**构造函数与析构函数**

**一、实验程序**

1、

#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 y(5);

y.InputCoord();

y.ShowCoord();

y.ShowAvgCoord();

return 0;

}

2、

#include <iostream>

#include <string>

using namespace std;

class score

{

private:

int time;

double cj[100][100];

string name[100];

string copy[100];

public:

score()

{

time = 2;//默认为2

cout << "默认两组" << endl;

}

score(int time1)

{

time = time1;

cout << "用户自定义组数" << endl;

}

~score()

{

cout << "Destructor called" << endl;

}

void input()

{

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

{

cout << "请输入学生的姓名和成绩：" << endl;

cin >> name[i];

cin >> cj[i][1] >> cj[i][2] >> cj[i][3];

}

}

void show()

{

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

{

cout << name[i] << "学科A的成绩为：" << cj[i][1] << " ";

cout << name[i] << "学科B的成绩为：" << cj[i][2] << " ";

cout << name[i] << "学科C的成绩为：" << cj[i][3] << endl;

}

}

void avg()

{

double a = 0;

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

{

a = cj[i][1] + cj[i][2] + cj[i][3];

cout << name[i] << "的平均成绩为：" << a / 3 << " ";

}

cout << endl;

}

void showavg()

{

double a = 0;

double b = 0;

double c = 0;

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

a = a + cj[i][1];

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

b = b + cj[i][2];

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

c = c + cj[i][3];

cout << "学科A的平均成绩为" << a / time << " ";

cout << "学科B的平均成绩为" << b / time << " ";

cout << "学科C的平均成绩为" << c / time << endl;

}

void px()

{

for (int i = 0; i < time; ++i) //将name数组复制一个

copy[i] = name[i];

for (int i = 0; i < time - 1; i++)

for (int j = 0; j < time - i - 1; j++)

if (cj[j][1] < cj[j + 1][1]) //改变了name数组数据的位置

{

double temp = cj[j + 1][1];

cj[j + 1][1] = cj[j][1];

cj[j][1] = temp;

string t = name[j + 1];

name[j + 1] = name[j];

name[j] = t;

}

cout << "学科A的排序为：" << endl;

for (int i = 0; i < time; ++i) //打印姓名和对应的成绩

{

cout << name[i] << " " << cj[i][1] << endl;

}

for (int i = 0; i < time; ++i) //重置name数组

name[i] = copy[i];

for (int i = 0; i < time - 1; i++)

for (int j = 0; j < time - i - 1; j++)

if (cj[j][2] < cj[j + 1][2])

{

double temp = cj[j + 1][2];

cj[j + 1][2] = cj[j][2];

cj[j][2] = temp;

string t = name[j + 1];

name[j + 1] = name[j];

name[j] = t；

}

cout << "学科B的排序为：" << endl;

for (int i = 0; i < time; ++i) //打印姓名和对应的成绩

{

cout << name[i] << " " << cj[i][2] << endl;

}

for (int i = 0; i < time; ++i) //重置name数组

name[i] = copy[i];

for (int i = 0; i < time - 1; i++)

for (int j = 0; j < time - i - 1; j++)

if (cj[j][3] < cj[j + 1][3])

{

double temp = cj[j + 1][3];

cj[j + 1][3] = cj[j][3];

cj[j][3] = temp;

string t = name[j + 1];

name[j + 1] = name[j];

name[j] = t;

}

cout << "学科C的排序为：" << endl; //打印姓名和对应的成绩

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

{

cout << name[i] << " " << cj[i][3] << endl;

}

}

};

int main()

{

score a(3);

a.input();

a.show();

a.avg();

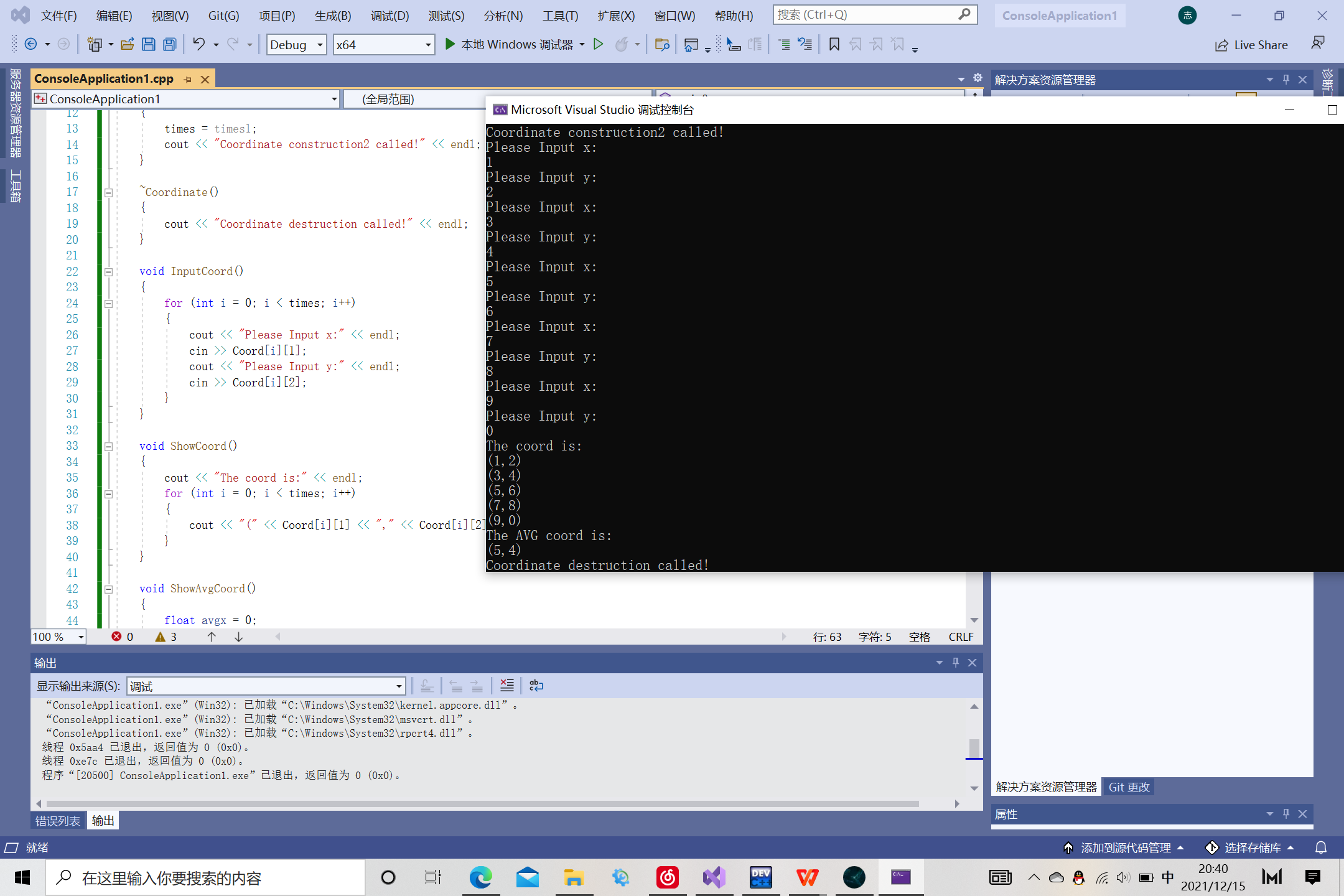
a.showavg();

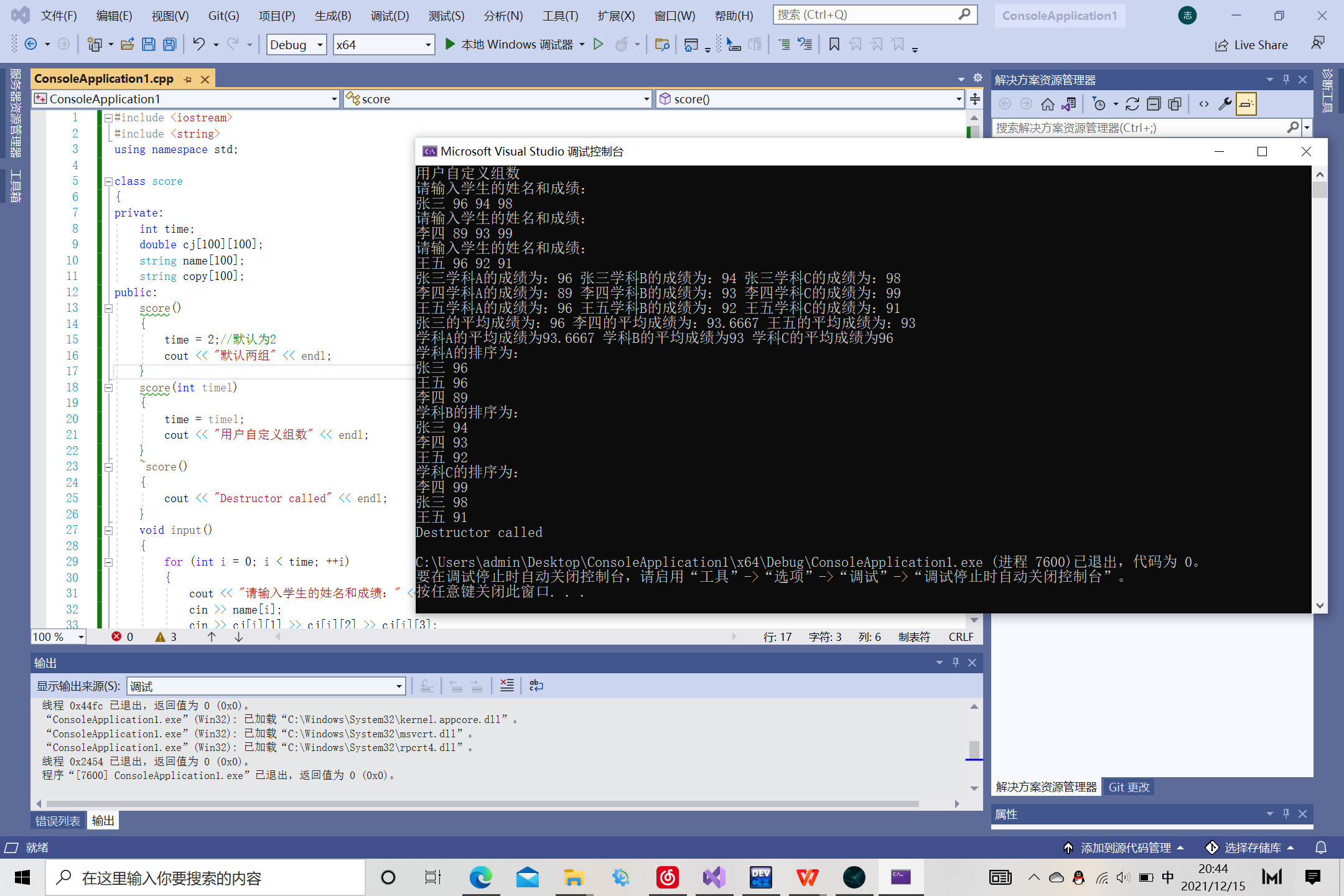
a.px();

return 0;

}

**二、实验结果**



i

**三、心得体会**

通过掌握构造函数和析构函数这两个c＋＋程序设计中最基础的两个函数，学习了怎样在c＋＋中使用这些基本的函数。为以后的程序设计打下基础。而这次的上机实验中的实际问题也和我们的平时生活密切相关。既起到了学习知识的作用，也极大的引起我们的兴趣。

Copyright ©2021-2099 电自2003班 陈阳202030310072. All rights reserved