实验内容1

* **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();**

程序代码

#include<iostream>

using namespace std;

class Coordinate {

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];

int times;

int main()

{

Coordinate x;

Coordinate y(5);

y.InputCoord();

y.ShowCoord();

y.ShowAvgCoord();

x.InputCoord();

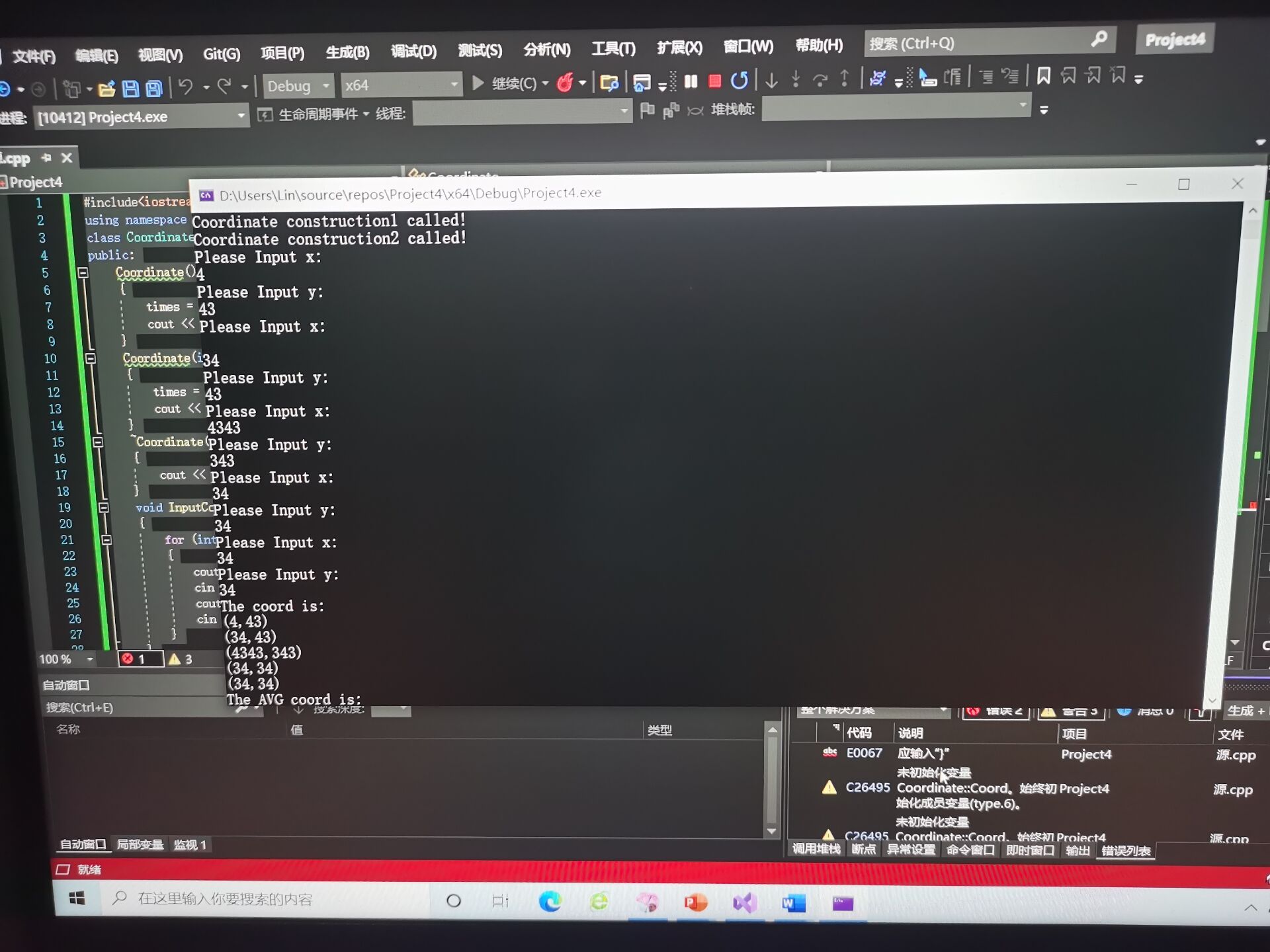
x.ShowCoord();

x.ShowAvgCoord();

return 0;

}

运行结果



实验内容2

* **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

#include<iostream>

#include<string>

using namespace std;

class Score {

public:

Score()

{

times = 2;

}

Score(int times1)

{

times = times1;

}

void InputNameAndScore()

{

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

{

cout << "请输入学生姓名:" << endl;

cin >> Name[i];

cout << "请输入科目A成绩:" << endl;

cin >> SScore[i][1];

cout << "请输入科目B成绩:" << endl;

cin >> SScore[i][2];

cout << "请输入科目C成绩:" << endl;

cin >> SScore[i][3];

}

}

void ShowNameAndScore()

{

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

{

cout << "姓名: " << Name[i] << " 科目A成绩: " << SScore[i][1] << " 科目B成绩 " << SScore[i][2] << " 科目C成绩: " << SScore[i][3] << endl;

}

}

void ShowStdentAvgScore(int Sid)

{

float avg = 0;

avg = (SScore[Sid][1] + SScore[Sid][2] + SScore[Sid][3]) / 3;

cout << "姓名: " << Name[Sid] << " 平均成绩: " << avg << endl;

}

void ShowClassAvgScore(string ClassName)

{

int Cid;

float avg = 0;

if (ClassName == "A") Cid = 1;

if (ClassName == "B") Cid = 2;

if (ClassName == "C") Cid = 3;

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

{

avg = avg + SScore[i][Cid];

}

avg = avg / times;

cout << "课程名称: " << ClassName << "平均成绩: " << avg << endl;

}

void OrderScore(string ClassName)

{

int Cid;

int i = 0;

if (ClassName == "A") Cid = 1;

if (ClassName == "B") Cid = 2;

if (ClassName == "C") Cid = 3;

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

{

SScore1[i] = SScore[i][Cid];

}

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

{

Name1[i] = Name[i];

}

for (i = 1; i < times; i++)

{

if (SScore1[i] > SScore1[i - 1])

{

float temp = SScore1[i - 1];

SScore1[i - 1] = SScore1[i];

SScore1[i] = temp;

string temp1;

temp1 = Name1[i - 1];

Name1[i - 1] = Name1[i];

Name1[i] = temp1;

}

}

cout << "课程名称: " << ClassName << endl;

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

{

cout << "姓名: " << Name1[i] << " 成绩: " << SScore1[i] << endl;

}

}

private:

float SScore[100][3], SScore1[100];

string Name[100], Name1[100];

int times;

};

int main()

{

Score x;

x.InputNameAndScore();

x.ShowNameAndScore();

x.ShowStdentAvgScore(1);

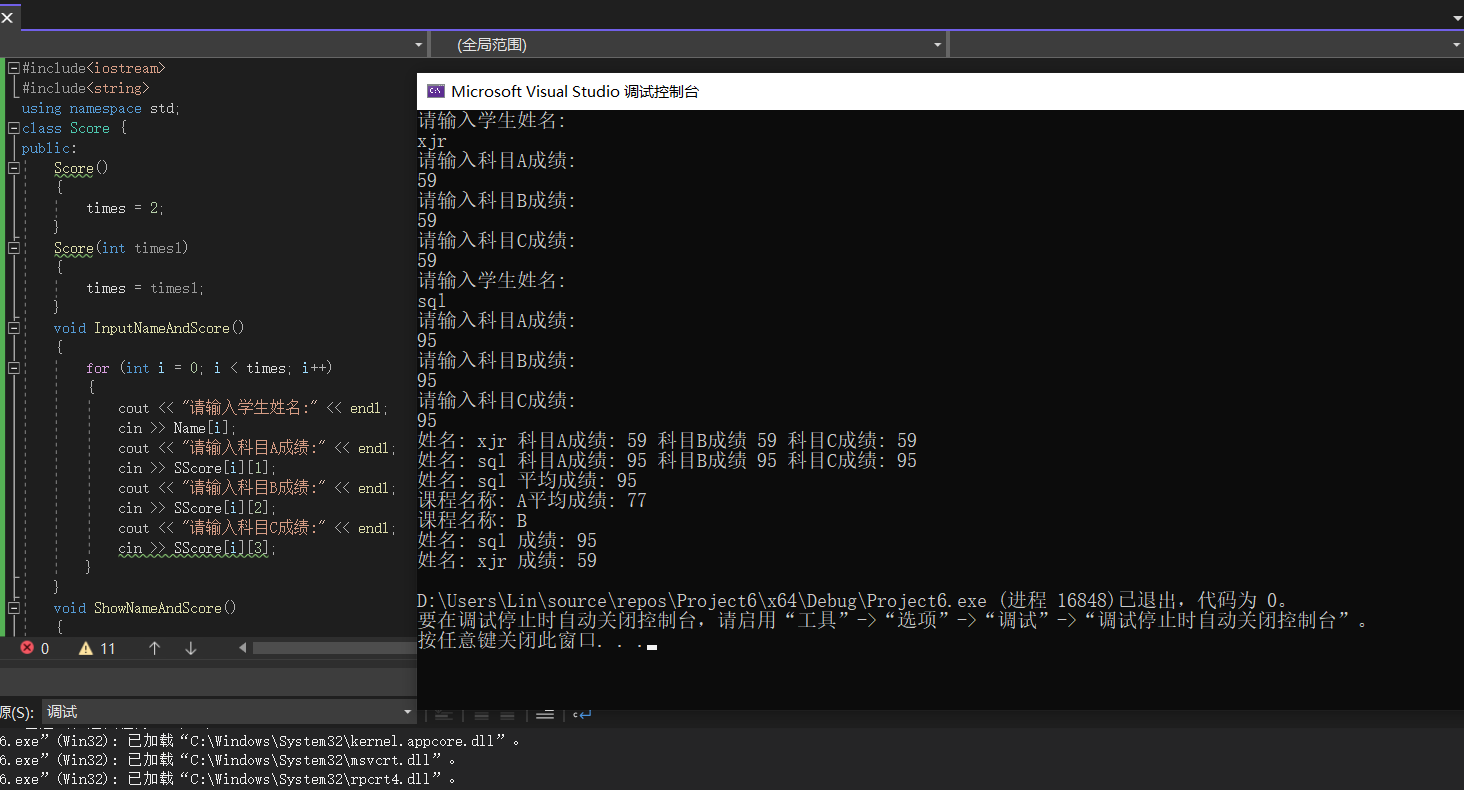
x.ShowClassAvgScore("A");

x.OrderScore("B");

return 0;

}

运行结果



感想心得

通过本次实验，我理解了类和对象的概念，初步掌握了类和定义对象的方法，掌握了构造函数和析构函数的实现方法。通过使用类，我第一次真真切切地感受到++与c语言的巨大不同，果然不能以c语言来类比c++。总之无论代码多长，稳住，就能编写出正确的代码，加油！