 5000)
            break;
    }
    break;
}

}

}

return 0;
}
```

九、附录三：评分表

| 项目 | 评价 | |
|--------------|----|--|
| 设计方案的合理性与创新性 | 6 | |
| 设计与调试结果 | 8 | |
| 设计说明书的质量 | 2 | |
| 程序基本要求涵盖状况 | 8 | |
| 程序代码编写素养情况 | 4 | |
| 课程设计周表现情况 | 2 | |
| 综合成绩 | 30 | |