## 上机三

**实验一**

#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;

x.inputCoord();

x.showCoord();

x.Showavgcoord();

return 0;

}

**实验二**

代码：

#include<iostream>

using namespace std;

class Score {

public:

Score()

{

Students = 2;

}

Score(int number)

{

Students = number;

}

~Score()

{

}

int i = 1;

void getgrade()

{

for (i = 1;i <= Students; i++) {

cout << "请输入学生姓名：" << endl;

cin >> name[i];

cout << "请输入科目A的成绩：" << endl;

cin >> a[i];

cout << "请输入科目B的成绩：" << endl;

cin >> b[i];

cout << "请输入科目C的成绩：" << endl;

cin >> c[i];

}

}

void showgrade()

{

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

{

cout <<"姓名："<<name[i] << "科目A的成绩：" << a[i] <<" " << "科目B的成绩：" << b[i]<<" " << "科目C的成绩：" << c[i]<<" " << endl;

}

}

void showavgstu()

{

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

{

cout <<name[i]<<":" << ((a[i] + b[i] + c[i]) / 3);

}

}

void showclassgrade(char f)

{

float avg = 0;

if (f == 'a')

{

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

{

avg = avg+a[i];

}

avg = avg / Students;

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

}

if (f == 'b')

{

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

{

avg = avg + b[i];

}

avg = avg / Students;

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

}

if (f == 'c')

{

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

{

avg = avg + c[i];

}

avg = avg / Students;

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

}

}

private:

int Students;

float a[100], b[100], c[100];

string name[100];

};

int main()

{

Score A;

A.getgrade();

A.showgrade();

A.showclassgrade('a');

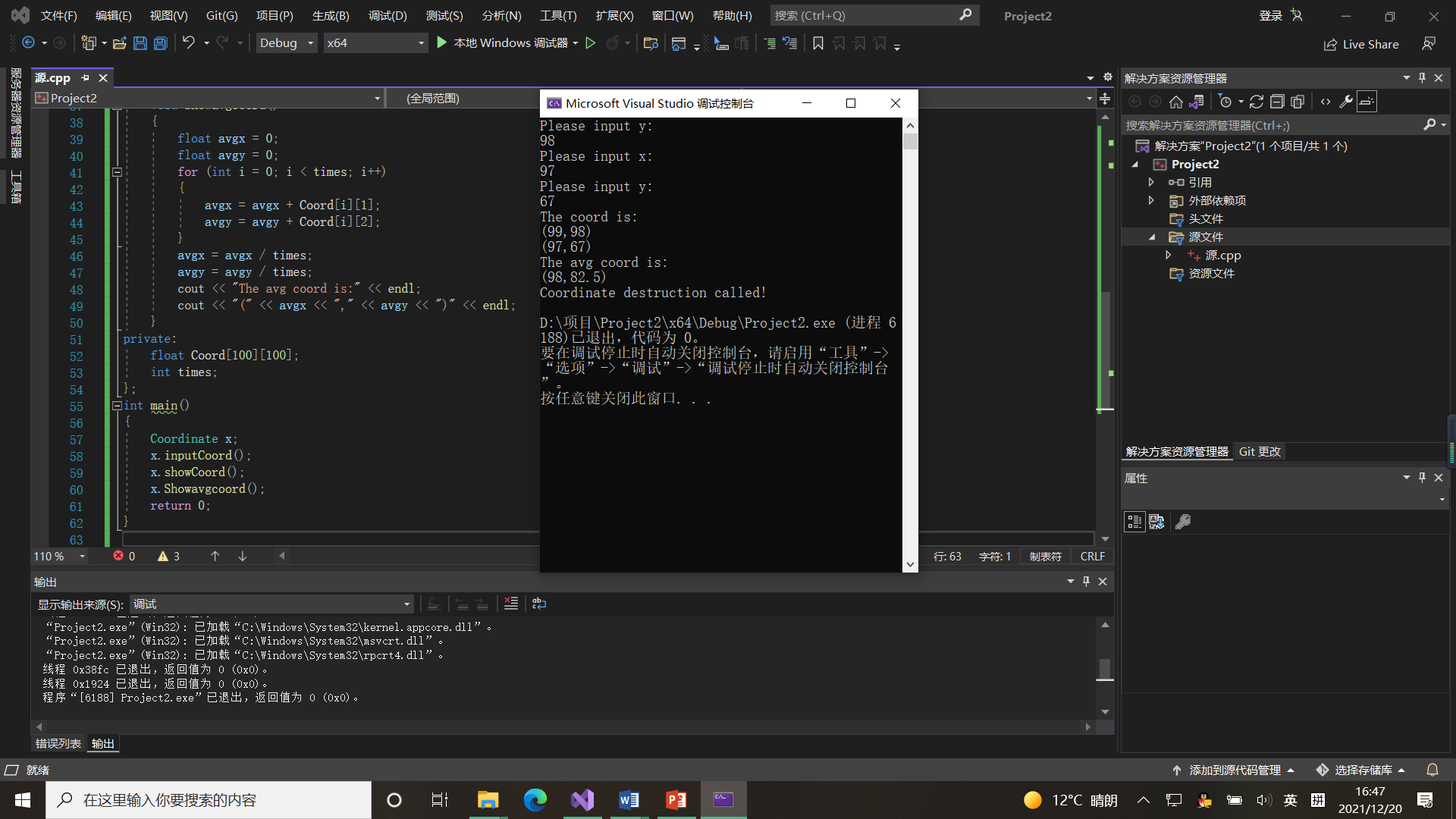
A.showavgstu();

return 0;

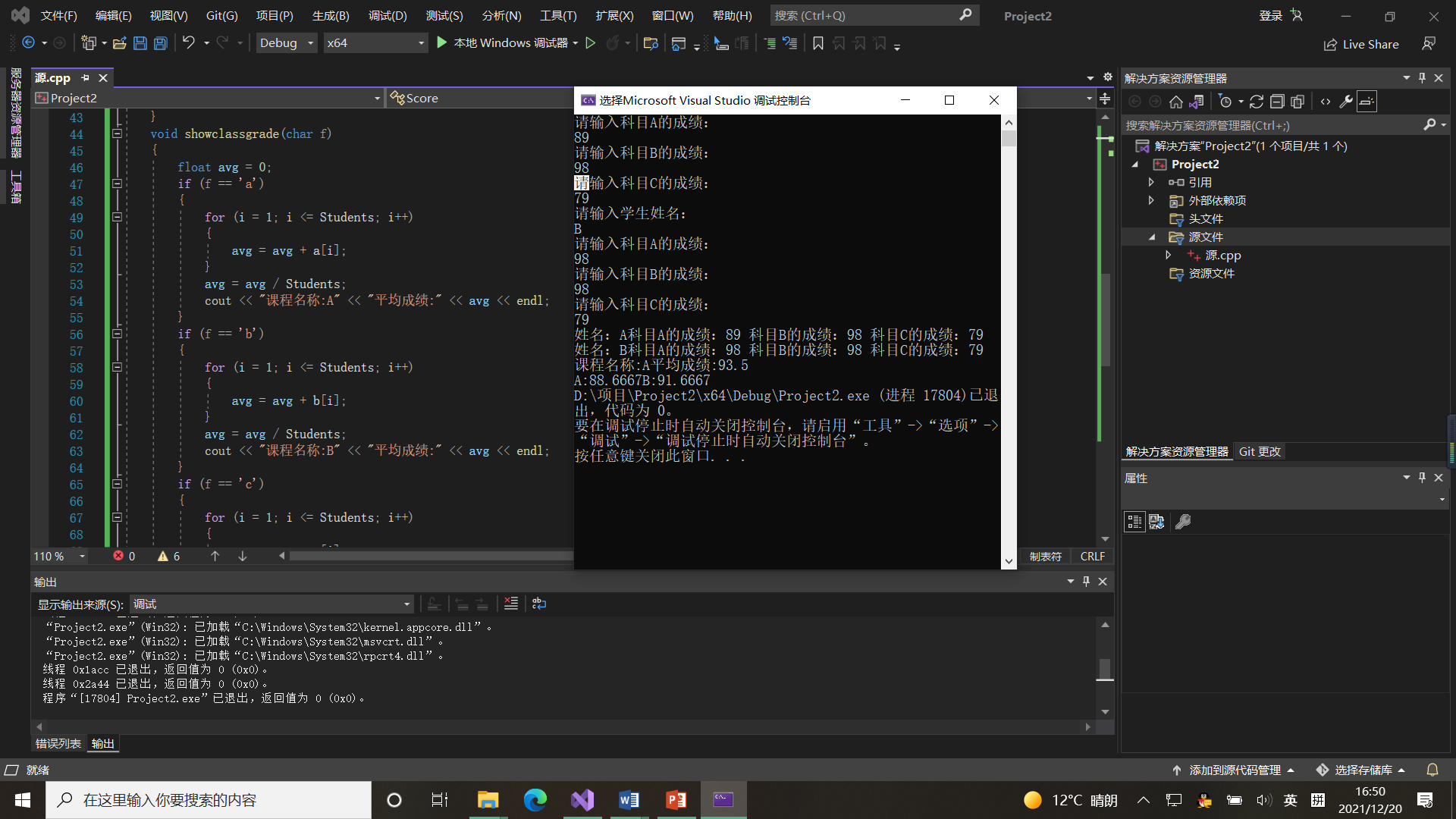
}

**运行结果**

**实验一**



**实验二**



**感想&心得：**

构造函数主要用来在创建对象时初始化对象，即为对象成员变量赋初始值构造函数的命名必须和类名完全相同。它不需要用户来调用，而是在建立对象时自动执行的。

析构函数负责回收存储空间。与构造函数相反，当对象结束其生命周期，如对象所在的函数已调用完毕时，系统会自动执行析构函数。

对于处理少量数据，感觉还是复杂了些，但是需要处理的数据变多之后就体现出类的优越性。

Copyright ©2021-2099 Shuyue. All rights reserved