实验报告（第三次上机）

自2001 杨洺宇

实验一:

代码：

#include<iostream>

#include<cmath>

//#include<string>

using namespace std;

class coordinate {

private:

int times;

float a[100][2];

public:

coordinate();

coordinate(int times1);

~coordinate();

void input();

void show();

};

coordinate::coordinate()

{

times = 2;

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

}

coordinate::coordinate(int times1)

{

if (times1 > 100)

times1 = 100;

times = times1;

cout << "Coordinate construction2 called! " << endl;

}

coordinate::~coordinate()

{

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

}

void coordinate::input()

{

int i;

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

{

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

cin >> a[i][1];

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

cin >> a[i][2];

}

}

void coordinate::show()

{

int n;

float avgx=0,avgy=0;

for (n = 0; n < times; n++)

{

avgx = avgx + a[n][1];

avgy = avgy + a[n][2];

}

avgx = avgx / times;

avgy = avgy / times;

cout << "Yuor numbers are: " << endl;

for (n=0;n<times;n++)

{

cout << a[n][1] << " " << a[n][2] << endl;

}

cout << "The averages are: " << endl;

cout << "X is : " << avgx << " " << "Y is : " << avgy << endl;

}

int main()

{

int t;

cout << "How many groups will you input? " << endl;

cin >> t;

coordinate coord(t);

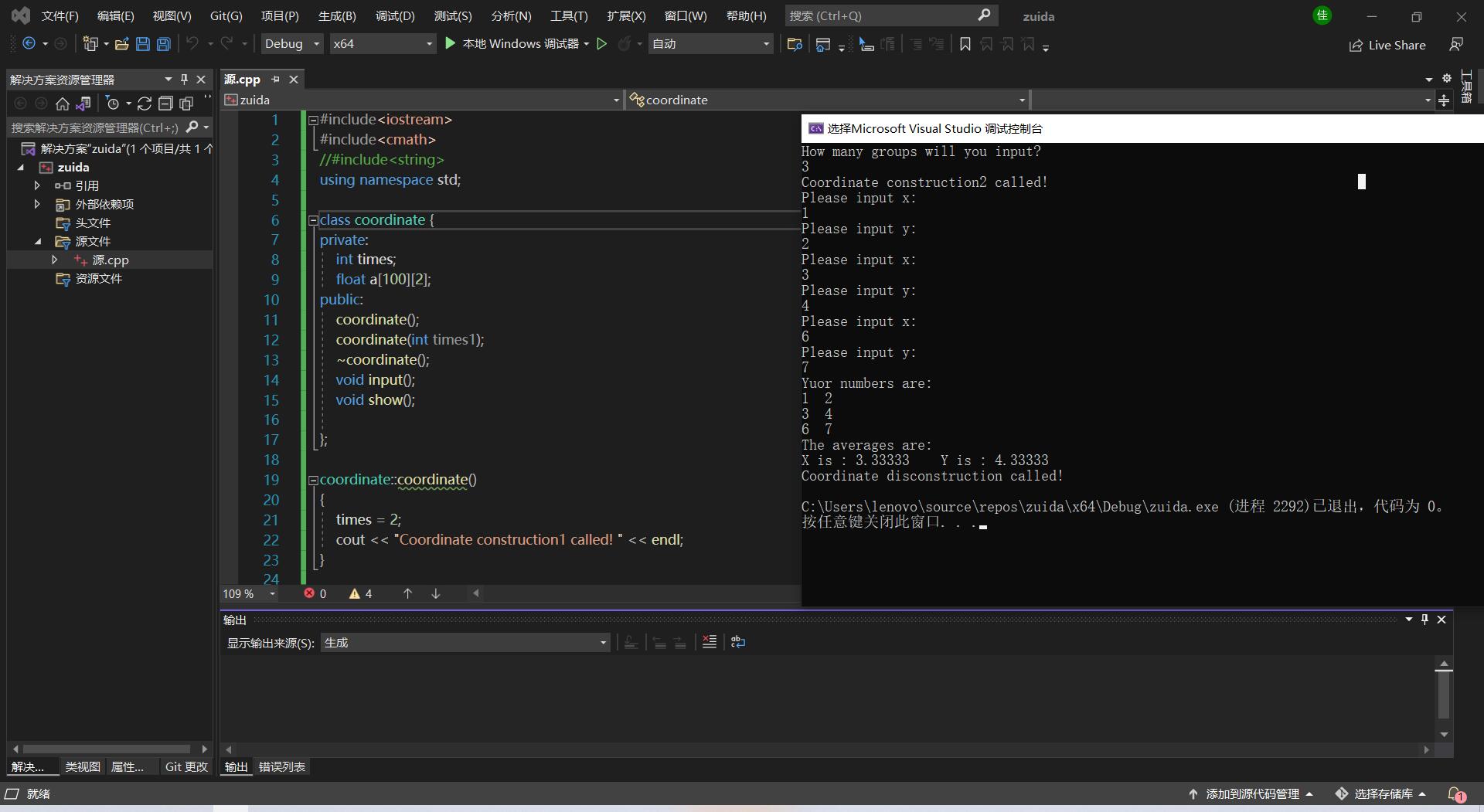
coord.input();

coord.show();

return 0;

}

图片：



实验二：

代码：

//Tips:You are allowed to input 100 students at most !

#include<iostream>

#include<cmath>

#include<string>

using namespace std;

class coordinate {

private:

int times;

float a[100][3];

string b[100];

public:

coordinate(int times1);

void input();

void show();

};

coordinate::coordinate(int times1)

{

if (times1 < 2)

times1 = 2;

times = times1;

//cout << "Coordinate construction2 called! " << endl;

}

void coordinate::input()

{

int i;

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

{

cout << "Please input student's name: " << endl;

cin >> b[i];

cout << "Please input score of courseA: " << endl;

cin >> a[i][1];

cout << "Please input score of courseB: " << endl;

cin >> a[i][2];

cout << "Please input score of courseC: " << endl;

cin >> a[i][3];

}

}

void coordinate::show()

{

int n,m;

float avgA=0,avgB=0,avgC=0,avgD;

for (n = 0; n < times; n++)

{

avgD = 0;

cout << "Name: " << b[n] << endl;

cout << "courseA: " << a[n][1] << " " << "courseB: " << a[n][2] << " " << "courseC: " << a[n][3] << endl;

avgD = (a[n][1] + a[n][2] + a[n][3]) / 3;

cout << "The average score of this student is: " << avgD << endl;

}

for (n = 0; n < times; n++)

{

avgA = avgA + a[n][1];

avgB = avgB + a[n][2];

avgC = avgC + a[n][3];

avgA = avgA / times;

avgB = avgB / times;

avgC = avgC / times;

}

cout << "The average score of courseA is: " << avgA << endl;

cout << "The average score of courseB is: " << avgB << endl;

cout << "The average score of courseC is: " << avgC << endl;

}

int main()

{

int t;

cout << "How many students will you input? " << endl;

cin >> t;

coordinate coord(t);

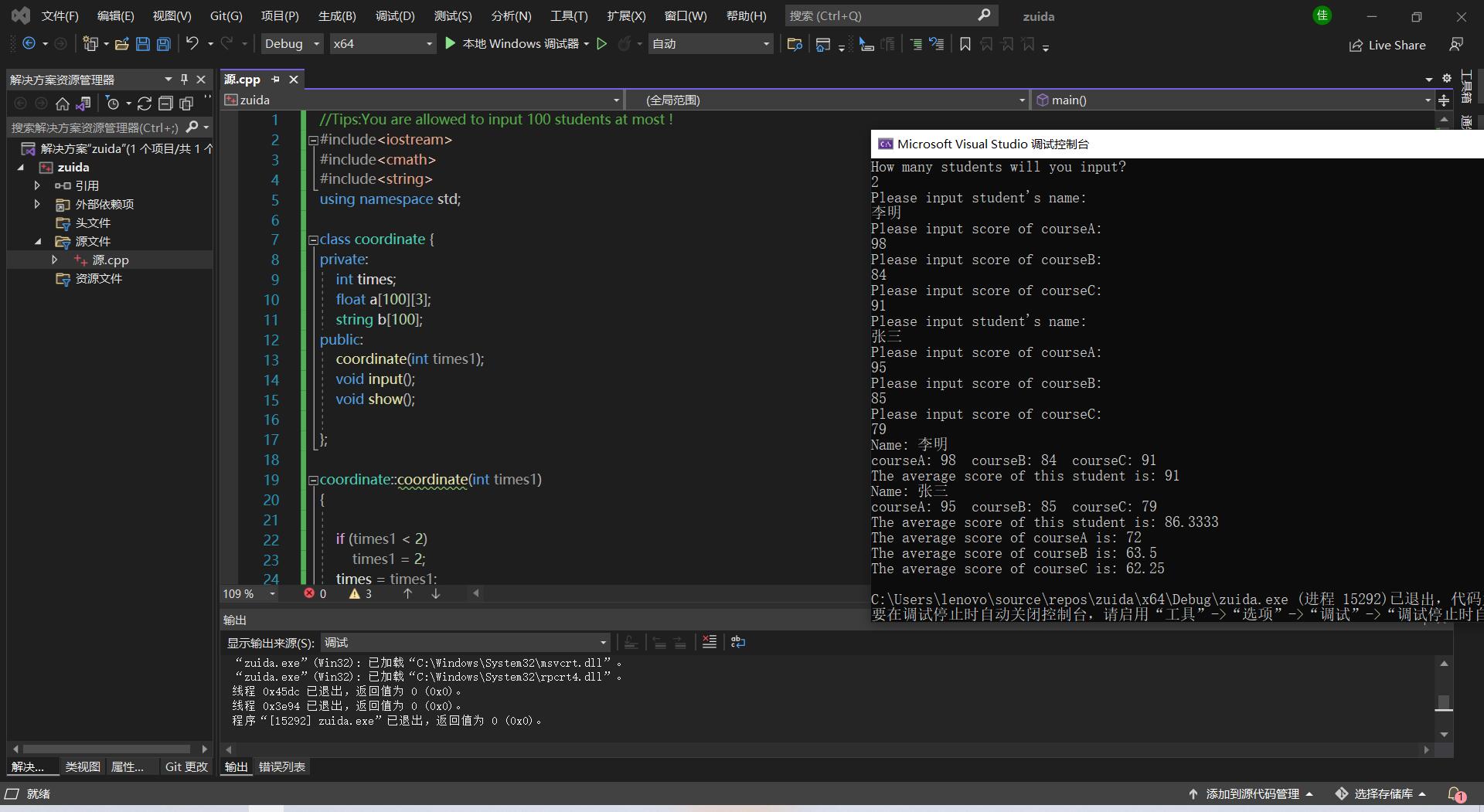
coord.input();

coord.show();

return 0;

}

图片：



总结：

本次实验我学会了构造函数只是起初始化值的作用，但实例化一个对象的时候，可以通过实例去传递参数，从主函数传递到其他的函数里面，这样就使其他的函数里面有值了。规则，只要你一实例化对象，系统自动回调用一个构造函数，就是你不写，编译器也自动调用一次。析构函数与构造函数的作用相反，用于撤销对象的一些特殊任务处理，可以是释放对象分配的内存空间；特点：析构函数与构造函数同名，但该函数前面加~。 析构函数没有参数，也没有返回值，而且不能重载，在一个类中只能有一个析构函数。 当撤销对象时，编译器也会自动调用析构函数。 每一个类必须有一个析构函数，用户可以自定义析构函数，也可以是编译器自动生成默认的析构函数。一般析构函数定义为类的公有成员。