

《学生成绩管理系统》

一、任务描述

设计一个具有网络访问功能的学生成绩管理系统，要求实现学生和老师两种用户对课程及其成绩的操作和信息查看

二、需求分析和功能说明

- 1.现两种用户角色：学生、教师。
- 2.实现两种课程：必修课、选修课
- 3.实现登录和注销功能。
- 4.实现选修课的选课功能，仅学生可用
- 5.实现课程的查看功能，学生和教师的功能有所不同
- 6.实现课程成绩的录入、修改、提交操作，仅老师可用
- 7.实现课程成绩和绩点的计算和查看功能，仅学生可用
- 8.教师数不少于3人，学生数不少于4人，必修课和选修课各自不少于3门。每个必修课的学生不少于2人，每个学生必修课和选修课均不少于两门
- 9.实现所有数据的文件存取功能，即重启程序以后可以访问上次使用后的数据。文件存取操作通过重载运算符">>"和"<<"来完成，有异常处理
- 10.实现排序功能，教师和学生的功能有所不同。排序采用泛型算法老师查看任门课程成绩时，可按学生学号、成绩升序或降序查看 学生查看自己的课程时，可按成绩、绩点升序或降序查看
- 11.实现程序的网络访问功能

三、概要设计

1.开发环境

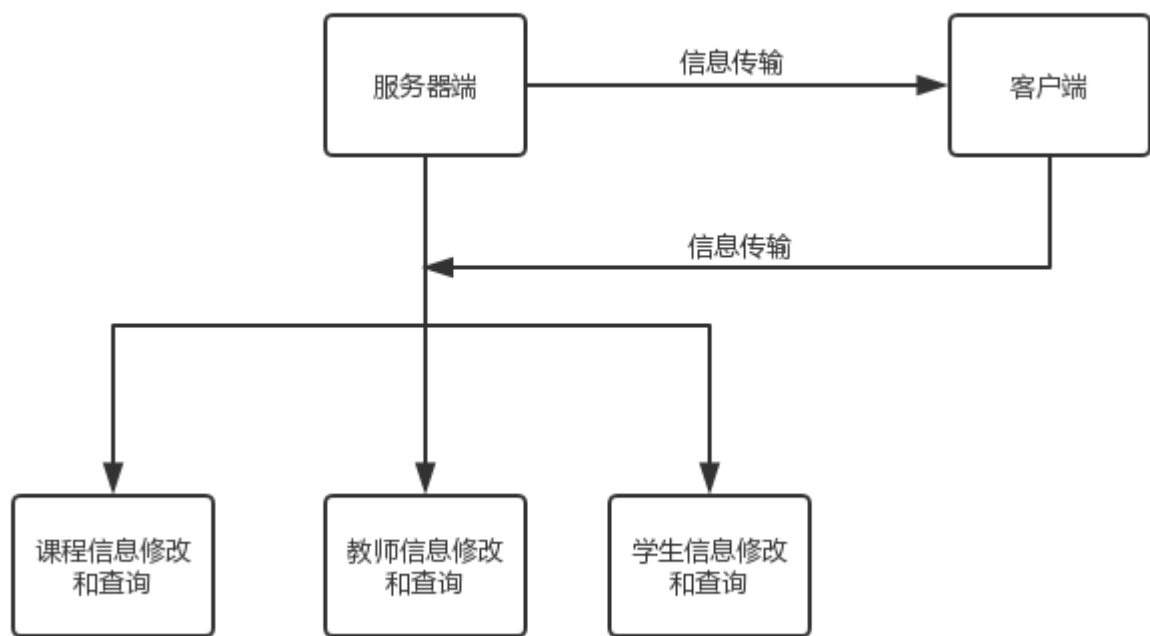
1.1软件环境

Visual C++2017

1.2语言

C++

2.总体结构（流程图）



3.模块划分

服务器端部分	客户端部分
服务器函数main.cpp	客户端主函数client.cpp
教师操作部分 teacher.cpp	
学生操作部分 student.cpp	
课程信息操作部分 course.cpp	

4.数据结构和字典

```
Class Course{//课程信息，基类

Class Bixiu: public Course{//必修课程信息，子类

Class Xuanxiu: public Course{//选修课程信息，子类

Class Student{//学生信息类

Class Teacher{//教师信息类

vector<Student>student;//内存中读取的学生信息

vector<Teacher>teacher;//内存中读取的教师信息

vector<Course>course;// 内存中读取的课程信息
```

```
Student now_student;//当前操作的学生的信息

Teacher now_teacher;//当前操作的教师的信息
```

5.各模块设计说明

5.1服务器主函数

main.cpp

```
//服务器主函数，等待客户端连接，作各种信息初始化
int main(int argc, char* argv[]);
```

5.2教师操作部分

Teacher.cpp

```
void Teacher::teacher_information(Teacher zancun);//打印教师信息
void print_course_list2(Course zancun); //打印课程信息
void Teacher::set_student_grade(Student zancun, Course zancun2);//录入指定选课的指定学生成绩
void Teacher::reset_student_grade(Student zancun, Course zancun2);//修改指定选课的指定学生成绩
void Teacher::logon_student_grade(Course zancun2)//提交本课程学生成绩
void Teacher::for_course_set();//成绩操作部分
void Teacher::show_all_course();//查看所有课程
void Teacher::show_teacher_course();//显示自己的所有课程
//某课程学生名单排序输出(四种方式)
bool num_increase(const Student &v1, const Student &v2);
bool num_decrease(const Student &v1, const Student &v2);
bool grade_increase(const Student &v1, const Student &v2);
bool grade_decrease(const Student &v1, const Student &v2);
void Teacher::show_specify_student_information(int nnum);//查看指定选课学生信息,nnum为课程编号
void Teacher::show_course_studentlist();//显示自己课程的所有选课学生名单
void Teacher::for_course_show();//课程查看部分
void Teacher::show_teacher_choice();//教师操作选择
void Teacher::teacher_register(string a);//教师注册
void Teacher::teacher_login();//教师登录
void initial_teacherlist();//内置教师信息打印
istream & operator >> (istream &in, Teacher &c); //文件教师信息流式读入
ostream& operator << (ostream &out, const Teacher &c); //文件教师信息流式输出
void teacher_save(); //教师信息保存
void initial_teacher();//教师信息初始化操作
```

5.3学生操作部分

student.cpp

```
void Student::student_information(Student zancun); //显示学生信息
void print_course_list(Course zancun); //打印课程信息
void Student::course_choose();//选课
```

```

void Student::show_allcan_course(); //查看所有可选课程(选修)
void Student::course_delete(); //删除选课
void Student::show_student_course(); //查看已选课程(选修)
void Student::for_course_choose(); //学生选课部分
//学生成绩排序输出(四种方式)
bool sgrade_increase(const Course &v1, const Course &v2)
bool sgrade_decrease(const Course &v1, const Course &v2)
bool sgrade_point_increase(const Course &v1, const Course &v2)
bool sgrade_point_decrease(const Course &v1, const Course &v2)
void Student::show_own_course(); //打印学生个人课程信息
void Student::show_all_course(); //打印所有程信息
void Student::for_course_show(); //学生的课程查看操作
void Student::show_student_grade(); //学生成绩查看
void Student::show_student_choice(); //学生操作选择
void Student::student_register(string a); //学生注册
void Student::student_login(); //学生登录
void initial_studentlist(); //内置学生信息打印
istream & operator >> (istream &in, Student &c); //文件学生信息流式读入
ostream & operator << (ostream &out, const Student &c); //文件学生信息流式输出
void student_save(); //学生信息保存
void initial_student(); //学生信息初始化

```

5.4课程信息操作部分

course.cpp

```

void initial_courselist(); //内置课程生信息打印
istream & operator >> (istream &in, Course &c); //文件课程信息流式读入
ostream & operator << (ostream &out, const Course &c); //文件课程信息流式输出
void course_save(); //课程信息保存
int coursemain(); //课程信息初始化

```

5.5客户端主函数

client.cpp

```

//客户端主函数，连接服务器端，作操作界面
int main();

```

四、详细设计

1.定义的数据结构

1.1课程信息类

```

class Course{//非多态类型
public:
    Course() {

```

```

}
~Course() {

}
int kind;//课程属性类型---1for必修; 2for选修
int NO;//课程编号
int point;//学分
string couname ;//课程名字
string teacher_num = "-1";//本课程教师工号

int putin_key = 0;//本课程成绩是否提交的判断信号
int grade = -1;//成绩
double grade_point = -1;//绩点

vector<Student>stu_list;//本课程学生名单及成绩信息

virtual double find_garde(Course a,int sum) const {
    return -1;
};//绩点计算

friend ostream & operator << (ostream &out, const Course &c);
friend istream & operator >> (istream &in, Course &c);
};

```

```

class Bixiu :public Course{//多态类型
public:

    virtual double find_garde(Course a,int sum) const {;//绩点计算
        a.grade_point = ((double)a.grade/ sum)*a.point;
        return a.grade_point;
    }

    int kind;//课程属性类型---1for必修; 2for选修
    int NO;//课程编号
    int point;//学分
    string couname;//课程名字

    int grade = -1;//成绩
    double grade_point = -1;//绩点

    int putin_key = 0;//本课程成绩是否提交的判断信号

    string teacher_num= "-1";//本课程教师工号
    vector<Student>stu_list;//本课程学生名单及成绩信息

    friend ostream & operator << (ostream &out, const Course &c);
    friend istream & operator >> (istream &in, Course &c);
};

```

```

class Xuanxiu :public Course{//多态类型

```

```

public:
    int kind; //课程属性类型---1for必修; 2for选修
    int NO; //课程编号
    int point; //学分
    string couname; //课程名字

    int grade = -1; //成绩
    double grade_point = -1; //绩点

    int putin_key = 0; //本课程成绩是否提交的判断信号

    string teacher_num = "-1"; //本课程教师工号
    vector<Student> stu_list; //本课程学生名单及成绩信息

    virtual double find_garde(Course a, int sum) const {
        //绩点计算
        a.grade_point = (double)((sqrt(a.grade) / 10) * (a.point));
        return a.grade_point;
    }
    friend ostream & operator << (ostream &out, const Course &c);
    friend istream & operator >> (istream &in, Course &c);

};

```

1.2 学生信息类

```

class Student
{
public:
    Student() {
        vector<Course> stu_grade; //学生成绩信息数组
        string stu_num; //记录学生学号
        string stu_name; //记录学生姓名
        string stu_college; //记录学生学院
        string stu_class; //记录学生班级
        vector<Course> stu_course; //记录学生自己的课程
        vector<Course> can_course_list; //存储可选课程
    }
    ~Student() {

    }

    void student_register(string a); //学生注册
    void student_login(); //学生登录
    void student_information(Student zancun); //显示学生信息

    void show_student_choice(); //显示学生可以进行的功能选择
    //void choice(int choice2); //对学生的选择进行相应操作

    void for_course_choose(); //选课部分
    void show_allcan_course(); //查看所有可选课程
    void course_choose(); //选课
    void show_student_course(); //查看已选课程

```

```

void course_delete();//删除课程

void for_course_show();//查看课程部分
void show_all_course();//查看所有课程
void show_own_course();//查看自己的课程

void show_student_grade();//查询学生成绩

//private:
vector<Course>stu_grade;//学生成绩信息数组
string stu_num;//记录学生学号
string stu_name ;//记录学生姓名
string stu_college ;//记录学生学院
string stu_class ;//记录学生班级

vector<Course>stu_course;//记录学生自己的课程
vector<Course>can_course_list;//存储可选课程
friend ostream & operator << (ostream &out, const Student &c);
friend istream & operator >> (istream &in, Student &c);
};

```

1.3教师信息类

```

class Teacher
{
public:
    Teacher() {

    }
    ~Teacher() {

    }

    void teacher_register(string a);//教师注册
    void teacher_login();//教师登录
    void teacher_information(Teacher zancun);//显示教师信息

    void show_teacher_choice();//显示教师可以进行的功能选择
    //void choice(int choice1);//对教师的选择进行相应操作

    void for_course_show();//课程查看部分
    void show_all_course();//查看所有课程 （区分必修，选修
    void show_teacher_course();//显示自己的所有课程（区分必修，选修
    void show_course_studentlist();//显示课程的所有选课学生名单
    void show_specify_student_information(int nnum);//显示指定选课学生信息（包括成绩

    void for_course_set();//成绩操作部分
    void set_student_grade(Student zancun, Course zancun2);//录入学生成绩
    void reset_student_grade(Student zancun, Course zancun2);//修改学生成绩
    void logon_student_grade(Course zancun2);//提交学生成绩

//private:

```

```
//vector<get_grade>gather;//学生成绩存储数组
//int student_count;//记录学生人数
//int teacher_count;//记录教师人数
string tea_num;//记录教师工号
string tea_name;//记录教师姓名
string tea_collage;//记录教师学院
vector<Course>tea_course;//记录老师自己的课程

friend ostream & operator << (ostream &out, const Teacher &c);
friend istream & operator >> (istream &in, Teacher &c);
};
```

见附件一、二、三版本

六、测试

1.运行服务器端

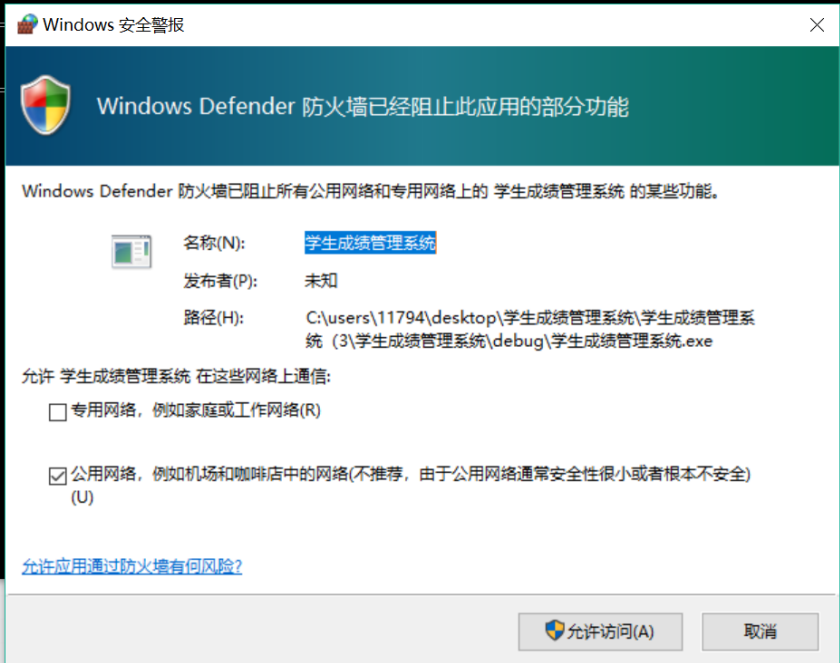
点击允许网络访问

**内置课程信息如下:

**内置教师信息如下:

**内置学生信息如下:

等待连接...



2.运行客户端

C:\Users\11794\Desktop\学生成绩管理系统\学生成绩管理系统 (3)\学生成绩管理系统\Debug\学生成绩管理系统.exe

等待连接...

接受到一个连接: 127. 0. 0. 1

C:\Users\11794\Desktop\学生成绩管理系统\学生成绩管理系统 (3)\学生成绩管理系统-客户端\Debug\学生成绩管理系统-客户端.exe

|| 欢迎进入学生成绩管理系统 ||

—请选择用户类型: 1. 学生 2. 教师 shut_down. 退出

3.两种用户角色

学生、教师:

**内置教师信息如下:

```
=====
教师工号: 1111
教师姓名: KSJ
教师学院: scs
教师课程: 1-Math 2-English 3-EE
=====
```

```
=====
教师工号: 2222
教师姓名: KNJ
教师学院: scs
教师课程: 4-CS 5-Music 6-PE
=====
```

```
=====
教师工号: 3333
教师姓名: MYK
教师学院: scs
教师课程: 7-Chinese 8-Dance
=====
```

```
=====
教师工号: 4444
教师姓名: FSH
教师学院: scs
教师课程: 9-Spanish
=====
```

```
**内置学生信息如下:
=====
学生学号: 1234
学生姓名: JHS
学生学院: scs
学生课程: 4-CS 1-Math 2-English 3-EE 5-Music
=====
学生学号: 4321
学生姓名: PJM
学生学院: scs
学生课程: 1-Math 2-English 3-EE 4-CS
=====
学生学号: 5678
学生姓名: KTH
学生学院: scs
学生课程: 4-CS 1-Math 2-English 3-EE 5-Music
=====
学生学号: 8765
学生姓名: JJK
学生学院: scs
学生课程: 1-Math 2-English 3-EE 4-CS
=====
```

4.实现两种课程

必修课、选修课

```

=====
**内置课程信息如下:
=====
课程编号: 1
课程名称: Math
课程教师编号: 1111
课程类型: 必修课
课程学分: 4
=====
课程编号: 2
课程名称: English
课程教师编号: 1111
课程类型: 必修课
课程学分: 5
=====
课程编号: 3
课程名称: EE
课程教师编号: 1111
课程类型: 必修课
课程学分: 5
=====
课程编号: 4
课程名称: CS
课程教师编号: 2222
课程类型: 必修课
课程学分: 5
=====
课程编号: 5
课程名称: Music
课程教师编号: 2222
课程类型: 选修课
课程学分: 2
=====
课程编号: 6
课程名称: PE
课程教师编号: 2222
课程类型: 选修课
课程学分: 1
=====
课程编号: 7
课程名称: Chinese
课程教师编号: 3333
课程类型: 选修课
课程学分: 2
=====
课程编号: 8
课程名称: Dance
课程教师编号: 3333
课程类型: 选修课
课程学分: 1
=====
课程编号: 9
课程名称: Spanish
课程教师编号: 4444
课程类型: 选修课
课程学分: 3
=====

```

5.实现登录和注销功能

```
--请选择用户类型：1. 学生    2. 教师    shut_down. 退出
1
--您已选择了学生身份，请输入学号：1234
姓名：JHS
学院：scs
班级：11
学号：1234
=====
--请选择将要进行的操作：1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录
4
|| 您已注销登录 ||
```

6.实现选修课的选课和删除选课功能

(仅学生可用)

6.1课程选择及结果查询

```

--请选择将要进行的操作: 1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录
1
--请选择要进行的操作: 1. 查看所有可选课程 2. 查看所有已选课程
1
课程编号: 6
课程名称: PE
课程类型: 选修课
课程学分: 1
=====
课程编号: 7
课程名称: Chinese
课程类型: 选修课
课程学分: 2
=====
课程编号: 8
课程名称: Dance
课程类型: 选修课
课程学分: 1
=====
课程编号: 9
课程名称: Spanish
课程类型: 选修课
课程学分: 3
=====
--请输入你想选择的课程编号
6
--请确认你将选择此课程:1 for 确定; else for 取消
1
--请选择将要进行的操作: 1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录
=====
选修课程:
课程编号: 5
课程名称: Music
课程类型: 选修课
课程学分: 2
=====
课程编号: 6
课程名称: PE
课程类型: 选修课
课程学分: 1
=====

```

6.2课程删除及结果查询

```

--请选择将要进行的操作：1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录
1
--请选择要进行的操作：1. 查看所有可选课程 2. 查看所有已选课程
2
=====
课程编号：5
课程名称：Music
课程类型：选修课
课程学分：2
=====
=====
课程编号：6
课程名称：PE
课程类型：选修课
课程学分：1
=====
--是否要删除已选课程 1 for选择删除;else for不删除
1
--请输入想要删除的选修课编号:6
你将要删去的课程是：
=====
课程编号：6
课程名称：PE
课程类型：选修课
课程学分：1
=====
--请确认是否删去本课程：1for确认，； else for 取消
1
--是否继续删除 1for继续； else for 退出
0
--请选择将要进行的操作：1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录
2
--请选择要进行的操作：1. 查看自己的课程 2. 查看所有课程
1
=====
选修课程：
课程编号：5
课程名称：Music
课程类型：选修课
课程学分：2
=====

```

7.实现课程的查看功能，学生和教师的功能有所不同

7.1学生课程查询操作

--请选择将要进行的操作: 1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录

2

--请选择要进行的操作: 1. 查看自己的课程 2. 查看所有课程

1

必修课程:

课程编号: 4

课程名称: CS

课程类型: 必修课

课程学分: 5

=====

课程编号: 1

课程名称: Math

课程类型: 必修课

课程学分: 4

=====

课程编号: 2

课程名称: English

课程类型: 必修课

课程学分: 5

=====

课程编号: 3

课程名称: EE

课程类型: 必修课

课程学分: 5

=====

选修课程:

课程编号: 5

课程名称: Music

课程类型: 选修课

课程学分: 2

=====

2

必修课程:

课程编号: 1

课程名称: Math

课程类型: 必修课

课程学分: 4

=====

课程编号: 2

课程名称: English

课程类型: 必修课

课程学分: 5

=====

课程编号: 3

课程名称: EE

课程类型: 必修课

课程学分: 5

=====

课程编号: 6

课程名称: PE

课程类型: 选修课

课程学分: 1

=====

课程编号: 7

课程名称: Chinese

课程类型: 选修课

课程学分: 2

=====

课程编号: 8

=====	=====
课程编号: 4	课程名称: Dance
课程名称: CS	课程类型: 选修课
课程类型: 必修课	课程学分: 1
课程学分: 5	=====
=====	课程编号: 9
选修课程:	课程名称: Spanish
课程编号: 5	课程类型: 选修课
课程名称: Music	课程学分: 3
课程类型: 选修课	=====
课程学分: 2	
=====	
课程编号: 6	
课程名称: PE	

7.2教师课程查询操作:

7.2.1查看自己教授的课程

```

--请选择将要进行的操作: 1. 课程查看 2. 成绩操作 3. 注销登录
1
--请选择要进行的操作: 1. 查看所有课程 2. 查看自己教授的课程 3. 查看所有选课学生名单
2
    必修课程:
课程编号: 1
课程名称: Math
课程类型: 1
课程学分: 4
=====
课程编号: 2
课程名称: English
课程类型: 1
课程学分: 5
=====
课程编号: 3
课程名称: EE
课程类型: 1
课程学分: 5
=====
    选修课程:
--请选择将要进行的操作: 1. 课程查看 2. 成绩操作 3. 注销登录

```

7.2.2查看课程选课学生名单

```

--请选择将要进行的操作：1. 课程查看 2. 成绩操作 3. 注销登录
1
--请选择要进行的操作：1. 查看所有课程 2. 查看自己教授的课程 3. 查看所有选课学生名单
3
课程：1Math
    学号：1234      姓名：JHS
    学号：4321      姓名：PJM
    学号：5678      姓名：KTH
    学号：8765      姓名：JJK
课程：2English
    学号：1234      姓名：JHS
    学号：4321      姓名：PJM
    学号：5678      姓名：KTH
    学号：8765      姓名：JJK
课程：3EE
    学号：1234      姓名：JHS
    学号：4321      姓名：PJM
    学号：5678      姓名：KTH
    学号：8765      姓名：JJK
是否想查询指定选课详细学生信息

```

7.2.3按照一定顺序查看某课程选课学生的详细信息

```

是否想查询指定选课详细学生信息
--1 for 是 ; else for 退出: 1
--请输入想要查询的课程编号: 1
--请选择学生信息呈现方式: 1for按学号升序; 2for按学号降序; 3for按成绩升序; 4for按成绩降序
1
姓名: JHS
学院: scs
班级: 11
学号: 1234
成绩: 成绩未录入
成绩是否提交: 未提交
=====
姓名: PJM
学院: scs
班级: 12
学号: 4321
成绩: 成绩未录入
成绩是否提交: 未提交
=====
姓名: KTH
学院: scs
班级: 13
学号: 5678
成绩: 成绩未录入
成绩是否提交: 未提交
=====
姓名: JJK
学院: scs
班级: 14
学号: 8765

```

8.实现课程成绩的录入、修改、提交操作，仅老师可用

8.1成绩录入和修改

```
-----
--请选择将要进行的操作：1. 课程查看 2. 成绩操作 3. 注销登录
2
--请输入想要进行操作的课程编码：1
--请选择成绩操作类型：1 for 成绩录入、修改;2 for 成绩提交
1
--请选择想要进行成绩操作的学生(学号):1234
--请选择操作：1 for 成绩录入；2 for 成绩修改
1
--请输入成绩[范围：0-100]：90
成绩记录成功！
--请选择将要进行的操作：1. 课程查看 2. 成绩操作 3. 注销登录
```

8.2成绩提交

```
--请选择将要进行的操作：1. 课程查看 2. 成绩操作 3. 注销登录
2
--请输入想要进行操作的课程编码：1
--请选择成绩操作类型：1 for 成绩录入、修改;2 for 成绩提交
2
成绩提交成功
--请选择将要进行的操作：1. 课程查看 2. 成绩操作 3. 注销登录
```

8.3查询指定选课学生信息

可按一定方式排序

是否想查询指定选课详细学生信息

--1 for 是 ; else for 退出: 1

--请输入想要查询的课程编号: 1

--请选择学生信息呈现方式: 1for按学号升序; 2for按学号降序; 3for按成绩升序; 4for按成绩降序

3

姓名: KTH

学院: scs

班级: 13

学号: 5678

成绩: 55

成绩是否提交: 已提交

=====

姓名: PJM

学院: scs

班级: 12

学号: 4321

成绩: 80

成绩是否提交: 已提交

=====

姓名: JHS

学院: scs

班级: 11

学号: 1234

成绩: 90

成绩是否提交: 已提交

=====

姓名: JJK

学院: scs

班级: 14

学号: 8765

成绩: 100

成绩是否提交: 已提交

=====

8.4无法提交成绩

本课程成绩有学生未录入时无法提交成绩

--请选择将要进行的操作: 1. 课程查看 2. 成绩操作 3. 注销登录

2

--请输入想要进行操作的课程编码: 5

--请选择成绩操作类型: 1 for 成绩录入、修改; 2 for 成绩提交

2

本课程还有学生成绩未录入, 请检查后再提交

9.实现课程成绩和绩点的计算和查看功能

可进行排序查看，仅学生可用

```
--请选择将要进行的操作：1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录
3
--请选择课程信息呈现方式：1for按成绩升序；2for按成绩降序；3for按绩点升序；4for按绩点降序
2
课程：CS
成绩：90
绩点：1.36364

课程：Math
成绩：90
绩点：1.10769

课程：English
成绩：未录入

课程：EE
成绩：未录入

课程：Music
成绩：未录入
```

8.系统关闭

```
C:\Users\11794\Desktop\学生成绩管理系统\学生成绩管理系统 (3\学生成绩管理系统\Debug\学生成绩管理系统.exe

接受到一个连接：127.0.0.1
等待连接...
接受到一个连接：127.0.0.1
等待连接...
接受到一个连接：127.0.0.1

||您已退出学生成绩管理系统||
请按任意键继续. . .

C:\Users\11794\Desktop\学生成绩管理系统\学生成绩管理系统 (3\学生成绩管理系统-客户端\Debug\学生成绩管理系统-客户端.exe

--请选择将要进行的操作：1. 课程选择 2. 课程查询 3. 成绩查询 4. 注销登录
4
||您已注销登录||
--请选择用户类型：1. 学生 2. 教师 shut_down. 退出
shut_down
客户端关闭
请按任意键继续. . .
```

9.实现所有数据的文件存取功能

即重启程序以后可以访问上次使用后的数据。文件存取操作通过重载运算符“>>”和“<<”来完成，有异常处理

9.1文件存取功能：

9.1.1系统运行前（修改前）：

课程信息:

1	1	4	Math	1111	0
1	2	5	English	1111	0
1	3	5	EE	1111	0
1	4	5	CS	2222	1
2	5	2	Music	2222	0
2	6	1	PE	2222	0
2	7	2	Chinese	3333	0
2	8	1	Dance	3333	0
2	9	3	Spanish	4444	0

学生信息:

5	1234	JHS	11	scs	4	90	1.36364	1	-1	-1	2	-1	-1	3	-1	-1	5	-1	-1
4	4321	PJM	12	scs	1	-1	-1	2	-1	-1	3	-1	-1	4	80	1.21212			
5	5678	KTH	13	scs	4	100	1.51515	1	-1	-1	2	-1	-1	3	-1	-1	5	-1	-1
4	8765	JJK	14	scs	1	-1	-1	2	-1	-1	3	-1	-1	4	60	0.909091			

9.1.2系统运行完正常退出后（修改后）：

课程信息:

1	1	4	Math	1111	1
1	2	5	English	1111	0
1	3	5	EE	1111	0
1	4	5	CS	2222	1
2	5	2	Music	2222	0
2	6	1	PE	2222	0
2	7	2	Chinese	3333	0
2	8	1	Dance	3333	0
2	9	3	Spanish	4444	0

学生信息:

5	1234	JHS	11	scs	4	90	1.36364	1	90	1.10769	2	80	-1	3	-1	-1	5	-1	-1	
4	4321	PJM	12	scs	1	80	0.984615	2	-1	-1	3	-1	-1	4	80	1.21212				
5	5678	KTH	13	scs	4	100	1.51515	1	55	0.676923	2	2	-1	-1	3	-1	-1	5	-1	-1
4	8765	JJK	14	scs	1	100	1.23077	2	-1	-1	3	-1	-1	4	60	0.909091				

9.2利用重载<</>>完成文件存取

(以课程信息为例)：

```
istream & operator >> (istream &in, Course &c)
{
    try {
        string s;
        int jishu = 1;
        while (in >> s) {
            stringstream ss(s);
            if (jishu % 6 == 1) {
                //c.kind = 's';
                ss >> c.kind;
                jishu++;
            }
            else if (jishu % 6 == 2) {
                ss >> c.NO;
                jishu++;
            }
            else if (jishu % 6 == 3) {
                ss >> c.point;
                jishu++;
            }
            else if (jishu % 6 == 4) {
                c.couname = s;
                jishu++;
            }
            else if (jishu % 6 == 5) {
                c.teacher_num = s;
                jishu++;
            }
            else if (jishu % 6 == 0) {
                ss >> c.putin_key;
                jishu++;
                course.push_back(c);
            }
        }
        if (jishu % 6 != 1) {

            closesocket(sClient);
            sClient = accept(slisten, (SOCKADDR *)&remoteAddr, &nAddrlen);
            const char * sendData = "读取文件内容有误! \n";
            send(sClient, sendData, strlen(sendData), 0);

            throw "课程信息文件内容有误! ";
        }
    }
    catch (char *msg) {
        cout << "错误! " << msg << endl;
        system("pause");
        exit(0);
    }
}
```



```

    }
    return in;
}

```

```

ostream & operator << (ostream &out, const Course &c)
{
    for (int i = 0; i < course.size(); i++) {
        out << course[i].kind << " ";
        out << course[i].NO << " ";
        out << course[i].point << " ";
        out << course[i].couname << " ";
        out << course[i].teacher_num << " ";
        out << course[i].putin_key << endl;
    }
    return out;
    //out<<
}

```

```

void course_save()
{
    ofstream out("C://Users//11794//Desktop//学生成绩管理系统//学生成绩管理系统 (3//学生成绩管理系
统//学生成绩管理系统//course_information.txt");//打开文件
    Course c2;
    out << c2;
    out.close();
}

```

```

int coursemain()
{
    ifstream in("C://Users//11794//Desktop//学生成绩管理系统//学生成绩管理系统 (3//学生成绩管理系
统//学生成绩管理系统//course_information.txt");//打开文件
    // C:\Users\11794\Desktop\学生成绩管理系统\学生成绩管理系统 (3\学生成绩管理系统\学生成绩管理系
统
    Course c1;
    in >> c1;
    initial_courselist();
    in.close();

    return 0;
}

```

9.3 差错控制部分:

```

//利用try-catch机制
try{
    if (jishu % 6 != 1) {
        closesocket(sClient);
        sClient = accept(slisten, (SOCKADDR *)&remoteAddr, &nAddrLen);
        const char * sendData = "读取文件内容有误! \n";
    }
}

```

```

        send(sClient, sendData, strlen(sendData), 0);

        throw "课程信息文件内容有误! ";
    }
    catch (char *msg) {
        cout << "错误! " << msg << endl;
        system("pause");
        exit(0);
    }
}

```

七、改进和评价

1.增强容错性

对于操作选择的键盘输入和文件内容的存取，作了许多错误判断，增强容错性

eg1.

教师信息的流式读入，判断某老师是否有教授的课程，若没有，则不予加入;同时也判断老师的个人信息是否完整，若不完整，则不予加入。并告知用户教师信息读取错误：

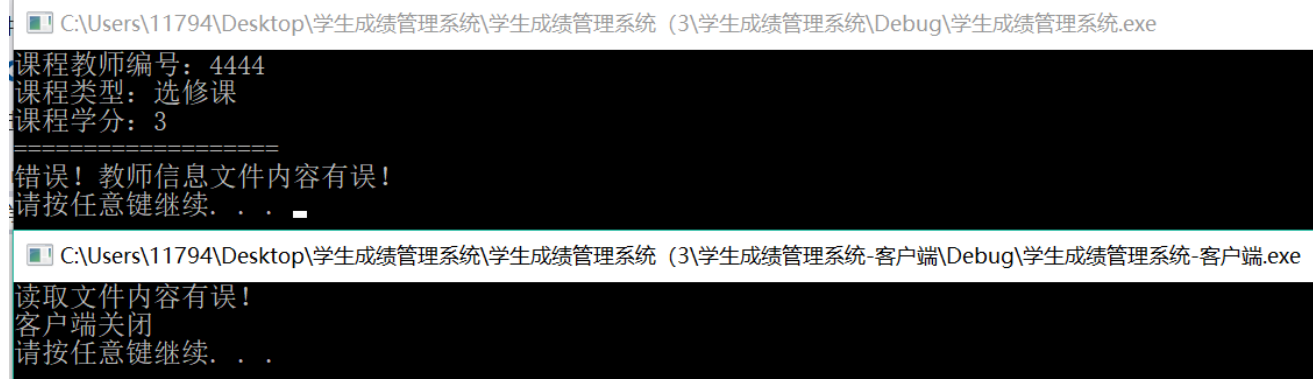
```

int find = 0;
for (int j = 0; j < teacher.size(); j++) {
    for (int i = 0; i < course.size(); i++) {
        if (teacher[j].tea_num == course[i].teacher_num) {
            find++;
        }
    }
}

if ((jishu % 3 != 1) || (find != course.size())) {
    closesocket(sClient);
    sClient = accept(slisten, (SOCKADDR *)&remoteAddr, &nAddrlen);
    const char * sendData = "读取文件内容有误! \n";
    send(sClient, sendData, strlen(sendData), 0);

    throw "教师信息文件内容有误! ";
}

```



eg2.

操作选择的键盘输入有误，告知用户重新输入：

```
--请选择用户类型：1. 学生    2. 教师    shut_down. 退出
2
--您已选择了教师身份，请输入工号：1111
教师姓名：KSJ
教师学院：scs
教师工号：1111
教授课程：1-Math  2-English  3-EE
=====
--请选择将要进行的操作：1. 课程查看  2. 成绩操作  3. 注销登录
4
请输入正确序号
--请选择将要进行的操作：1. 课程查看  2. 成绩操作  3. 注销登录
```

2.套接字信息接收出错

一开始编写网络访问部分代码时，发现客户端第二次接收网络上的套接字时无法接收，但服务器端可以成功发送。

经查询资料发现，accept()函数容易发生阻塞，考虑设置子套接字，用于特定信息传输，每次信息传输完成后即释放连接和分配给此套接字的资源（原始的父亲套接字则在总连接停止后关闭）

3.信息的读取、修改和保存

对于课程、教师、学生信息的读取、修改和保存，考虑读入时用全局的vector保存，方便信息的修改和交互