# 实验五

5-1

#include<iostream>   
using namespace std;   
class Time // 定义Time类   
{   
private: // 数据成员为公用的   
 int hour;   
 int minute;   
 int sec;   
public:   
 Time() {   
 hour = 0;   
 minute = 0;   
 sec = 0;   
 }   
 Time(int newHour, int newMinute, int newSec)   
 {   
 hour = newHour;   
 minute = newMinute;   
 sec = newSec;   
 }   
 void inputTime() {   
 cout << "please enter hour:";   
 cin >> hour;   
 cout << "please enter minute:";   
 cin >> minute;   
 cout << "please enter second:";   
 cin >> sec;   
 }   
 void displayTime() const {   
 cout << hour << ":" << minute << ":" << sec << endl;   
 }   
};   
int main()   
{   
 Time t1; //定义t1为Time类对象   
 t1.inputTime();   
 t1.displayTime();   
 return 0;   
}   
//

5-2 student.h

#pragma once   
   
class Student //类声明   
{   
public:   
 void set\_value(int n, const char\* str, char s);//公用成员函数原型声明   
 void display();   
private:   
 int num;   
 char name[20];   
 char sex;   
};

5-2 student.cpp

#include <iostream>   
#include <cstring>   
#include "student.h"   
using namespace std;   
void Student::set\_value(int n, const char\* str, char s) {   
 num = n;   
   
 // 使用 strncpy\_s 安全地复制字符串   
 strncpy\_s(name, sizeof(name), str, sizeof(name) - 1);   
 name[sizeof(name) - 1] = '\0'; // 确保字符串以 '\0' 结尾   
   
 sex = s;   
}   
void Student::display() {   
 cout << "num: " << num << endl;   
 cout << "name: " << name << endl;   
 cout << "sex:" << sex << endl;   
}

5-2 main.cpp

#include <iostream>   
#include "student.h"   
using namespace std;   
   
int main()   
{   
 Student student1;   
 student1.set\_value(7, "tcg", 'm');   
 student1.display();   
 return 0;   
}

5-3

#include <iostream>   
using namespace std;   
class Cuboid {   
private:   
 double length;   
 double width;   
 double height;   
public:   
 Cuboid() {   
 length = 0;   
 width = 0;   
 height = 0;   
 }   
 //长方体尺寸   
 void inputCuboid() {   
 cout << "Please enter length:";   
 cin >> length;   
 cout << "Please enter width:";   
 cin >> width;   
 cout << "Please enter height:";   
 cin >> height;   
 }   
 //计算体积   
 double calculate() const {   
 return length \* width \* height;   
 }   
};   
int main()   
{   
 Cuboid Cb[4];   
 for (int i = 1; i <= 3; i++)   
 {   
 cout << "输入第" << i << "个长方体尺寸：" << endl;   
 Cb[i].inputCuboid();   
 cout << "第" << i << "个长方体体积为:" << Cb[i].calculate() << endl;   
 }   
 return 0;   
}

5-4

#include <iostream>   
using namespace std;   
   
class Student {   
private:   
 int id;   
 double score;   
public:   
 Student() {   
 id = 0;   
 score = 0;   
 }   
 void input() {   
 cout << "Please enter id:";   
 cin >> id;   
 cout << "Please enter score:";   
 cin >> score;   
 }   
 int getId() const   
 {   
 return score;   
 }   
 double getScore() const {   
 return score;   
 }   
};   
void max(Student\* students, int n) {   
 Student\* maxStudent = students;   
 for (int i = 1; i <= n; i++)   
 {   
 if (students[i].getScore() > \   
 maxStudent->getScore())   
 {   
 maxStudent = &students[i];   
 }   
 }   
 cout << "The student with the best grades is:"\   
 <<maxStudent->getId() << endl;   
}   
   
int main()   
{   
 int n = 5;   
 Student students[5];   
   
 //输入信息   
 for (int i = 1; i <= n; i++)   
 {   
 cout << "Please enter Student " << i << " :";   
 students[i-1].input();   
 }   
 //处理   
 max(students, n);   
 return 0;   
}

5-5

#include <iostream>   
using namespace std;   
   
class Point {   
private:   
 int x, y;   
public:   
 Point(int a=60,int b=80) {   
 x = a;   
 y = b;   
 }   
 //设置坐标点   
 void setPoint(int i, int j) {   
 x += i;   
 y += j;   
 }   
 //显示坐标   
 void display() const {   
 cout << "坐标：(" << x << "," << y << ")" << endl;   
   
 }   
};   
   
int main()   
{   
 Point p1;   
 p1.display();   
   
 int x1, y1;   
 cout << "输入改变坐标：";   
 cin >> x1 >> y1;   
 p1.setPoint(x1, y1);   
 //输出   
 p1.display();   
 return 0;   
}

四、遇到的问题与解决方案

在5-2中对与函数头文件的编写比较陌生，在查阅资料后，回顾了头文件的使用。1. 创建后缀名为.h的头文件 2. 创建后缀名为.cpp的源文件3. 在头文件中写函数的声明4. 在源文件中写函数的定义。

在实验5-2中，字符串作为函数参数进行复制时出现了问题，查阅资料得知 strncpy 函数被认为是“不安全”的，因为它可能导致字符串未以 \0 结尾（如果源字符串超出了目标数组的大小，从而引发潜在的缓冲区溢出问题。替换为了strncpy\_s函数。

五、体会与总结

本次实验对于面向对象编程做了练习与回顾，同时温习了函数的分文件编写，还学习到了数组作为一个类来进行访问，加深了对于C++面向对象编程相关知识的巩固。