上机实验3构造函数和析构函数

实验内容1

1.实验内容

观察例程中的构造函数和析构函数的运行顺序；

在main()函数中加入如下代码，观察运行结果：

Coordinate y(5);

y.InputCoord();

y.ShowCoord();

y.ShowAvgCoord();

2.程序代码

#include<iostream>

using namespace std;

class Coordinate {

public:

Coordinate()

{

times = 2;

cout << "Coordinate construction 1 called!" << endl;

}

Coordinate(int times1)

{

times = times1;

cout << "Coordinate construction 2 called!" << endl;

}

~Coordinate()

{

cout << "Coordinate destruction called!" << endl;

}

void inputCoord()

{

for (int i = 0; i < times; i++)

{

cout << "Please input x:" << endl;

cin >> Coord[i][1];

cout << "Please input y;" << endl;

cin >> Coord[i][2];

}

}

void showCoord()

{

cout << "The coord is:" << endl;

for (int i = 0; i < times; i++)

{

cout << "(" << Coord[i][1] << "," << Coord[i][2] << ")" << endl;

}

}

void ShowAvgCoord()

{

float avgx = 0;

float avgy = 0;

for (int i = 0; i < times; i++)

{

avgx = avgx + Coord[i][1];

avgy = avgy + Coord[i][2];

}

avgx = avgx / times;

avgy = avgy / times;

cout << "The AVG coord is:" << endl;

cout << "(" << avgx << "," << avgy << ")" << endl;

}

private:

float Coord[100][100];

int times;

};

int main()

{

Coordinate x;

x.inputCoord();

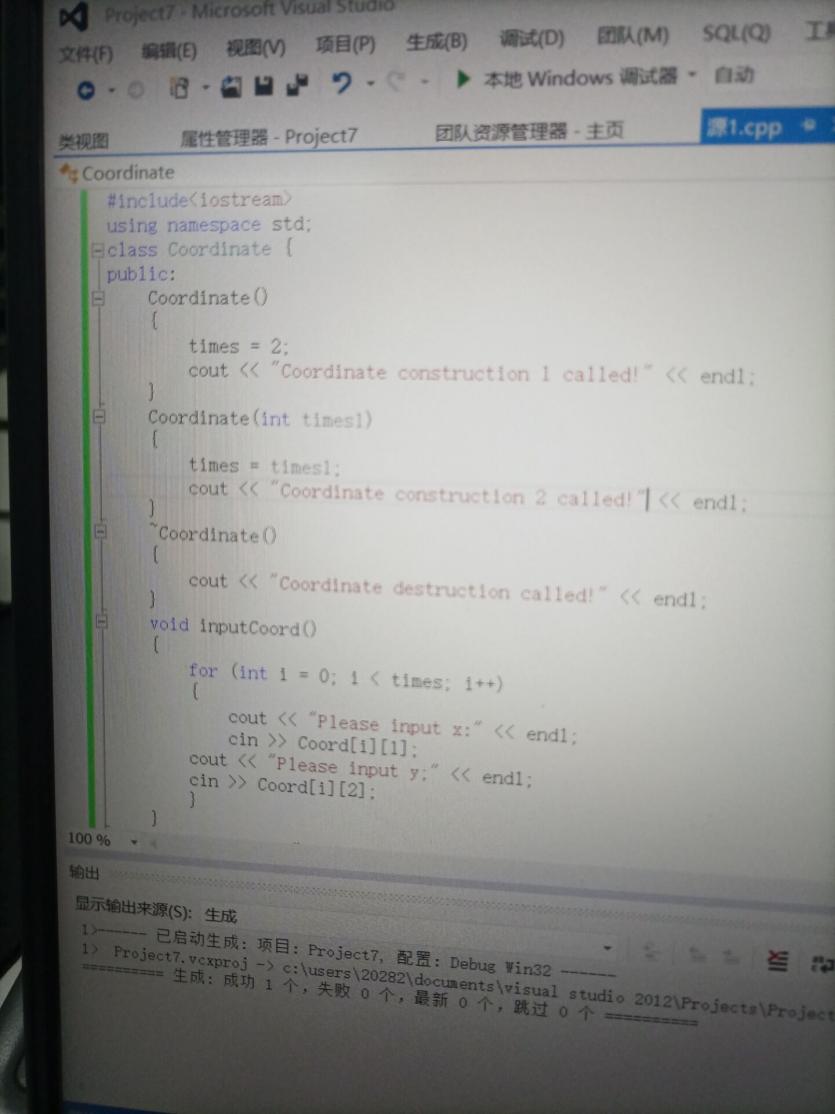
x.showCoord();

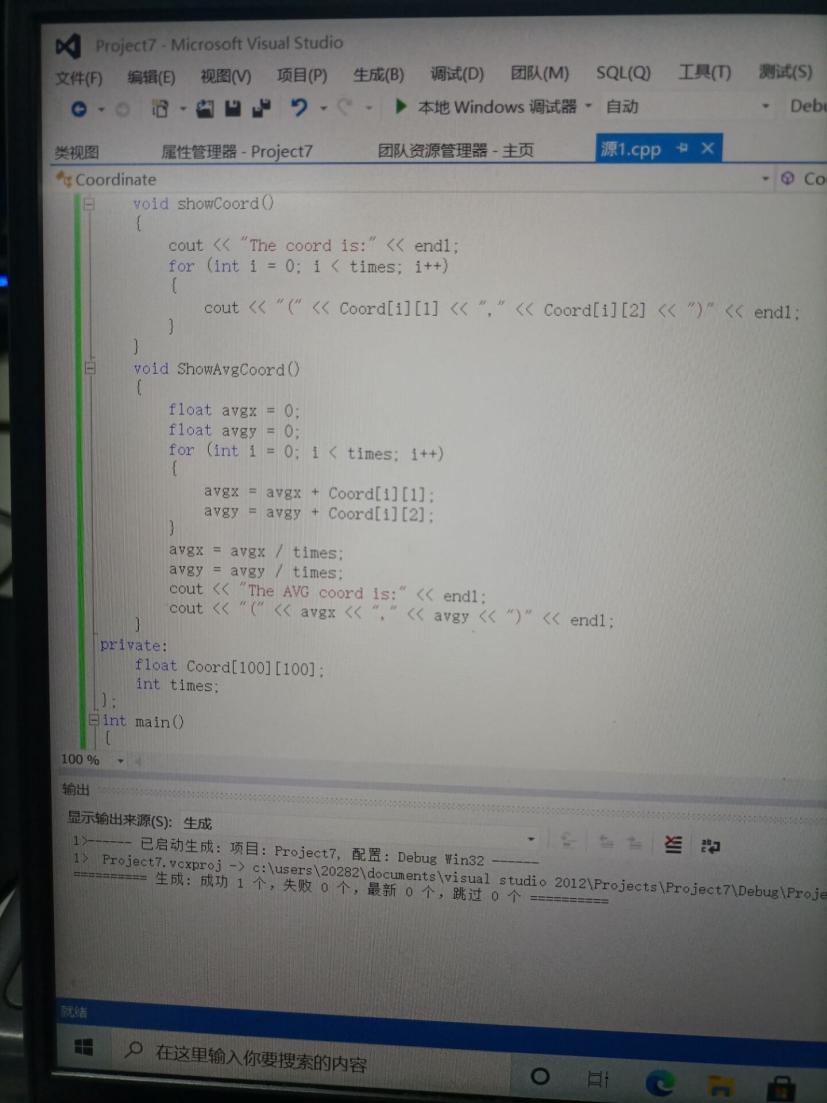
x.ShowAvgCoord();

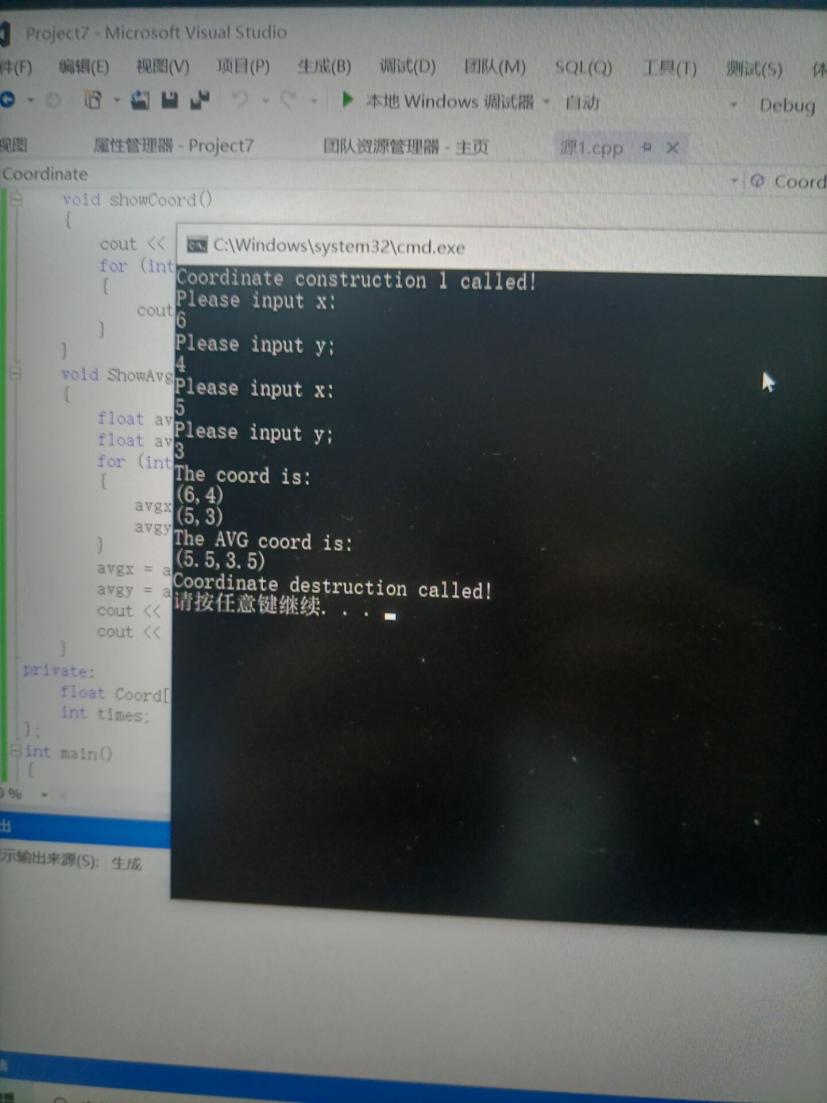
return 0;

}

3.程序结果







实验内容2

1. 实验内容

创建一个Score类，完成以下功能：

连续输入多位学生的成绩（成绩=科目A成绩+科目B成绩+科目C成绩）；

学生数目可以由用户自定义（默认为2个，最多为100个）；

显示每位同学的每科成绩和平均分；

显示每门科目的平均成绩；

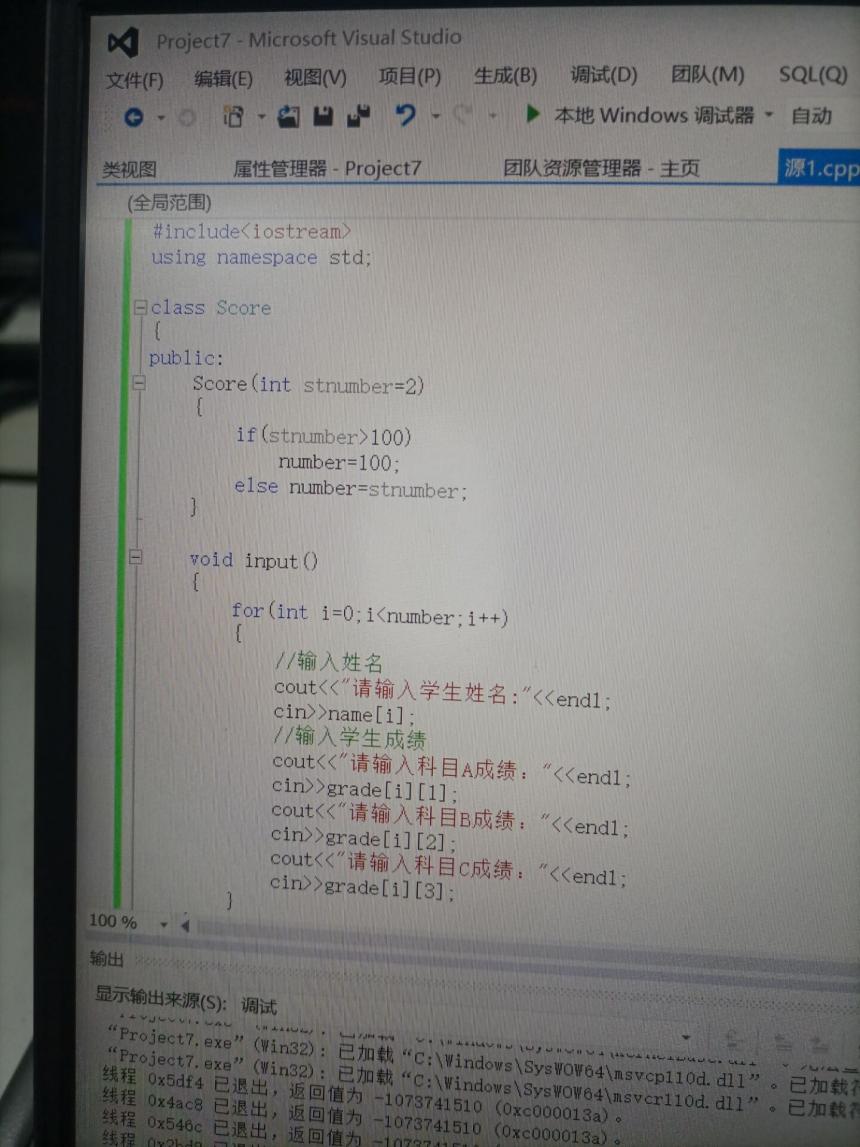
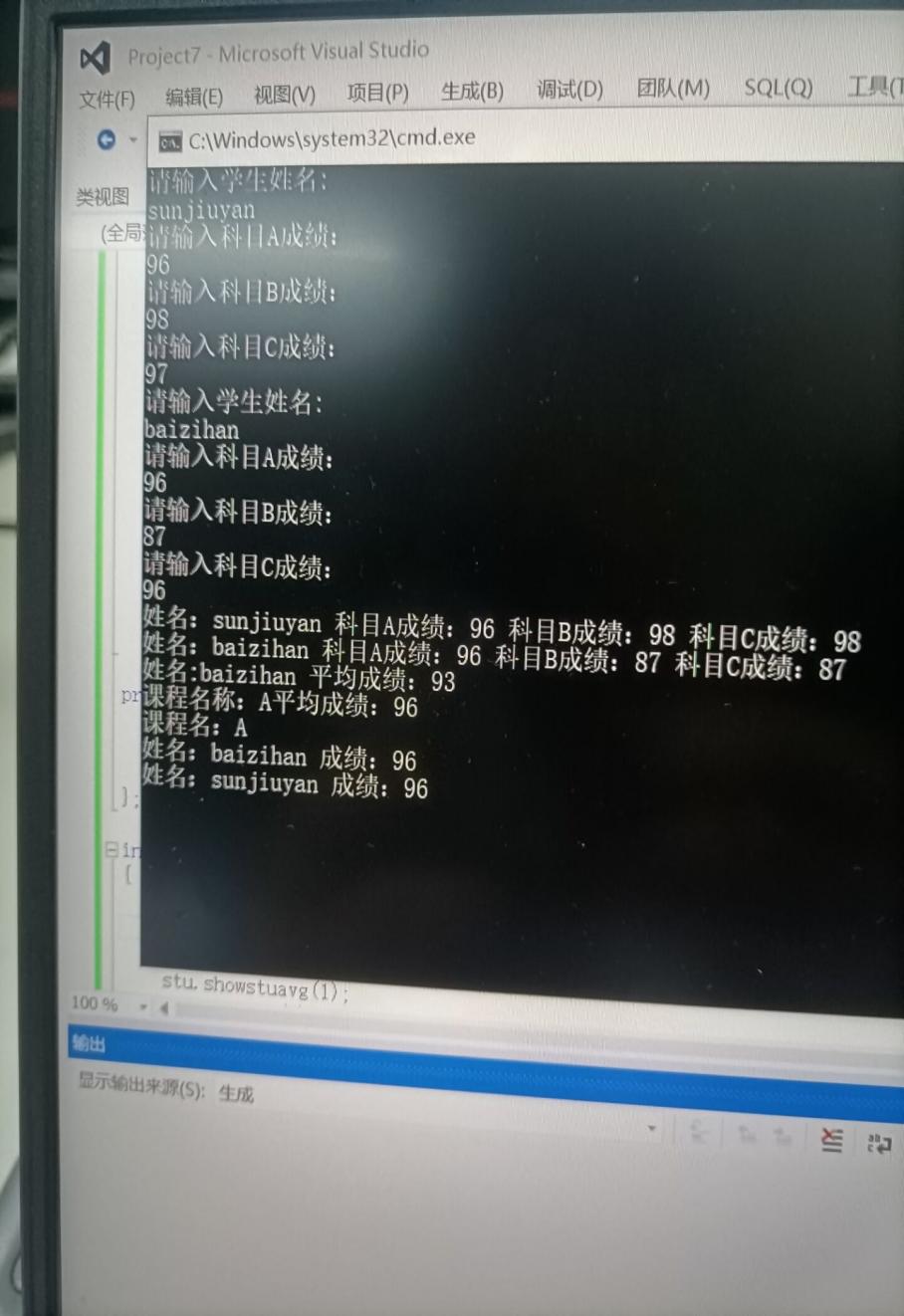
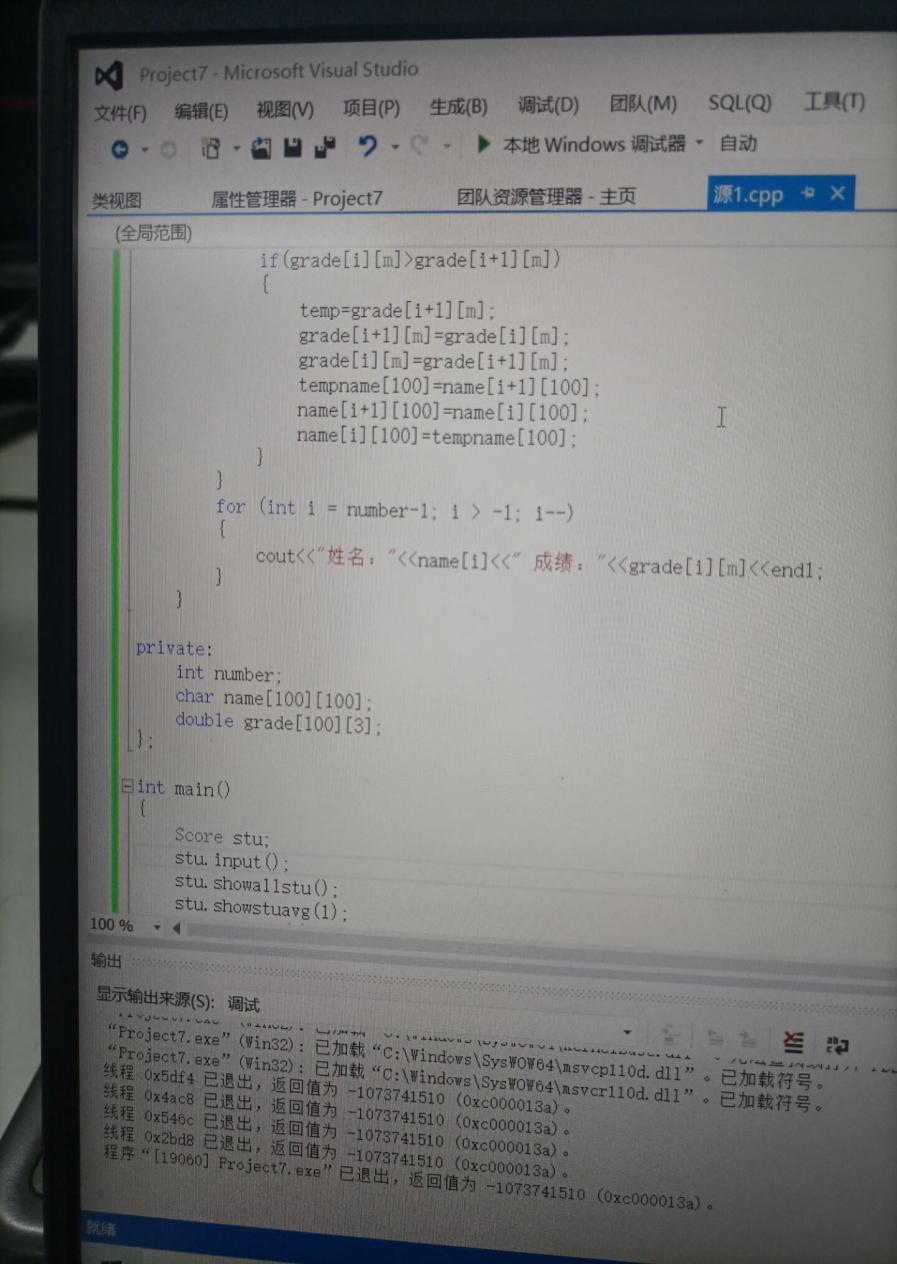
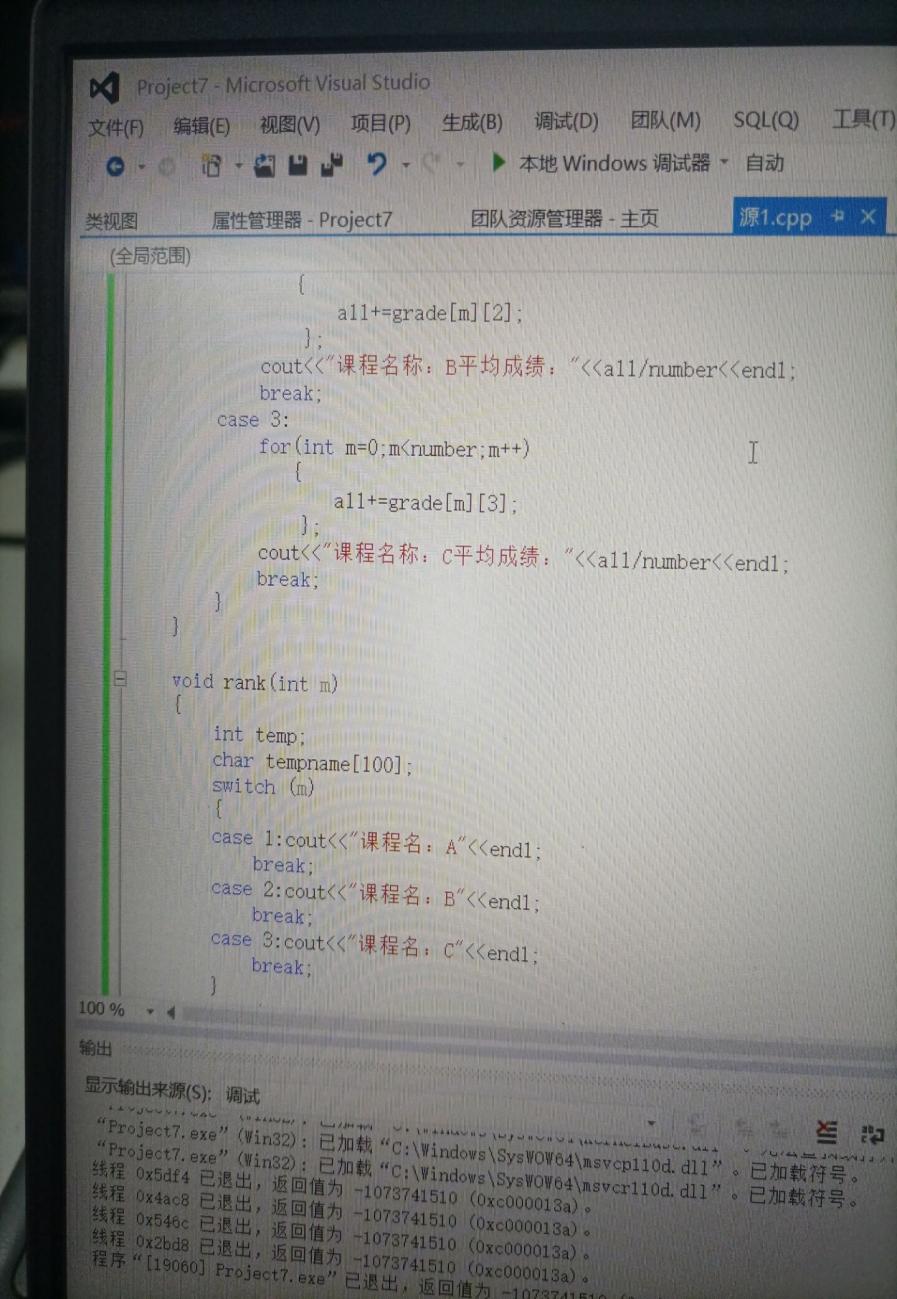
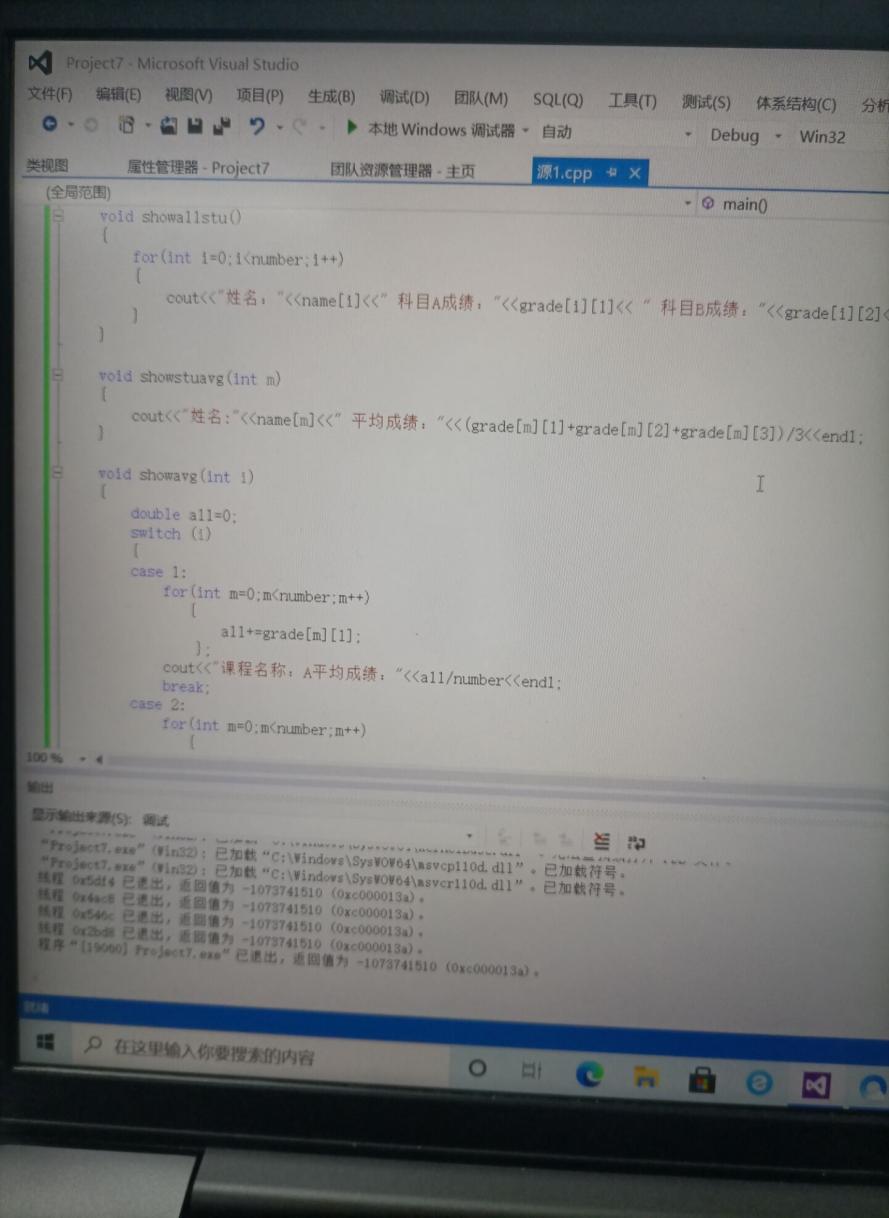
对每门成绩进行排序并由高到底显示；

对整个文件进行打包。

1. 程序代码

#include<iostream>  
using namespace std;  
  
class Score  
{  
public:  
Score(int stnumber=2)  
{  
if(stnumber>100)  
number=100;  
else number=stnumber;  
}  
  
void input()  
{  
for(int i=0;i<number;i++)  
{  
//输入姓名  
cout<<"请输入学生姓名:"<<endl;  
cin>>name[i];  
//输入学生成绩  
cout<<"请输入科目A成绩："<<endl;  
cin>>grade[i][1];  
cout<<"请输入科目B成绩："<<endl;  
cin>>grade[i][2];  
cout<<"请输入科目C成绩："<<endl;  
cin>>grade[i][3];  
}  
}  
  
void showallstu()  
{  
for(int i=0;i<number;i++)  
{  
cout<<"姓名："<<name[i]<<" 科目A成绩："<<grade[i][1]<< " 科目B成绩："<<grade[i][2]<<" 科目C成绩："<<grade[i][2]<<endl;  
}  
}  
  
void showstuavg(int m)  
{  
cout<<"姓名:"<<name[m]<<" 平均成绩："<<(grade[m][1]+grade[m][2]+grade[m][3])/3<<endl;  
}  
  
void showavg(int i)  
{  
double all=0;  
switch (i)  
{  
case 1:  
for(int m=0;m<number;m++)  
   {  
   all+=grade[m][1];  
};  
cout<<"课程名称：A平均成绩："<<all/number<<endl;  
break;  
case 2:  
for(int m=0;m<number;m++)  
   {  
   all+=grade[m][2];  
};  
cout<<"课程名称：B平均成绩："<<all/number<<endl;  
break;  
case 3:  
for(int m=0;m<number;m++)  
   {  
   all+=grade[m][3];  
};  
cout<<"课程名称：C平均成绩："<<all/number<<endl;  
break;  
}  
}  
  
void rank(int m)  
{  
int temp;  
char tempname[100];  
switch (m)  
{  
case 1:cout<<"课程名：A"<<endl;  
break;  
case 2:cout<<"课程名：B"<<endl;  
break;  
case 3:cout<<"课程名：C"<<endl;  
break;  
}  
for (int i = 0; i < number-1; i++)  
{  
if(grade[i][m]>grade[i+1][m])  
{  
temp=grade[i+1][m];  
grade[i+1][m]=grade[i][m];  
grade[i][m]=grade[i+1][m];  
tempname[100]=name[i+1][100];  
name[i+1][100]=name[i][100];  
name[i][100]=tempname[100];  
}  
}  
for (int i = number-1; i > -1; i--)  
{  
cout<<"姓名："<<name[i]<<" 成绩："<<grade[i][m]<<endl;  
}  
}  
  
private:  
int number;  
char name[100][100];  
double grade[100][3];  
};  
  
int main()  
{  
Score stu;  
stu.input();  
stu.showallstu();  
stu.showstuavg(1);  
stu.showavg(1);  
stu.rank(1);  
getchar();  
getchar();  
return 0;

1. 程序结果



感想心得

本次上机的话，自我感觉与上课内容联系非常密切，难度也递增了，通过询问同学以及网上查找，解决了部分问题，收获较多。