#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;

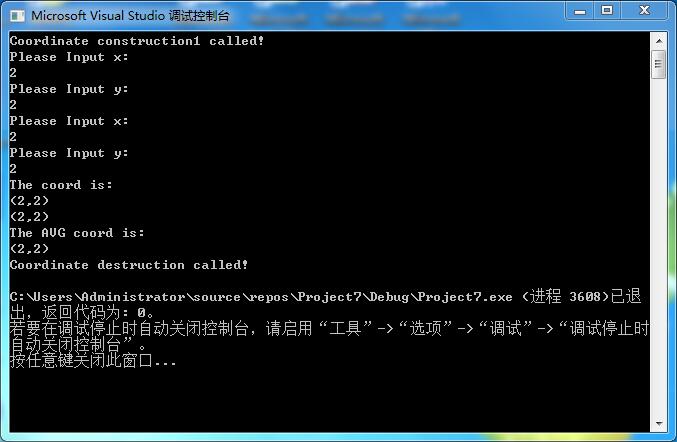
x.InputCoord();

x.ShowCoord();

x.ShowAvgCoord();

return 0;

}



#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);

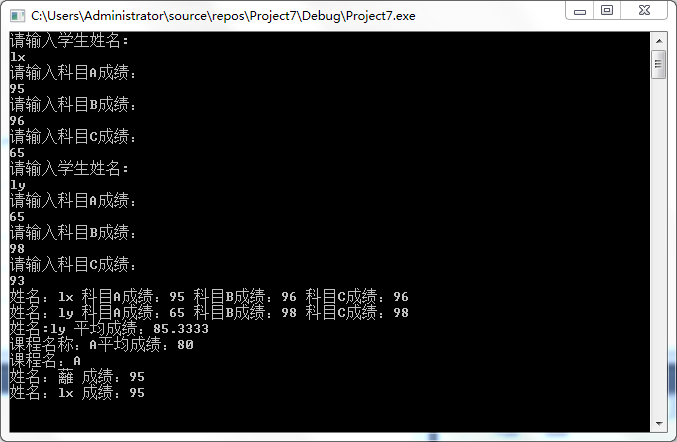
stu.rank(1);

getchar();

getchar();

return 0;

}



心得体会：本次实验了解了构造函数和析构函数的原理和执行顺序，内置类型的成员变量在初始化列表里初始化和在构造函数体里初始化的效率是一致的。类类型的成员变量在初始化列表里初始化，才是真正的初始化，在构造函数体里仅仅是赋值，析构函数名要与构造函数保持一致且函数前要加“~”等。

Copyright ©2021-2099 Wanghaohan.All rights reserved