

Solution2

191220008 陈南瞳

概念题

1、构造函数的成员初始化列表的作用是什么？

对于常量和引用数据成员，不能在声明时初始化，也不能采用赋值操作在构造函数中对它们进行初始化。成员初始化表可以用来对常量和引用数据成员进行初始化。成员初始化表中成员的书写顺序并不决定它们的初始化次序，它们的初始化次序由它们在类定义中的描述次序来决定。

2、C++的成员对象是什么？当创建包含成员对象的类的对象时，构造函数的调用顺序是什么样的？

成员对象：当一个类的某个成员是另一个类的对象时，这个对象就叫做成员对象。当创建包含成员对象的类的对象时，先执行成员对象自己所属类的构造函数，当全部成员对象都执行了自身类的构造函数后，再执行当前类的构造函数。

编程题

1、定义一个时间类Time，它能表示时、分、秒，并提供以下操作：

```
class Time
{
public:
    Time(int h, int m, int s);
    void set(int h, int m, int s);
    void increment();
    void display();
    bool equal(Time& other_time);
    bool less_than(Time& other_time);
private:
    int hour, min, sec;
};

Time::Time(int h, int m, int s)
{
    hour = h;
    min = m;
    sec = s;
}

void Time::set(int h, int m, int s)
{
    hour = h;
    min = m;
```

```

        sec = s;
    }

    void Time::increment()
    {
        if (sec == 59)
        {
            if (min == 59)
            {
                if (hour == 23)
                    hour = min = sec = 0;
                else
                {
                    hour++;
                    min = sec = 0;
                }
            }
            else
            {
                min++;
                sec = 0;
            }
        }
        else
            sec++;
    }

    void Time::display()
    {
        cout << "The current time is " << hour << ":" << min << ":" << sec << endl;
    }

    bool Time::equal(Time& other_time)
    {
        return other_time.hour == hour && other_time.min == min && other_time.sec ==
sec;
    }

    bool Time::less_than(Time& other_time)
    {
        int sub_hour, sub_min, sub_sec;
        sub_hour = other_time.hour - hour;
        sub_min = other_time.min - min;
        sub_sec = other_time.sec - sec;
        if (sub_hour > 0 || sub_hour == 0 && sub_min > 0 || sub_hour == 0 && sub_min
== 0 && sub_sec > 0)
            return true;
        else
            return false;
    }

```

