实验一　C++类与对象

一、目的和要求

1. 掌握类、对象的基本概念，理解类中成员的访问权限，正确理解类与结构体的异同；
2. 学习对象的说明和使用方法，掌握构造函数、析构函数的工作原理；
3. 了解 I/O 流类的层次结构，掌握 C++标准输入输出流的用法，能够使用操纵算子格式化输入输出；
4. 掌握文件流的用法。

二、实验内容

（一）验证性实验

1．学生通讯录

如下定义了一个描述学生通讯录的类，数据成员包括：姓名、学校、电话号码和邮编；函数成员包括：输出各个数据成员的值，分别设置和获取各个数据成员的值。

[程序]

根据题目的要求，学生的属性有 4 个，在此把姓名、学校、电话号码和邮编都定义成 string 类型，将这些数据成员均定义为私有的。用一个成员函数输出所有的成员数据，用四个成员函数分别设置姓名、单位、电话号码和邮编，再用四个成员函数分别获取姓名、单位、电话号码和邮编。主函数完成简单的测试工作。

其完整的参考程序如下：

```
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;

class   COMMU
{
    string   pName;                    //姓名，数据成员为私有的
    string   pSchool;                  //单位
    string   pNum;                     //电话号码
    string   Box;                      //邮编
public:
    void   Print(void)                 //输出数据成员
    {
        cout << "姓名:" << pName << "\t";
        cout << "单位:" << pSchool << "\t";
        cout << "电话号码:" << pNum << "\t";
        cout << "邮编:" << Box << endl;
    }
    void   Init(string, string, string, string);
    void   SetName(string name)
    {
        pName = name;
    }
    void SetScool(string unit)         //设置学校名称
```

```
    {
        pSchool = unit;
    }
    void SetNum(string num)                 //设置电话号码
    {
        pNum = num;
    }
    void SetBox(string mailnum)             //设置邮编
    {
        Box = mailnum;
    }
    string GetName(void)                    //取姓名
    {
        return pName;
    }
    string GetScool(void )                  //取学校
    {
        return    pSchool;
    }
    string GetNum(void)                     //取电话号码
    {
        return pNum;
    }
    string GetBox(void)                     //取邮编
    {
        return Box;
    }
};

void    COMMU::Init(string name, string unit, string num, string b)
{                                           //完成初始化
    pName = name;
    pSchool = unit;
    pNum = num;
    Box = b;
}

int main()
{
    COMMU    c1,c2;
    c1.Init("王元","上海大学","021-66138861","200444");
    c2.Init("于海","上海大学","021-66138862","200444");
    c1.Print();
    c2.Print();
    c1.SetName("王国安");
    cout << c1.GetName()<<'\n';
    c1.SetScool("复旦大学");
    cout << c1.GetScool()<<'\n';
    c1.SetNum("021-66138960");
    cout << c1.GetNum()<<"\n";
    c1.SetBox("200201");
    cout<<c1.GetBox()<<"\n";
    c1.Print();

    system("pause");
    return 0;
```

}

[思考]

- 用以下数据测试程序的正确性：
  名字改为“李明明”，并输出；
  学校改为“上海交通大学”并输出；
  电话改为“021—55667878”，并输出；
  邮编改为“200108”并输出。
- 将成员函数的权限改为私有的，观察现象。
- 将类改为结构体，重新执行程序，输出成员数据。
- 增加数据成员：邮件地址，手机号码，并将电话号码分为住宅电话和办公电话。

2．图书信息管理

如下设计了一个管理图书的简单程序，描述一本书的信息包括：书号，书名，出版社和作者等。提供的基本功能包括：可连续将新书存入文件“book.dat”中，新书信息加入到文件的尾部；也可以根据输入的书名进行查找。

[程序]

可以把描述一本书的信息定义为一个 Book 类，它包含必要的成员函数。题目要求把加入的新书总是添加到文件尾部，因此以增补方式打开输出文件。从文件中查找书时，总是从文件开始位置查找，因此以读方式打开文件。用一个循环语句实现可连续地将新书加入文件或从文件中查找指定的书名。由于是以一个 Book 类的实例进行文件输入输出，所以这文件的类型应该是二进制文件。

参考程序如下：

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <fstream>
using namespace std;

class   Book
{
    long int    num;                                    //书号
    char bookname[40];                                  //书名
    char publicname[40];                                //出版社
    char name[20];                                      //作者
public:
    Book(){ num=0; bookname[0] =0;publicname[0] =0; name[0] =0;}
    char * Getbookname(void)     { return bookname ;}
    long    Getnum(void ) { return num;}
    void Setdata(long , char *,char *,char *);
    void Show(void );
    Book(long , char *,char *,char *);
};
void    Book::Setdata(long nu , char *bn,char *p,char *n)
{
    num = nu; strcpy(bookname,bn);
    strcpy(publicname,p); strcpy(name,n);
}
void    Book::Show(void )
{
    cout<<"书号:"<<num<<'\t'<<"书名:"<<bookname<<'\t';
    cout<<"出版社:"<<publicname<<'\t'<<"作者:"<<name<<'\n';
}
Book::Book(long nu, char * bp,char *p,char *n)
{
```

```
    Setdata(nu , bp, p, n);
}
int   main(void)
{
    Book   b1,b2;
    long   nu;
    char bn[40];                                        //书名
    char pn[40];                                        //出版社
    char na[20];                                        //作者
    ifstream   file1;
    ofstream   file3;
    char flag = 'y';
    while( flag=='y' ||flag=='Y') //由 flag 控制循环
    {
        cout<<"\t\t 1: 按书名查找一本书!\n";
        cout<<"\t\t 2: 加入一本新书!\n";
        cout<<"\t\t 3: 退出!\n 输入选择：";
        int f;
        cin>>f;
        switch(f)
        {
        case 1:
            cout<<"输入要查找的书名："; cin>>bn;
            file1.open("book.dat",ios::in | ios::binary);//按读方式打开文件
            while(!file1.eof())
            {
                int n;
                file1.read((char *)&b1,sizeof(Book));
                n=file1.gcount();
                if(n==sizeof(Book))
                {
                    if(strcmp(b1.Getbookname(),bn)==0)   //显示书的信息
                        b1.Show();
                }
            }
            file1.close();
            break;
        case 2:
            cout<<"输入书号："; cin>>nu;
            cout<<"输入书名："; cin>>bn;
            cout<<"输入出版社："; cin>>pn;
            cout<<"输入作者："; cin>>na;
            b1.Setdata(nu,bn,pn,na);
            file3.open("book.dat",ios::app|ios::binary);//增补方式打开文件
            file3.write((char*)&b1,sizeof(b1));
            file3.close();
            break;
        default:   flag = 'n';
        }
    }
    system("pause");
    return 0;
}
```

[思考]修改程序增加

● 按出版社、作者从文件中查找书名；

- 输出同一出版社出版的所有书名；
- 按列表的格式输出文件中所有图书的信息。

（二）设计性实验

3．猴子分桃

有一堆桃子和甲、乙两组猴子，甲组有 3 只猴子，乙组有 5 只猴子。甲组的猴子先看到这堆桃子。第一只猴子把桃分成了相等的 3 堆，多出 2 个。于是，它吃掉了 2 个，又拿走了 1 堆；第二只猴子把 2 堆桃子合在一起，又分成相等的 3 堆，又多出 2 个。于是，它也吃掉了 2 个，拿走了 1 堆；第三只猴子也照样办理。甲组猴子走后，乙组猴子也看到剩下的 2 堆桃子，第一只猴子把桃分成了相等的 5 堆，多出 1 个。于是，它吃掉了 1 个，又拿走了 1 堆；第二只猴子把 4 堆桃子合在一起，又分成相等的 5 堆，又多出 1 个。于是，它也吃掉了 1 个，拿走了 1 堆；第三、四、五只猴子也照样办理。请设计算法，求：8 只猴子走后至少还剩下多少个桃子？原来这堆桃至少有多少个？

4．圆盘找数

设计一个圆盘类 Circle，它包括 20 个随机数组成一个圆（首尾相接），分别求出连续的数之和最大的 4 个数以及连续的数之和最小的 4 个数，能够输出整个序列（20 个随机数）、最大、最小的 4 个连续数及其和。

5．人员信息

设计一个有关人的基类，它有姓名、性别、年龄等属性，再由基类派生出教师类和学生类，教师类增加工号、职称和工资等属性；学生类增加学号、班级、专业和入学成绩等属性。并定义相应的方法设置和修改相应的属性。

6．学生信息管理

设计一个简单的学生信息管理程序，信息包括：学号、姓名、性别、年龄、班级等。基本功能包括：可以将信息存入文件或从文件中读出，可以添加、删除、修改学生信息，可以根据学号、姓名从文件中查找学生信息。

（三）综合性实验

7．静态成员函数和成员函数

[问题描述]

某商店经销一种货物，货物成箱购进，成箱卖出，购进和卖出时以重量为单位，各箱的重量不一样。因此，商店需要记下目前库存货物的总量，现在要求把商店货物购进和卖出的情况模拟出来。

[基本要求]

在程序中将商品看成是一个个对象，建立一个商品类。设置类的静态数据成员用以表示总重量。分别用静态成员函数 GetTotal()和成员函数 Get()来访问总重量和单个对象重量。

[测试数据]

由读者自己确定。

8．类的派生

[问题描述]

给出下面的基类：

```
Class AreaClass
 {
  public:
     //…
  protected:
     double   height;
     double   width;
  };
```

建立两个派生类 Box（方形类）与 Isosceles（等腰三角形类），让每个派生类包含一个函数 Area()，分别用来返回方形与等腰三角形的面积。用参数化构造函数对高 height 与宽 width 进行初始化。

[基本要求]

两个派生类的数据成员分别有不同的构造函数，但初始化的内容一样，都是给 height 和 width 赋初值。两个派生类的 Area()成员函数是计算其面积，不必专门为其设计输出成员函数。

[测试数据]

由读者自己确定。

9. 类的继承

[问题描述]

设计一个基类 Building 用来存储一座楼房的层数、房间数以及它的总平方米数。建立派生类 Housing，继承 Building，并存储下述内容：卧室和浴室的数量。另外，建立派生类 OfficeBuilding，继承 Building，并存储灭火器和电话的数目。

住宅楼是楼房的一种，办公楼也是楼房的一种。它们都有房间数、楼层数和总面积数。此外住宅楼还有卧室数和卫生间数，办公楼还有电话机数和灭火器数等。

[基本要求]

建立住宅楼的对象和办公楼对象。分别显示住宅楼的和办公楼的一应数据，可以在其基类楼房的显示动作基础上，再专门显示各自自己的相应数据。要求在住宅楼的输出成员函数和办公楼的输出成员函数中实现。

[测试数据]

可设定 Housing 的如下数据：
floors:5
rooms: 7
total area:200
bathrooms: 2
可设定 OfficeBuilding 的如下数据：
floors:8
rooms:12
total area:800
phones: 12

extinguishers:2

10.模板

[问题描述]

对于基本数据类型，两个数据对象值是是可以比较的，但对于类类型的对象是不可以直接比较的。例如，复数类，由于 complex 类中没有重载大于（>）的操作运算，两个复数对象的值是不可能直接进行大于比较。为了使两个复数对象的值可比，可以定义自己的 MyComplex 类继承复数类。增加大于（>）操作符。同时重载构造函数以适应 MyComplex 类构造对象的方式。

[基本要求]

为了使大于操作灵活，定义大于（>）的友元。由于 MyComplex 类对象继承了 complex 类，也可以进行 abs()求绝对值的操作，所以在友元定义中，将大于操作定义为两个对象的绝对值大于比较。

[测试数据]

读者可依照所编的程序，在程序中设定测试数据或用输入语句输入测试数据。