

实验前言（**建议认真阅读**）

一、上机实验的目的

上机实验的目的是提高学生的分析问题、解决问题的能力 and 动手能力，通过实践环节理解 C++ 语言的基本结构和程序设计方法。通过亲手编程掌握 C++ 语言编程的方法。

二、实验基本内容

为了使学生在上机实验时目标明确，本实验指导书针对课程内容编写了五个实验。学生在课内机时先完成指导书中标有“*”号的程序，理解所学的知识，在此基础上再编写其他应用程序。

指导书中的五个实验如下：

1. 函数与编译处理
2. 指针与引用
3. 类和对象
4. 继承和派生类
5. 多态性

三、实验提交（**特别提醒**）

1. 实验题目以附件形式提交，每个附件限一个源程序，**附件以实验编号-题号-自己学号.cpp** 命名；如学号为 10051234 的同学提交实验 1 的第 5 题，则附件命名方式为：**1-5-10051234.cpp**，**错误提交的将视为未提交。**
2. 提交时间定为该次实验课的一周内，过期不再接收。

四、评分和作弊处理

1. 要求提交实验题总数为 6 题（带“*”号），占期末总评成绩的 20%；
2. 存在抄袭和被抄袭的作业，无论抄袭与被抄袭者，本门课程平时成绩以**零**分记。存在两次及两次以上抄袭和被抄袭者，本门课程以**不及格**记。

五、实验课要求（**重点强调**）

1. 严禁 QQ、MSN 等即时软件工具聊天；
2. 严禁接听手机、发送短信；
3. 严禁玩游戏。

实验 1 函数与编译处理

一、实验目的

1. 掌握函数声明、定义和使用的方法；
2. 掌握形参与实参之间的对应关系；
3. 掌握函数调用时，形参、实参之间的“值传递”和“引用传递”的区别；
4. 掌握函数递归调用的方法；
5. 掌握全局变量、局部变量、静态变量的使用方法；
6. 掌握文件包含的使用方法。

二、实验内容

1. 定义两个函数 `swap1` 和 `swap2`，目的是实现两个整数的次序交换，其中 `swap1` 的参数是传递值参数，`swap2` 的参数是引用参数，在程序中分别调用这两个函数，其输出结果如下所示：

初始值 : `x=1,y=2`

调用 `swap1` 后的结果: `x=1 y=2`

调用 `swap2` 后的结果: `x=2 y=1`

2. 编写一个函数，用来计算从键盘上输入的整数的阶乘。
3. 编写函数判断一个数是否是素数，在主程序中实现输入、输出。
4. 编写一个程序，设计一个函数可以计算并返回输入的 4 个数的平均值。

*5. 给定某个年、月、日的值，例如，2005 年 7 月 11 日，计算出这一天属于该年的第几天，要求写出计算闰年的函数和计算日期的函数。

三、实验要求

1. 结合上课内容，写出程序、调试程序，使用典型数据测试程序，并就实验结果加以分析。
2. 整理上机步骤，总结经验和体会。
3. 完成实验，并上交程序。

实验2 指针与引用

一、实验目的

1. 熟练掌握指针、地址、指针类型、**void** 指针、空指针等概念；
2. 熟练掌握指针变量的定义和初始化、指针的间接访问、指针的加减运算和指针表达式；
3. 会使用数组的指针和指向的指针变量；
4. 会使用字符串的指针和指向字符串的指针变量；
5. 学会使用指向函数的指针变量；
6. 了解指针与链表关系。

二、实验内容

1. 编写一个函数 f ，将传入此函数的直角坐标值转换为极坐标值，并返回主调函数中。求极坐标的公式是：

$$c = \sqrt{x^2 + y^2}$$

$$q = \arctan(y/x)$$

若要将两值返回主调函数，有多种方式可以完成，请试之：

- (1) 两值均以指针形参带回
- (2) 由指针形参带回一个值，函数值返回另一个值
- (3) 两值均以引用形参返回

(提示：程序中可以使用库函数 **sprt**、**pow** 及 **atan**，它们的头文件为 **math.h**)

2. 使用指针变量对一字符串按照字母，空格、数字和其他字符进行分类统计
(提示：读一行字符包括空格用函数 **cin.getline(ch, 81)**)。

3. 用指针变量设计一通用函数，该函数查找实型数组中最大和最小元素并输出相应元素和下标。

三、实验要求

1. 结合上课内容，写出程序、调试程序，使用典型数据测试程序，并就实验结果加以分析。
2. 整理上机步骤，总结经验和体会。
3. 完成实验，并上交程序。

实验3 类和对象

一、实验目的

1. 掌握类的概念、类的定义格式、类的成员属性和类的封装性；
2. 掌握对象的定义；
3. 理解类的成员的访问控制的含义，公有、私有和保护成员的区别；
4. 掌握构造函数和析构函数的含义与作用、定义方式和实现，能够根据要求正确定义和重载构造函数，能够根据给定的要求定义类并实现类的成员函数；
5. 掌握友元函数的含义，友元函数和成员函数的区别。

二、实验内容

1. 定义一个圆类，计算圆的面积和周长。

要求：分别用成员函数和友元函数来求圆的面积和周长。

2. 定义一个学生类，其中有3个数据成员有学号、姓名、年龄，以及若干成员函数。同时编写主函数使用这个类，实现对学生数据的赋值和输出。

要求：

- (1) 使用成员函数实现对数据的输入、输出；
- (2) 使用构造函数和析构函数实现对数据的输入、输出。

- *3. 定义日期类型 **Date**。要求有以下成员：

- (1) 可以设置日期；
- (2) 日期加一天操作；
- (3) 输入函数，输入格式为：XXXX XX XX，如 2010 4 13；
- (4) 输出函数，输出格式为：XXXX 年 XX 月 XX 日，如 2010 年 4 月 11

日。

- *4. 设计一个 **float** 类型的数组类 **CFloatArray**，要求 **CFloatArray** 可以进行如下操作：

- (1) 可以重置数组的大小 (**Resize**)
- (2) 可以通过下标返回数组元素，并对下标越界情况进行检查
- (3) 可以利用已知数组对象对整个数组赋值和初始化
- (4) 可以返回当前数组的大小 (**Size**)

最后用数据验证你所设计的类。

提示：

- (1) 利用已知数组对象对整个数组赋值，说明要求重载等号运算符
- (2) 利用已知数组对象对整个数组初始化，说明要求重载拷贝构造函数
- (3) 要求用指针存储数组对象中元素

三、实验要求

1. 结合上课内容，写出程序、调试程序，使用典型数据测试程序，并就实验结果加以分析。
2. 整理上机步骤，总结经验和体会。
3. 完成实验，并上交程序。

实验 4 继承与派生类

一、实验目的

1. 理解继承的含义，掌握派生类的定义方法和实现；
2. 理解公有继承下基类成员对派生类成员和派生类对象的可见性，能正确地访问继承层次中的各种类成员；
3. 理解保护成员在继承中的作用，能够在适当的时候选择使用保护成员以便派生类成员可以访问基类的部分非公开的成员；
4. 理解虚函数在类的继承层次中的作用，虚函数的引入对程序运行时的影响，能够对使用虚函数的简单程序写出程序结果。

二、实验内容

*1. 编写一个学生和教师数据输入和显示程序，学生数据有编号、姓名、班级和成绩，教师数据有编号、姓名、职称和部门。要求将编号、姓名输入和显示设计成一个类 **person**，并作为学生数据操作类 **student** 和教师类数据操作类 **teacher** 的基类。

*2. 编写一个程序计算出球、圆柱和圆锥的表面积和体积。

要求：

- (1) 定义一个基类圆，至少含有一个数据成员半径；
- (2) 定义基类的派生类球、圆柱、圆锥，都含有求表面积和体积的成员函数和输出函数。
- (3) 定义主函数，求球、圆柱、圆锥的和体积。

三、实验要求

1. 结合上课内容，写出程序、调试程序，使用典型数据测试程序，并就实验结果加以分析。
2. 整理上机步骤，总结经验和体会。
3. 完成实验，并上交程序。

实验5 多态性

一、实验目的

1. 掌握用成员函数重载运算符的方法;
2. 掌握用友元函数重载运算符的方法;
3. 理解并掌握利用虚函数实现动态多态性和编写通用程序的方法。

二、实验内容与步骤

1. 上机实验题一

定义一个复数类，通过重载运算符：*，/，直接实现二个复数之间的乘除运算。编写一个完整的程序，测试重载运算符的正确性。要求乘法“*”用友元函数实现重载，除法“/”用成员函数实现重载。

(1) 分析

两复数相乘的计算公式为： $(a+bi)*(c+di)=(ac-bd)+(ad+bc)i$

两复数相除的计算公式为： $(a+bi)/(c+di)=(ac+bd)/(c*c+d*d)+(bc-ad)/(c*c+d*d)i$

复数类及运算符重载函数可定义为：

```
class Complex
{
    float Real, Image;
public:
    Complex(float r=0, float i=0)
    {
        Real=r;
        Image=i;
    }
    void Show()
    {
        cout << "Real=" << Real << "\t" << "Image=" << Image << "\n";
    }
    friend Complex operator *(Complex &, Complex &);
    Complex operator/(Complex &); //重载运算符/
};

Complex operator*(Complex &c1, Complex &c2)
```

```

{
    Complex t;
    t.Real = c1.Real * c2.Real - c1.Image * c2.Image;
    t.Image = c1.Image * c2.Real + c1.Real * c2.Image;
    return t;
}

Complex Complex::operator/(Complex &c)
{
    Complex t;
    t.Real = (Real *c.Real+ Image * c.Image)/(c.Real*c.Real+ c.Image *
c.Image);
    t.Image = (Image *c.Real - Real * c.Image)/(c.Real*c.Real+ c.Image *
c.Image);
    return t;
}

```

(2) 上机要求

增加重载复数的加法和减法运算符的功能，实现两个复数的加法，一个复数与一个实数的加法；两个复数的减法，一个复数与一个实数的减法。用成员函数实现加法运算符的重载，用友元函数实现减法运算符的重载。

自己设计主函数，完成程序的调试工作。

*2. 上机实验题二

编写程序：声明一个哺乳动物类 **Mammal**，再由此派生出狗类 **Dog** 和猫类 **Cat**。

程序要求：

(1) **Mammal** 类中定义名为 **color** 和 **weight** 的数据成员，分别用来存储动物的颜色和体重；并在 **Mammal** 类中定义一个虚函数 **print**，用于输出 **color** 和 **weight** 值。

(2) 每个类中都定义一个名为 **count** 的静态数据成员，用来存储各类对象的总数；

(3) 每个类都有构造函数和析构函数，在构造函数中对 **color** 和 **weight** 进行初始化，以及执行 **count++** 语句，在析构函数中执行 **count--** 语句。在 **Dog** 和 **Cat** 类中输出各自的信息，输出信息要有所区别；

(4) 主函数中以一般方法定义各类若干个对象，用 **new** 创建几个对象指针，用 **delete** 删除对象指针。在不同的位置输出各类 **count** 的值。最后输出 "Main function ends."。

3. 上机实验题三

利用虚函数实现的多态性来求四种几何图形的面积之和。这四种几何图形是：三角形、矩形、正方形和圆。几何图形的类型可以通过构造函数或通过成员函数来设置。

(1) 分析

计算这四种几何图的面积公式分别是：

三角形的边长为 **W**，高为 **H** 时，则三角形的面积为 $W * H / 2$ ；矩形的边长为 **W**，宽为 **H** 时，则其面积为 $W * H$ ；正方形的边长为 **S**，则正方形的面积为 $S * S$ ；圆的半径为 **R**，其面积为 $3.1415926 * R * R$ 。

为设置几何图形的数据并求出几何图形的面积，需要定义一个包含两个虚函数的类：

```
class Shape
{
public:
    virtual float Area( void) =0;           //求面积
    virtual void Setdata(float ,float =0) =0; //设置图形数据
};
```

因面积的计算依赖于几何图形，故在类中只能定义一个纯虚函数 **Area**。同理，设置几何图形数据的函数 **Setdata** 也只能定义为虚函数。

把这个基类派生出其它几何图形类。如派生出的三角形类为：

```
class Triangle : public Shape
{
    float W, H;    //三角形边长为 W，高为 H
public:
    Triangle(float w=0, float h=0)
    {
        W = w;
        H = h;
    }
    float Area( void){ return W*H/2; }
```

```

        void Setdata(float w, float h=0){ W=w; H = h; }
};

```

在派生类中定义了基类中两个虚函数的实现。为了实现求面积和设置数据的多态性，必须定义一个类，该类中定义一个指向基类 **Shape** 的指针数组，其元素分别指向由基类 **Shape** 派生出的不同的几何图形类，并完成求出所有几何图形面积之和，以及设置参数的函数。

一个完整的参考程序如下：

```

#include <iostream.h>
#include <string.h>

class Shape
{
public:
    virtual float Area( void) = 0;           //虚函数
    virtual void Setdata(float, float=0) = 0; //虚函数
};

class Triangle : public Shape
{
    float W, H;      //三角形边长为 W，高为 H
public:
    Triangle(float w=0, float h=0) { W=w; H=h; }
    float Area( void) { return W*H/2; }      //定义虚函数
    void Setdata(float w, float h=0) { W=w; H=h; } //定义虚函数
};

class Rectangle:public Shape
{
    float W, H;      //矩形边长为 W，高为 H
public:
    Rectangle(float w=0, float h=0) { W=w; H=h; }
    float Area(void) { return W*H; } //定义虚函数
    void Setdata(float w, float h=0) { W=w; H=h; } //定义虚函数
};

```

```

class Square:public Shape
{
    float S;                                //正方形边长 S
public:
    Square(float a=0) { S=a; }
    float Area(void) { return S*S/2; }      //定义虚函数
    void Setdata(float w, float h=0) { S=w; } //定义虚函数
};

class Circle:public Shape
{
    float R;                                //圆的半径为 R
public:
    Circle(float r=0) { R=r; }
    float Area(void) { return 3.1415926*R *R; } //定义虚函数
    void Setdata(float w, float h=0) { R=w; } //定义虚函数
};

class Compute
{
    Shape **s;                              //指向基类的指针数组
public:
    Compute()
    {
        s= new Shape *[4];                //给几何图形设置参数
        s[0]= new Triangle(3,4);
        s[1]= new Rectangle(6,8);
        s[2]= new Square(6.5);
        s[3]= new Circle(5.5);
    }
    float SumArea(void ) ;
    ~Compute();
    void Setdata(int n, float a,float b=0)

```

```

    {
        s[n]->Setdata(a,b);
    }
};

Compute::~~Compute()                //释放动态分配的存储空间
{
    for(int i=0; i<4; i++)
        delete s[i];
    delete []s;
}

float Compute::SumArea(void)
{
    float sum=0;
    for( int i=0; i<4; i++ )
        sum += s[i]->Area();    //通过基类指针实现多态性
    return sum;
}

void main(void )
{
    Compute a;
    cout<<"四种几何图形的面积="<<a.SumArea()<<"\n";
    a.Setdata(2,10);            //设置正方形的边长
    cout<<"四种几何图形的面积="<<a.SumArea()<<"\n";
    a.Setdata(0,10,12);        //设置三角形的边长和高
    cout<<"四种几何图形的面积="<<a.SumArea()<<"\n";
    a.Setdata(1,2,5);          //设置正方形的长和宽
    cout<<"四种几何图形的面积="<<a.SumArea()<<"\n";
    a.Setdata(3,15.5);
    cout<<"四种几何图形的面积="<<a.SumArea()<<"\n";
}

```

程序中 A 行的 **Setdata** 函数属于函数重载，它不是虚函数。该函数中的 B 行

通过基类指针实现多态性。

三、实验要求

1. 结合上课内容，写出程序、调试程序，使用典型数据测试程序，并就实验结果加以分析。
2. 整理上机步骤，总结经验和体会。
3. 完成实验，并上交程序。

附录：（添加中.....）