**实验内容：**

**创建一个类，完成以下功能：**

**连续输入一组二维坐标值；**

**二维坐标值的数目可以由用户自定义（默认为2个，最多为100组）；**

**显示用户输入的坐标值；**

**显示用户输入坐标值的均值；**

**一、程序代码**

#include<iostream>

using namespace std;

class Coordinate { // 定义Coordinate类

public:

Coordinate()

{

times = 2;

cout << "Coordinate construction1 called!" << endl;

}

Coordinate(int times1)

{

times = times1;

cout << "Coordinate construction2 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]; // 存放输入坐标的数组

int times; // 存放输入坐标数目

};

int main(){

Coordinate y(5);

y.InputCoord();

y.ShowCoord();

y.ShowAvgCoord();

return 0;

}

**实验内容（二）：**

**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。**

程序代码如下：

#include <iostream>

#include <string>

using namespace std;

class score

{

private:

int time;

double cj[100][100];

string name[100];

string copy[100];

public:

score()

{

time = 2;//默认为2

cout << "默认两组" << endl;

}

score(int time1)

{

time = time1;

cout << "用户自定义组数" << endl;

}

~score()

{

cout << "Destructor called" << endl;

}

void input()

{

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

{

cout << "请输入学生的姓名和成绩：" << endl;

cin >> name[i];

cin >> cj[i][1] >> cj[i][2] >> cj[i][3];

}

}

void show()

{

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

{

cout << name[i] << "学科A的成绩为：" << cj[i][1] << " ";

cout << name[i] << "学科B的成绩为：" << cj[i][2] << " ";

cout << name[i] << "学科C的成绩为：" << cj[i][3] << endl;

}

}

void avg()

{

double a = 0;

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

{

a = cj[i][1] + cj[i][2] + cj[i][3];

cout << name[i] << "的平均成绩为：" << a / 3 << " ";

}

cout << endl;

}

void showavg()

{

double a = 0;

double b = 0;

double c = 0;

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

a = a + cj[i][1];

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

b = b + cj[i][2];

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

c = c + cj[i][3];

cout << "学科A的平均成绩为" << a / time << " ";

cout << "学科B的平均成绩为" << b / time << " ";

cout << "学科C的平均成绩为" << c / time << endl;

}

void px()

{

for (int i = 0; i < time; ++i) //将name数组复制一个

copy[i] = name[i];

for (int i = 0; i < time - 1; i++)

for (int j = 0; j < time - i - 1; j++)

if (cj[j][1] < cj[j + 1][1]) //改变了name数组数据的位置

{

double temp = cj[j + 1][1];

cj[j + 1][1] = cj[j][1];

cj[j][1] = temp;

string t = name[j + 1];

name[j + 1] = name[j];

name[j] = t;

}

cout << "学科A的排序为：" << endl;

for (int i = 0; i < time; ++i) //打印姓名和对应的成绩

{

cout << name[i] << " " << cj[i][1] << endl;

}

for (int i = 0; i < time; ++i) //重置name数组

name[i] = copy[i];

for (int i = 0; i < time - 1; i++)

for (int j = 0; j < time - i - 1; j++)

if (cj[j][2] < cj[j + 1][2])

{

double temp = cj[j + 1][2];

cj[j + 1][2] = cj[j][2];

cj[j][2] = temp;

string t = name[j + 1];

name[j + 1] = name[j];

name[j] = t；

}

cout << "学科B的排序为：" << endl;

for (int i = 0; i < time; ++i) //打印姓名和对应的成绩

{

cout << name[i] << " " << cj[i][2] << endl;

}

for (int i = 0; i < time; ++i) //重置name数组

name[i] = copy[i];

for (int i = 0; i < time - 1; i++)

for (int j = 0; j < time - i - 1; j++)

if (cj[j][3] < cj[j + 1][3])

{

double temp = cj[j + 1][3];

cj[j + 1][3] = cj[j][3];

cj[j][3] = temp;

string t = name[j + 1];

name[j + 1] = name[j];

name[j] = t;

}

cout << "学科C的排序为：" << endl; //打印姓名和对应的成绩

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

{

cout << name[i] << " " << cj[i][3] << endl;

}

}

};

int main()

{

score a(3);

a.input();

a.show();

a.avg();

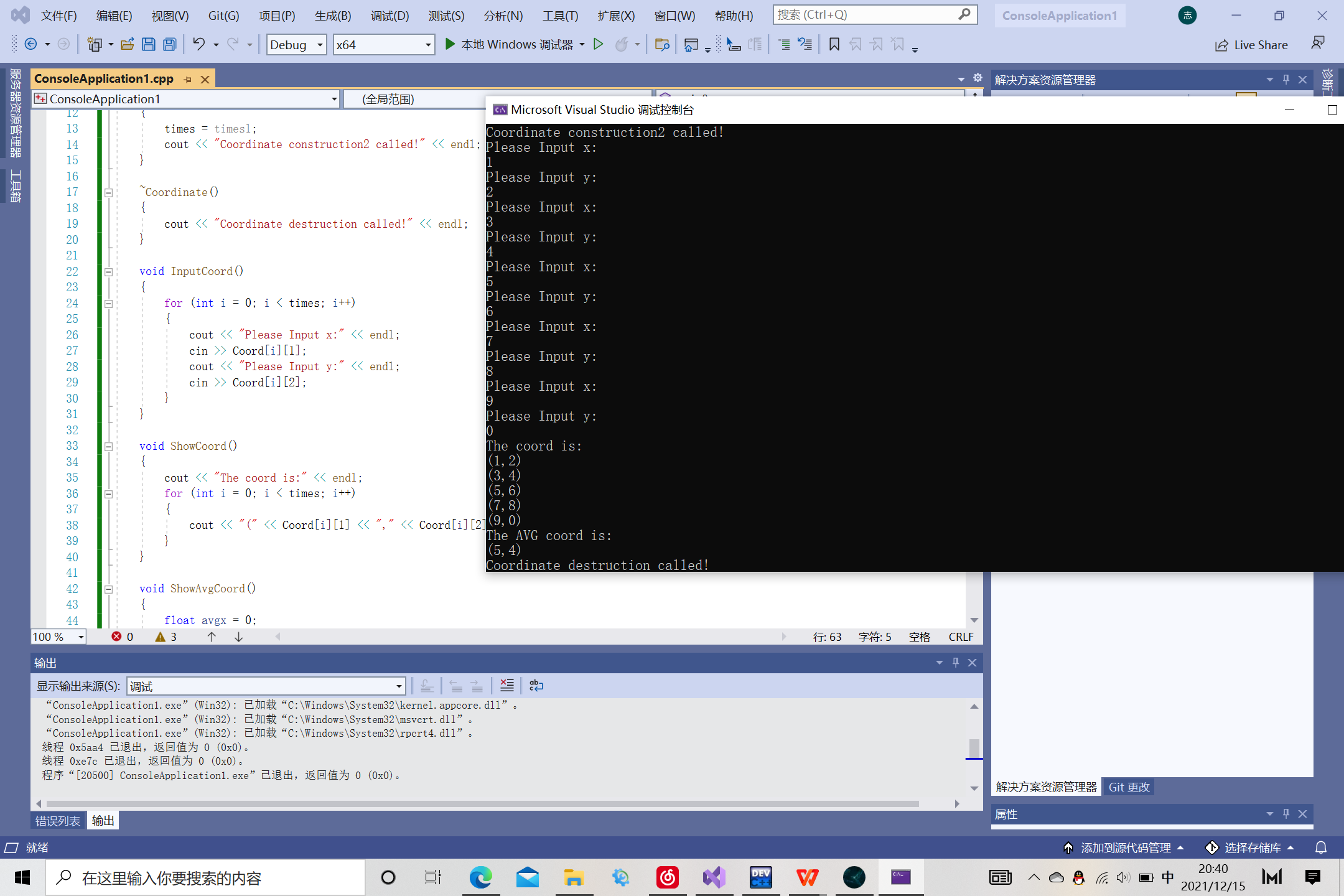
a.showavg();

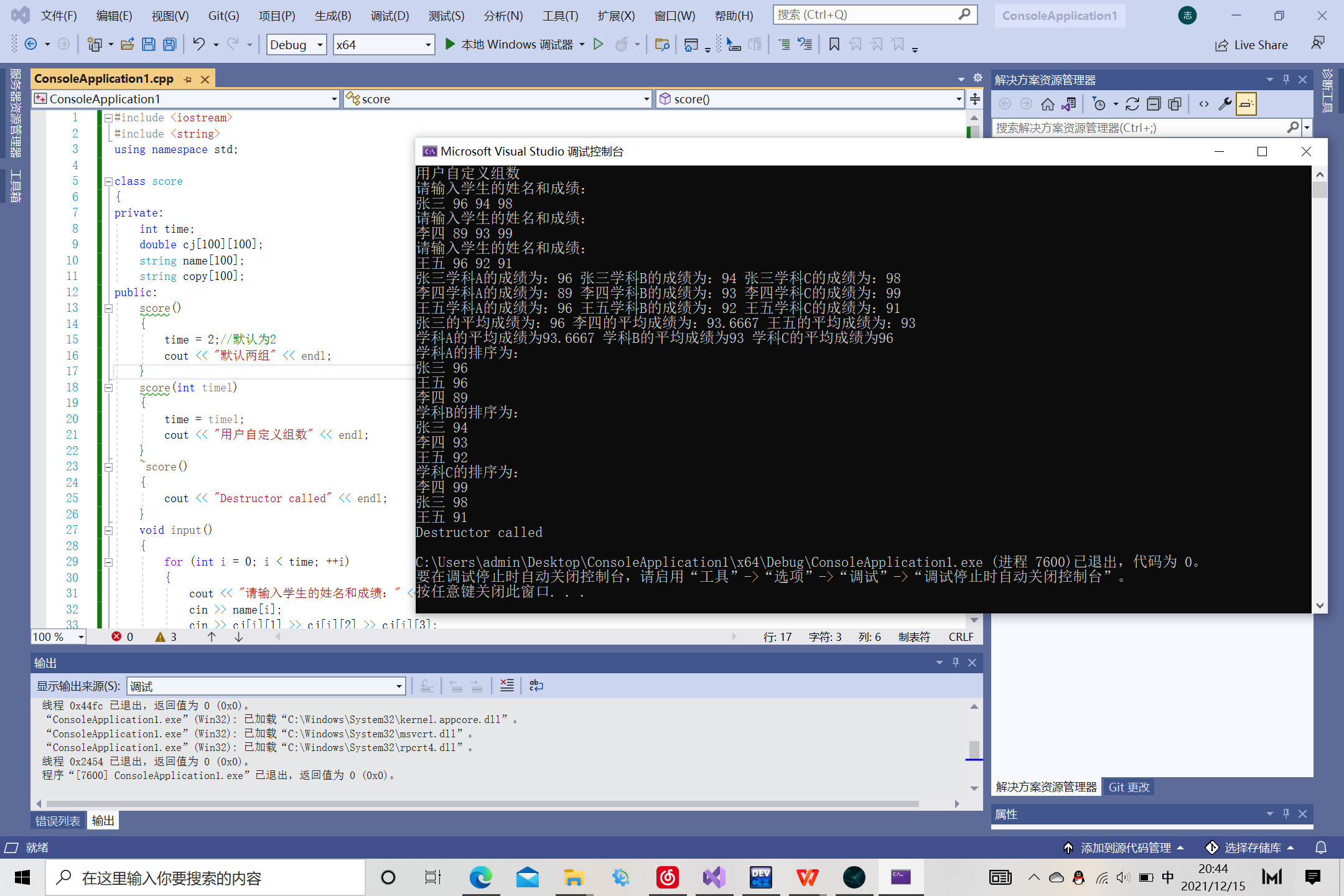
a.px();

return 0;

}

1. 程序结果





1. 感想心得

这章开始，知识点和课后作业都有了一定难度，尤其是在上机实践大家都感到很多阻力，也是第一次书写这么大体量的代码，找同学和学长讨论，解决了运行代码错误的问题，大概认识到了c++的难度以及上机实践能够发现许多问题，完成程序的编写，决不意味着万事大吉。你认为万无一失的程序，实际上机运行时可能持续出现麻烦。如编译程序检测出一大堆错误。有时程序本身不存有语法错误，也能够顺利运行，但是运行结果显然是错误的。