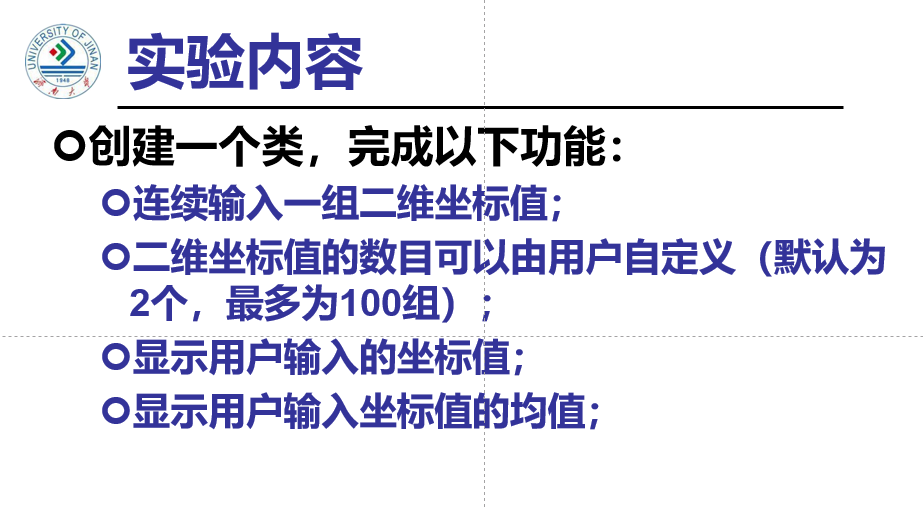
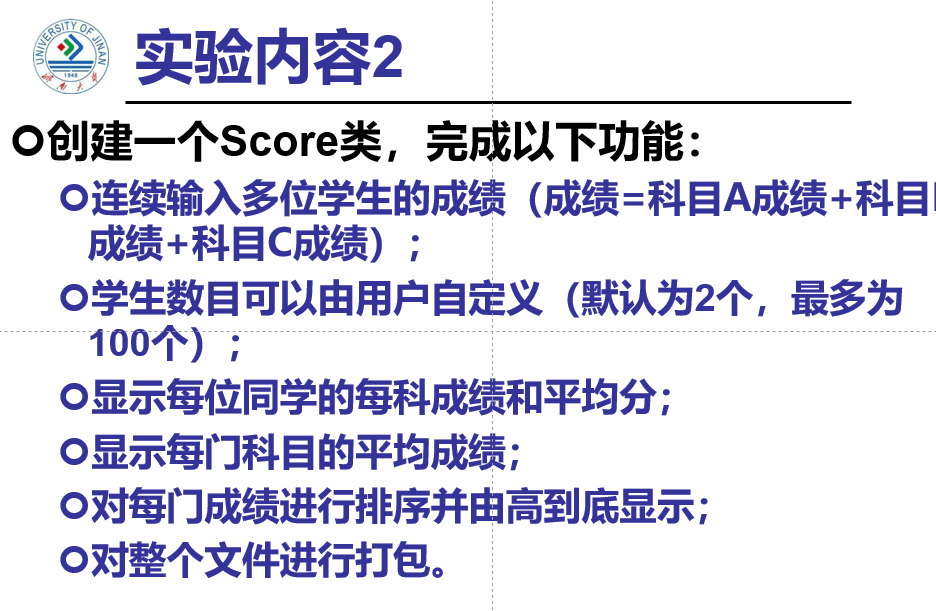
**第三次试验报告**

孙浩然 202031906076

Copyright ©2021-2099 HaoranSun. All rights reserved

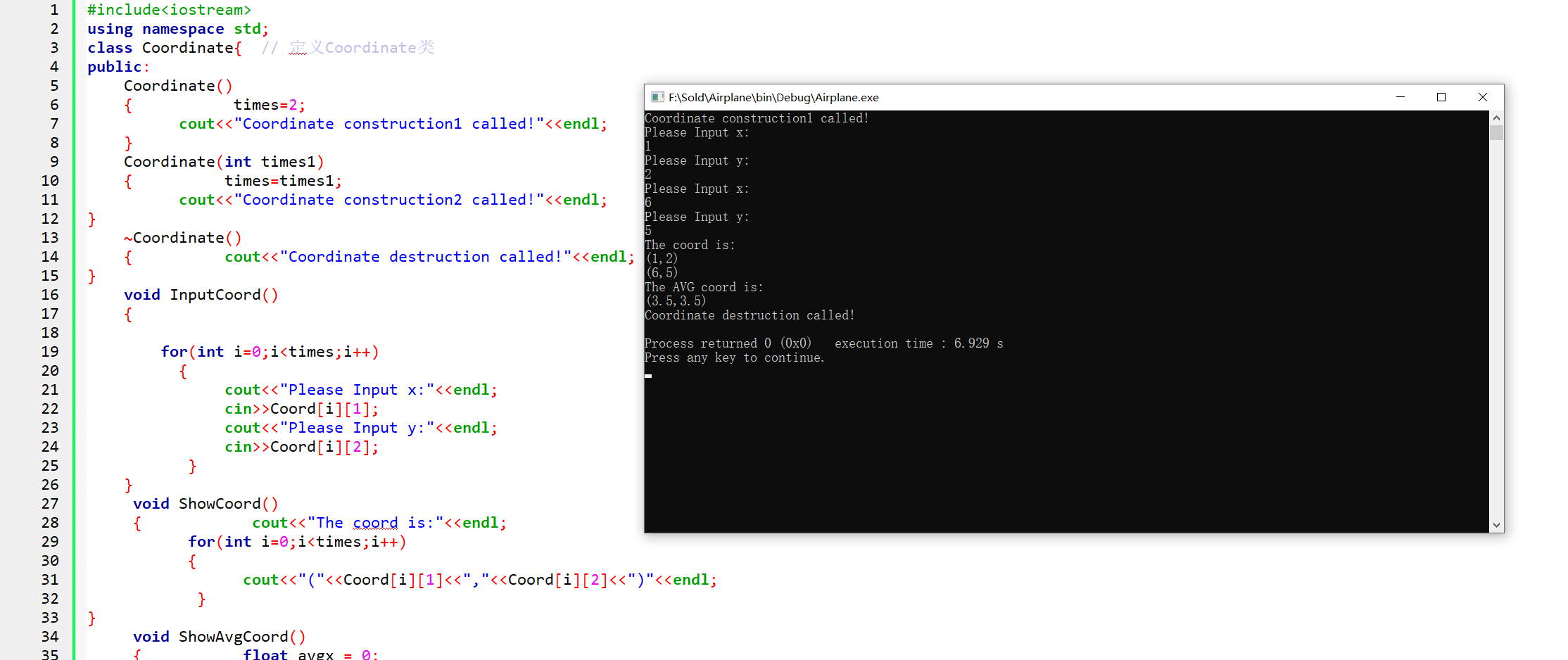
**一、实验要求**





**二、心得体会**

第三个试验的完成结果如下：





本次的实验是C++中面向对象编程的开始。这是我第一次使用对象的角度去思考问题，在之前的话总是想使用某个函数来实现某个功能，但是像这样的将数据和对应的函数放在一起的方式是第一次进行实现。在本次的实验过程中，我尝试定义了一个类并通过它的成员函数去进行一些运算，从数据的输入到距离的计算最终在main函数中调用都是采用了面向对象的实现方法。本次实验并不困难，重点在于基本概念的理解。在实现score类的时候，这一块的代码量还是不小的，但是整体的逻辑并不困难。整体锻炼了我的代码水平并让我尝试写出了交互界面友好的程序。

**三、实验代码与程序结果**

#include<iostream>

using namespace std;

class Coordinate{ // 定义Coordinate类

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]; // 存放输入坐标的数组

int times; // 存放输入坐标数目

};

int main()

{

Coordinate x;

x.InputCoord();

x.ShowCoord();

x.ShowAvgCoord();

return 0;

}

#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;

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

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

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

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

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

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

{ Name1[i] = Name[i]; }

for(int 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(int 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;

}

Copyright ©2021-2099 HaoranSun. All rights reserved