1. 目的：

一：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);  
yInputCoord();  
y.ShowCoord();  
y.ShowAvgCoord();  
二：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。

#include <iostream>

using namespace std;

class Score

{

public:

double A;

double B;

double C;

double ave;

string name;

}stu[100], z;

double all\_ave[5] = { 0 };

int x;

void sort()

{

char X;

cout << "选择排序科目：A B C" << endl;

cin >> X;

int i;

switch (X)

{

case 'A':

for (i = 0; i < x - 1; i++)

{

if (stu[i].A < stu[i + 1].A)

{

z = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = z;

}

}

break;

case 'B':

for (i = 0; i < x - 1; i++)

{

if (stu[i].B < stu[i + 1].B)

{

z = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = z;

}

}

break;

case 'C':

for (i = 0; i < x - 1; i++)

{

if (stu[i].C < stu[i + 1].C)

{

z = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = z;

}

}

break;

}

}

void print()

{

int i;

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

{

cout << "\n姓名：" << stu[i].name << "\n科目： A：" << stu[i].A << " 科目： B：" << stu[i].B << " 科目： C：" << stu[i].C << " 平均成绩：" << stu[i].ave << endl << endl;

}

}

void print\_ave()

{

cout << "\n科目A平均成绩：" << all\_ave[0] << " 科目B平均成绩：" << all\_ave[1] << " 科目C平均成绩：" << all\_ave[2] << endl << endl;

}

void menu()

{

int i;

while (1)

{

cout << "1.显示每位同学成绩和平均分\n2.显示每门科目平均成绩\n3.成绩排序\n4.退出\n" << endl;

cin >> i;

switch (i)

{

case 1:

print();

break;

case 2:

print\_ave();

break;

case 3:

sort();

print();

break;

}

if (i == 4)

break;

}

}

void input()

{

int i = 0, y = 0;

cout << "请输入学生数？(2-100)" << endl;

while (1)

{

cin >> x;

if (x > 100 || x < 2)

{

cout << "输入错误！" << endl;

}

else

break;

}

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

{

cout << "输入学生" << i + 1 << "姓名" << endl;

cin >> stu[i].name;

cout << "输入成绩A：";

cin >> stu[i].A;

cout << "输入成绩B：";

cin >> stu[i].B;

cout << "输入成绩C：";

cin >> stu[i].C;

stu[i].ave = (stu[i].A + stu[i].B + stu[i].C) / 3;

all\_ave[0] += stu[i].A / x;

all\_ave[1] += stu[i].B / x;

all\_ave[2] += stu[i].C / x;

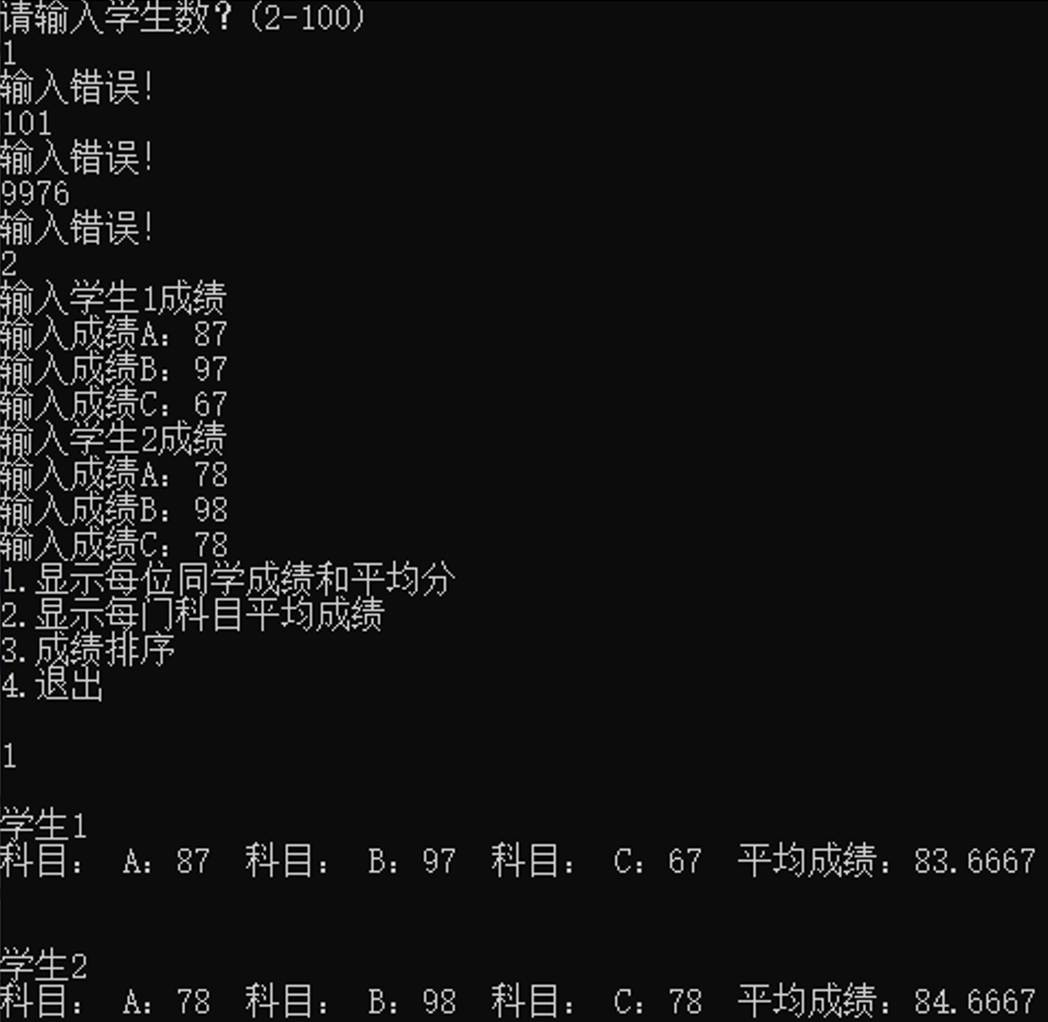
}

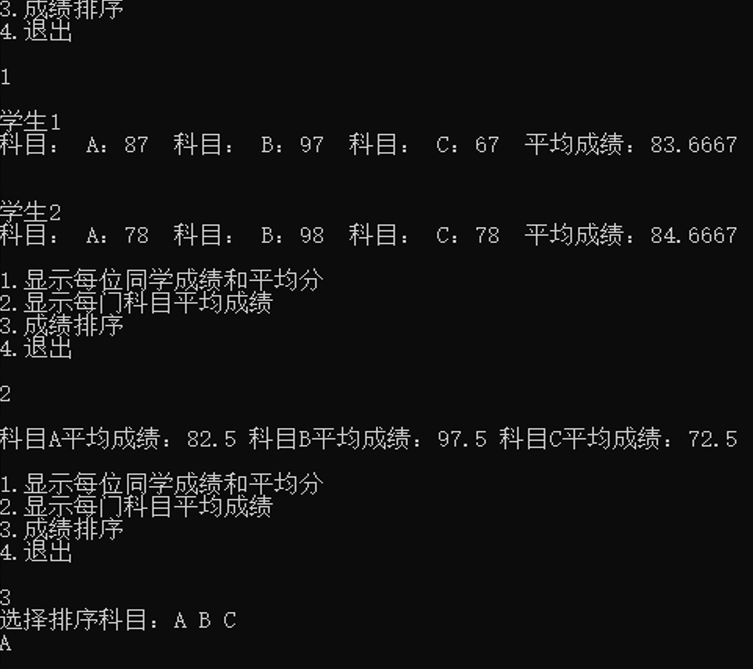
}

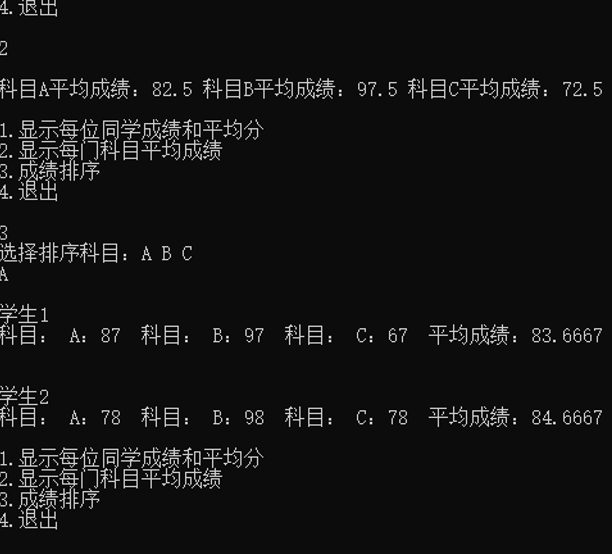
int main()

{

input();

menu





感想：

1. 明白了如何对类进行声明，以及类中private,public对应用范围和条件。
2. 理解了构造函数和析构函数的定义，以及一些它们的特点。
3. 明白了对象的定义和对象如何对函数成员进行访问。

Copyright ©2021-2099 hukun. All rights reserved