电子信息与工程学院

姓名	学号	专业	联系方式
黄炜恺	17311023	微电子科学与工程	13725951349

第一题

代码

```cpp
#include <iostream>
#include <string>
using namespace std;

class Teacher
{
public:
    Teacher(string nam,int ag,string se,string add,string tele,string tit,string ms);
    void display();
protected:
    string name;
    int age;
    string sex;
    string address;
    string telephone;
    string title;
    string marital_status;
};

Teacher::Teacher(string nam,int ag,string se,string add,string tele,string tit,string ms):
        name(nam),age(ag),sex(se),address(add),telephone(tele),title(tit),marital_status(ms){}

void Teacher::display()
{
    cout << "name:        " << name << endl
         << "age:         " << age  << endl
         << "sex:         " << sex  << endl
         << "address:     " << address  << endl
         << "telephone:   " << telephone  << endl
         << "title:       " << title  << endl;
}

```

```cpp
class Cadre
{
public:
    Cadre(string nam,int ag,string se,string add,string tele,string po,string ms);
    void display();
protected:
    string name;
    int age;
    string sex;
    string address;
    string telephone;
    string post;
    string marital_status;
};

Cadre::Cadre(string nam,int ag,string se,string add,string tele,string po,string ms):
        name(nam),age(ag),sex(se),address(add),telephone(tele),post(po),marital_status(ms){}

void Cadre::display()
{
    cout << "name:       " << name << endl
         << "age:        " << age << endl
         << "sex:        " << sex << endl
         << "address:    " << address << endl
         << "telephone:  " << telephone  << endl
         << "post:       " << post << endl;
}

class Teacher_Cadre:public Teacher,public Cadre
{
public:
    Teacher_Cadre(string nam,int ag,string se,string add,string tele,string tit,string po,string ms,int wa);
    void show();
private:
    int wages;
};

Teacher_Cadre::Teacher_Cadre(string nam,int ag,string se,string add,string tele,string tit,string po,string ms,int wa):
        Teacher(nam,ag,se,add,tele,tit,ms),Cadre(nam,ag,se,add,tele,po,ms),wages(wa){}

void Teacher_Cadre::show()
{
    Teacher::display();
    cout << "post:       " << post << endl
         << "wages:      " << wages << endl
         << "marital status: " << Teacher::marital_status << endl;
}

int main()
{
    Teacher_Cadre Thisguy("Finch",64,"male","Baker Street 221B","30624700","professor","chairman","married",15000);
    Thisguy.show();
    return 0;
}
```

分析

1、string 的用法

使用等号的初始化叫做拷贝初始化，不使用等号的初始化叫做直接初始化。
构造函数：　string();
string(size_type length, char ch);
以 length 为长度的 ch 的拷贝（即 length 个 ch）
string(const char *str);
以 str 为初值 (长度任意)
string(const char *str, size_type length);
复制字符串 str 的前 length 个字符到 s 当中
string(string &str, size_type index, size_type length);
以 index 为索引开始的子串，长度为 length

2、二义性问题

Teacher 类和 Cadre 类中都有数据成员 marital_status，Teacher_Cadre 类是 Teacher 类和 Cadre 类的直接派生类；当 Teacher_Cadre 类要引用数据成员 marital_status 时，必须指明要访问的是哪一个基类的成员，如 Teacher::marital_status。

3、多重继承派生类的构造函数

Teacher_Cadre::Teacher_Cadre(string nam,int ag,string se,string add,string tele,string tit,string po,string ms,int wa):

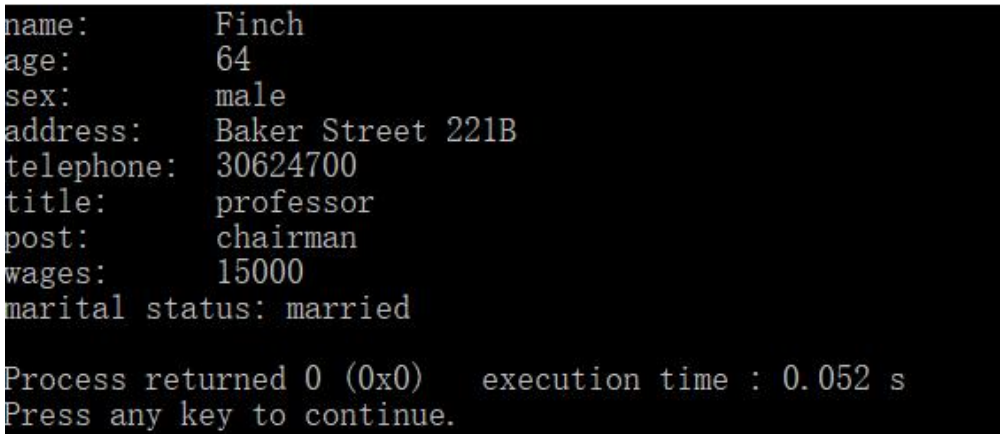
Teacher(nam,ag,se,add,tele,tit,ms),Cadre(nam,ag,se,add,tele,po,ms){wages = wa;}

或

Teacher_Cadre::Teacher_Cadre(string nam,int ag,string se,string add,string tele,string tit,string po,string ms,int wa):

Teacher(nam,ag,se,add,tele,tit,ms),Cadre(nam,ag,se,add,tele,po,ms),wages(wa){}

运行结果截图


name: Finch
age: 64
sex: male
address: Baker Street 221B
telephone: 30624700
title: professor
post: chairman
wages: 15000
marital status: married
Process returned 0 (0x0) execution time : 0.052 s
Press any key to continue.