**实验目的：**

* **理解类和对象的概念，掌握声明类和定义对象的方法**
* **掌握构造函数和析构函数的实现方法**
* **初步掌握使用类和对象编制C++程序**

**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；**

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；**

代码：

#include<iostream>

using namespace std;

class Coordinate

{

public:

Coordinate()

{

times = 2;

cout << "Coordinate construction1 called!" << endl;// 设置默认的输入坐标数目

}

Coordinate(int times1)

{

times = times1;

cout << "Coordinate construction2 called!" << endl; // 设置输入坐标数目

}

~Coordinate()

{

cout << "Coordinate destruction called!" << endl;// 析构函数

}

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];

}// 输入坐标

}

void ShowCoord()

{

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

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

{

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

}// 显示已经输入的坐标

}

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;// 显示输入坐标的均值

}

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 x; // 定义对象

x.InputCoord();

x.ShowCoord();

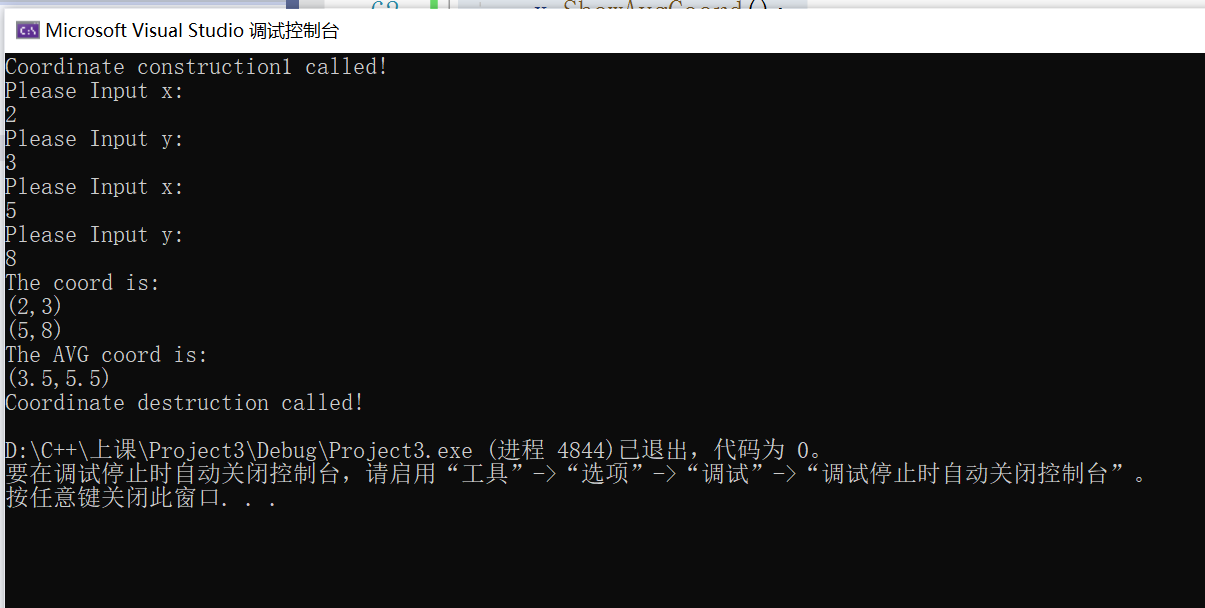
x.ShowAvgCoord();

return 0;

// 执行显示和坐标均值运算

}

运行结果：



* **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();**

代码：（与上面代码重复地方已省）

int main()

{ Coordinate y(5);

y.InputCoord();

y.ShowCoord();

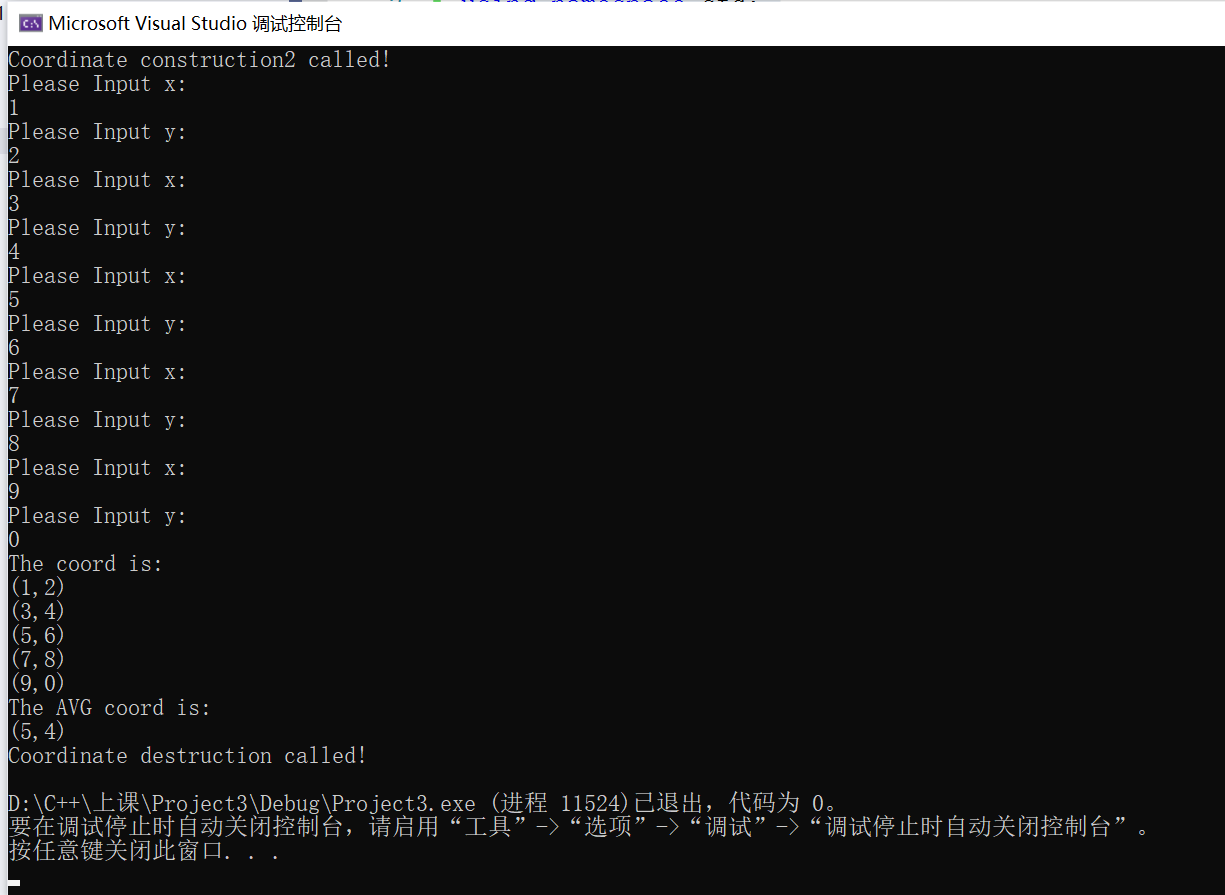
y.ShowAvgCoord();

return 0;

// 执行显示和坐标均值运算

}

**执行结果：**



实验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。**

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);

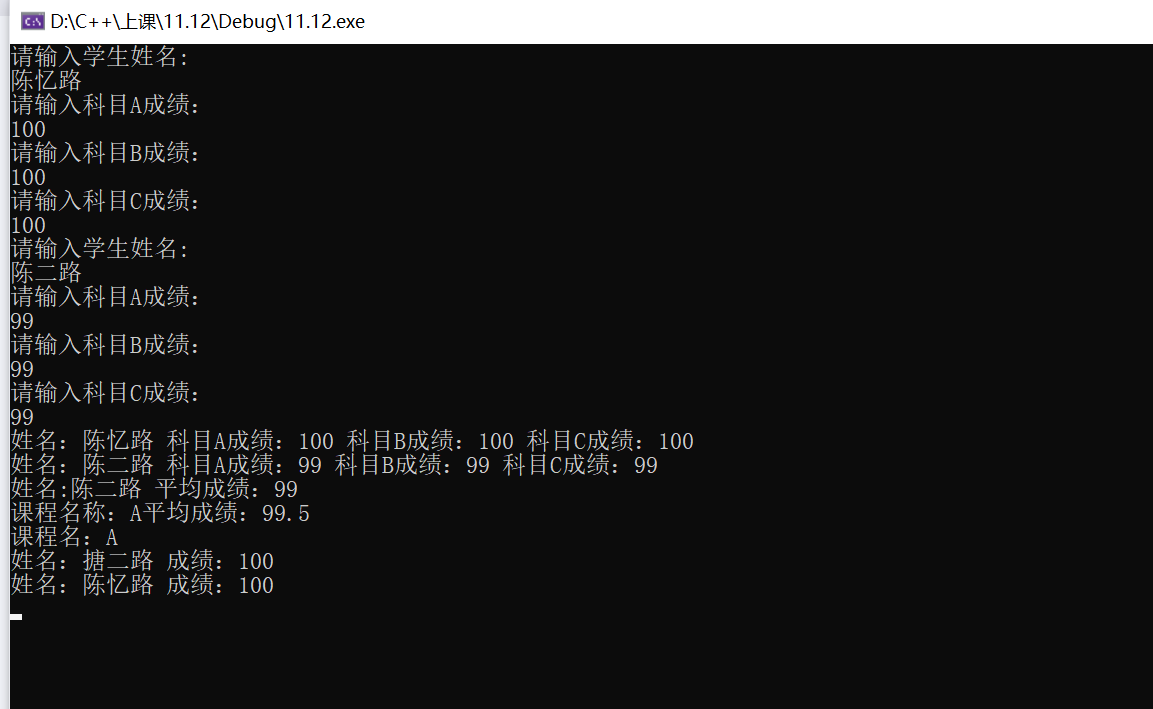
getchar();

getchar();

return 0;

}

**运行结果：**



**感想：**

本次实验我看了课本，又借助了csdn，但是在上机规定实践里还是没有完成。这是第一次接触类的程序，稍微有点陌生，编程起来有点吃力，不过下一次的上机看了老师的程序，有了点感觉。

Copyright ©2021-2099 Chen Yilu. All rights reserved