



《程序设计基础 II》

课程实验指导书

《程序设计基础 II》课程组 编著

上海海洋大学海洋智能信息实验教学示范中心

实验一：面向对象程序设计基础---基础知识实验

实验目的：

- (1) 进一步学习 VC++6.0 开发环境及程序调试方法
- (2) 掌握 C++语言的基本语法

实验环境：

- (1) 硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）
- (2) 软件：操作系统为 Windows 7，工具软件为 Visual C++6.0

实验内容：

一、 编译下列程序，改正所出现的错误信息，并写出输出结果。

(1)第 1 小题

```
//test1.cpp
#include<iostream>
using namespace std;
int main()
{
cin>>a;           //语句 1
int b=3*a;        //语句 2
cout<<"b=<<b<<\n"; //语句 3
return 0;
}
```

(2)第 2 小题

```
//test2.cpp
int main()
{ cout<<"Hell0!\n";
cout<<"Welcome to C++ !"
}
```

(3)第 3 小题

```
//test3.cpp
#include<iostream>
using namespace std;
int main()
```

```

    {int x,y;
    x=5;
    y=6;
    int z=x*y;
    cout<<"x*y="<<z<<endl;
    return 0;
    )

```

二、请填空完成以下程序 test4.cpp，上机调试、运行并写出输出结果。

```

//test4.cpp
#include<iostream>
using namespace std;
int add(int a, int b);
int main()
{ int x,y,sum;
  cout<<"Enter two numbers:"<<'\n';
  //在下划线处填上语句，完成用 cin 读入 x 和 y
  _____
  _____

  sum=add(x,y);
  cout<<"The sum is:"<<sum<<'\n';
  return 0;
}
int add(int a,int b)
{
  //在下划线处填上语句,完成计算 a 与 b 的和并返回之
  _____

}

```

三、完成以下程序

(1) 数据类型与输入输出方法

分别定义字符、整型、浮点型、双精度型变量，并标准输入和输出这些变量。

(2) 控制结构

采用 if else 结构，按 bool 变量的值，输出“天晴”或“下雨”。

采用循环结构，输出九九乘法表。

采用 while 结构，输出 10 之前的正整数。

(3) 函数重载

定义三个重载函数，分别计算整型、浮点型和双精度型参数的平方根。

实验步骤：

- (1) 启动 Visual C++6.0 开发环境；
- (2) 创建一个项目；
- (3) 建立 C++源程序文件；
- (4) 编辑 C++源程序文件内容；
- (5) 建立并运行可执行程序；
- (6) 关闭工作空间

实验二：面向对象程序设计基础---类和对象（1）

实验目的：

- （1）理解掌握类的定义
- （2）理解掌握对象的创建
- （3）掌握对象的简单应用

实验环境：

（1）硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）

（2）软件：操作系统为 Windows7，工具软件为 Visual C++6.0

实验内容：

一、写出以下程序的运行结果

（1）第 1 小题

```
#include <iostream>
using namespace std;
class A{
public:
    void set(int i,int j)
    {
        x=i;
        y=j;
    }
    int get_y()
    {
        return y;
    }
private:
    int x,y;
};
class box{
public:
    void set(int l,int w, int s,int p)
```

```

    {
        length=l;
        width=w;
        label.set(s,p);
    }
    int get_area()
    {
        return length*width;
    }
private:
    int length,width;
    A label;
};
int main()
{
    box b;
    b.set(4,6,1,20);
    cout<<b.get_area()<<endl;
    return 0;
}

```

(2) 第 2 小题

```

#include <iostream>
using namespace std;
#include<iostream>
using namespace std;
class example
{
public:
    example(int n)
    {
        i=n;
        cout<<"constructing\n";
    }
    ~example()

```

```

    {
        cout<<"Destructing\n";
    }
    int get_i()
    {
        return i;
    }
private:
    int i;
};
int sqr_it(example o)
{
    return o.get_i()*o.get_i();
}
int main()
{
    example x(10);
    cout<<x.get_i()<<endl;
    cout<<sqr_it(x)<<endl;
    return 0;
}

```

(3) 第3小题

```

#include <iostream>
using namespace std;
class M{
    int x,y;
public:
    M()
    {
        x=y=0;
    }
    M(int i,int j)
    {
        x=i;
    }
}

```

```

        y=j;
    }
    void copy(M *m);
    void setxy(int i,int j)
    {
        x=i;
        y=j;
    }
    void print()
    {
        cout<<x<<" "<<y<<endl;
    }
};
void M::copy(M *m)
{
    x=m->x;
    y=m->y;
}
void fun(M m1,M *m2)
{
    m1.setxy(12,15);
    m2->setxy(22,25);
}
int main()
{
    M p(5,7),q;
    q.copy(&p);
    fun(p,&q);
    p.print();
    q.print();
    return 0;
}

```

二、编程题

- (1) 编写一个程序，任意从键盘输入两个字符，能将它们按由大到小的顺序输

出。要求程序中有一个交换两个字符的函数，其形参是变量的引用。

(2) 建立类 `cylinder`，`cylinder` 的构造函数被传递了两个 `double` 值，分别表示圆柱体的半径和高度。用类 `cylinder` 计算圆柱体的体积，并存储在一个 `double` 变量中。在类 `cylinder` 中包含一个成员函数 `vol`，用来显示每个 `cylinder` 对象的体积。

(3) 设计一个 4*4 魔方程式。魔方的各行元素值之和等于各列元素值之和，且等于两对角线元素之和。如下列魔方的各行、各列、两对角线上元素值之和等于 64。

31	3	5	25
9	21	19	15
17	13	11	23
7	27	29	1

[提示 1] 求 4*4 魔方的步骤如下：

- 1) 设置魔方的初始值 `first` (0 行 0 列元素的值)，和各相邻元素之间的差值 `step`。
- 2) 生成初始魔方。若 `first=1, step=2`，则得初始魔方为：

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

- 3) 求初始魔方的最大元素与最小元素之和 `sum`。`sum=max+min`。如以上初始魔方，`sum=32`。
- 4) `sum` 减去初始魔方中两对角线上各元素值，将差放在原来的位置，即得最终魔方。

[提示 2] 魔方类可设计如下：

```
class magic {
private:
    int m[4][4];
int first, step, sum;
public:
    void Getdata (int x1, int y1); // 输入first, step 的初值
    void InitMagic (); // 根据first, step 值生成初始魔方
    void FinalMagic(); // 生成最终魔方
    void DisplayMagic(); // 显示魔方
};
```

实验步骤：

- (1) 启动 Visual C++6.0 开发环境;
- (2) 创建一个项目;
- (3) 建立 C++源程序文件;
- (4) 编辑 C++源程序文件内容;
- (5) 建立并运行可执行程序;
- (6) 关闭工作空间

实验三：面向对象程序设计基础---类和对象（2）

实验目的：

- （1）学习掌握静态数据成员及静态成员函数的定义及使用
- （2）学习掌握友元函数的定义及使用

实验环境：

- （1）硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）
- （2）软件：操作系统为 Windows 7，工具软件为 Visual C++6.0

实验内容：

一、写出以下程序输出结果

```
#include <iostream>
using namespace std;
class M{
    int A;
    static int B;
public:
    M(int a)
    {
        A=a;
        B+=a;
        cout<<"Constructing"<<endl;
    }
    static void fl(M m);
    ~M()
    {
        cout<<"Destructing\n";
    }
};

void M::fl(M m)
{
    cout<<"A="<<m.A<<endl;
```

```

        cout<<"B="<<B<<endl;
    }
    int M::B=0;
    int main()
    {
        M P(5),Q(10);
        M::f1(P);
        M::f1(Q);
        return 0;
    }

```

二、编程题

(1) 编写一个程序，已有若干学生的数据，包括学号、姓名、成绩，要求输出这些学生的数据并计算出学生人数和平均成绩(要求将学生人数和总成绩用静态数据成员表示)。

(2) 设计一个用来表示直角坐标系的 `Location` 类，在主程序中创建类 `Location` 的两个对象 `A` 和 `B`，要求 `A` 的坐标点在第 3 象限，`B` 的坐标点在第 2 象限，分别采用成员函数和友元函数计算给定两个坐标点之间的距离，要求按如下格式输出结果：

`A(x1,y1), B(x2,y2)`

`Distance1=d1;`

`Distance2=d2;`

其中，`x1`、`y1`、`x2`、`y2` 为指定的坐标值，`d1` 和 `d2` 为两个坐标点之间的距离。

[提示]：

类 `Location` 的参考框架如下：

```

class Location{
public:
    Location(double,double);           //构造函数
    double Getx();                     //成员函数，取 x 坐标值
    double Gety();                     //成员函数，取 y 坐标值
    double dist (Location&, Location&); //成员函数，求给定两点之间的距离
    friend double dist (Location &,Location &);//友元函数，求给定两点之间的
    距离
private:
    double x,y;

```

};

实验步骤:

- (1) 启动 Visual C++6.0 开发环境;
- (2) 创建一个项目;
- (3) 建立 C++源程序文件;
- (4) 编辑 C++源程序文件内容;
- (5) 建立并运行可执行程序;
- (6) 关闭工作空间

实验四：面向对象程序设计基础---派生类与继承（1）

实验目的：

- （1）掌握派生类的声明方法和派生类构造函数的定义方法。
- （2）掌握不同继承方式下，基类成员在派生类中的访问属性。

实验环境：

- （1）硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）
- （2）软件：操作系统为 Windows 7，工具软件为 Visual C++6.0

实验内容：

一、写出以下程序的输出结果

```
#include <iostream>
using namespace std;
class B1 {
public:
    B1(int i)
    {
        b1=i;
        cout<<"Constructor B1."<<endl;
    }
    void print()
    {
        cout<<b1<<endl;
    }
private:
    int b1;
};
class B2 {
public:
    B2(int i)
    {
        b2=i;
```

```

        cout<<"Constructor B2."<<endl;
    }
    void Print()
    {
        cout<<b2<<endl;
    }
private:
    int b2;
};
class A:public B2,public B1 {
public:
    A(int i,int j,int l);
    void Print();
private:
    int a;
};
A::A(int i,int j,int l):B1(i),B2(j)
{
    a=l;
    cout<<"Constructor A."<<endl;
}
void A::Print()
{
    B1::print();
    B2::Print();
    cout<<a<<endl;
}
int main()
{
    A aa(3,2,1);
    aa.Print();
    return 0;
}

```

二、输入下列程序，回答题后问题

```

#include<iostream>
using namespace std;
class Base{
public:
void setx(int i)
{x=i;}
int getx()
{return x;}
public:
int x;
};
class Derived :public Base{
public:
void sety(int i)
{y=i;}
int gety()
{return y;}
void show()
{cout<<"Base ::x="<<x<<endl;} //语句 1
public:
int y;
};
int main()
{Derived bb; //语句 2
bb.setx(16); //语句 3
bb.sety(25); //语句 4
bb.show(); //语句 5
cout<<"Base::x="<<bb.x<<endl; //语句 6
cout<<"Derived::y="<<bb.y<<endl; //语句 7
cout<<"Base::x="<<bb.getx()<<endl; //语句 8
cout<<"Derived::y="<<bb.gety()<<endl; //语句 9
return 0;
}

```

1、写出程序的运行结果。

2、2、按照以下要求，对程序进行修改后再调试，指出调试中出错的原因

(1) 将基类 `Base` 中数据成员 `x` 的访问权限改为 `private` 时，会出现哪些错误？为什么？

(2) 将基类 `Base` 中数据成员 `x` 的访问权限改为 `protected` 时，会出现哪些错误？为什么？

(3) 在原程序的基础上，将派生类 `Derived` 的继承方式改为 `private` 时，会出现哪些错误？为什么？

(4) 在原程序的基础上，将派生类 `Derived` 的继承方式改为 `protected` 时，会出现哪些错误？为什么？

三、编程题

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` 和教师数据操作类 `teacher` 的基类。

实验步骤：

- (1) 启动 Visual C++6.0 开发环境；
- (2) 创建一个项目；
- (3) 建立 C++源程序文件；
- (4) 编辑 C++源程序文件内容；
- (5) 建立并运行可执行程序；
- (6) 关闭工作空间

实验五：面向对象程序设计基础---派生类与继承（2）

实验目的：

- （1）学习掌握基类与派生类对象之间的赋值兼容方法使用。
- （2）学习掌握多重继承派生类的声明和使用方法。

实验环境：

- （1）硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）
- （2）软件：操作系统为 Windows 7，工具软件为 Visual C++6.0

实验内容：

一、给出下面的基类：

```
class area_c1 {
protected:
    double height;
    double width;
public:
    area_c1(double r,double s)
    { height=r;width=s;}
    virtual double area()=0;
};
```

要求：

- （1）建立基类 `area_c1` 的两个派生类 `rectangle` 与 `isosceles`，让每一个派生类都包含一个函数 `area()`，分别用来返回矩形与三角形的面积。用构造函数对 `height` 与 `width` 进行初始化。
- （2）写出主程序，用来求 `height` 与 `width` 分别为 10.0 与 5.0 的矩形面积，以及求 `height` 与 `width` 分别为 4.0 与 6.0 的三角形面积。
- （3）要求通过使用基类指针访问虚函数的方法（即运行时的多态性）分别求矩形和三角形面积。

二、已有类 `Time` 和 `Date`，要求设计一个派生类 `Birthtime`，它继承类 `Time` 和 `Date`，并且增加一个数据成员 `Childname` 用于表示小孩的名字，同时设计主程序显示一个小孩的出生时间和名字。

```
class Time {
public:
    Time(int h,int m,int s)
```

```

        {hours=h;
          minutes=m;
          seconds=s;
        }
void display()
{cout<<"出生时间: " <<hours<<"时" <<minutes<<"分" <<seconds<<"秒" <<endl;
}
protected:
    int hours,minutes,seconds;
};
class Date{
public:
    Date(int m,int d,int y)
    {month=m;
      day=d;
      year=y;
    }
    void display()
    {cout<<"出生年月: " <<year<<"年" <<month<<"月" <<day<<"日" <<endl;
    }
protected:
    int month,day,year;
};

```

三、按要求阅读、编辑、编译、调试及运行以下程序

(1) 阅读、编辑、编译、调试及运行以下程序，并写出程序的运行结果

```

#include<iostream>
#include<string>
using namespace std;
class MyArray{                                //声明一个基类 MyArray
public:
    MyArray(int leng);                        //构造函数
    ~MyArray();                               //析构函数
    void Input();                             //输入数据的成员函数
    void Display(string);                     //输出数据的成员函数
protected:
    int *alist;                               //基类中存放一组整数
    int length;                               //整数的个数
};
MyArray::MyArray(int leng)
{
    if(leng<=0)
    {cout<<"error length";
      exit(1);
    }
}

```

```

    }
    alist=new int[leng];
    length=leng;
    if(alist==NULL)
    {
        cout<<"assign failure";
        exit(1);
    }
    cout<<"MyArray 类对象已创建。"<<endl;
}
MyArray::~MyArray()
{
    delete[] alist;
    cout<<"MyArray 类对象被撤销"<<endl;
}
void MyArray::Display(string str)
{
    int i;
    int *p=alist;
    cout<<str<<length<<"个整数: ";
    for(i=0;i<length;i++,p++)
        cout<<*p<<" ";
    cout<<endl;
}
void MyArray::Input() //从键盘输入若干整数
{
    cout<<"请从键盘输入"<<length<<"个整数";
    int i;
    int *p=alist;
    for(i=0;i<length;i++,p++)
        cin>>*p;
}
int main()
{
    MyArray a(5);
    a.Input(); //输入整数
    a.Display("显示已输入的"); //显示已输入的整数
    return 0;
}

```

(2) 声明一个类 `SortArray` 继承类 `MyArray`，在该类中定义一个函数，具有将输入的整数从小到大进行排序的功能。

【提示】

在第 (1) 步的基础上可增加下面的参考框架：

```
class SortArray:public MyArray{
```

```

public:
    void Sort();
    SortArray(int leng):MyArray(leng)
    {cout<<"SortArray 类对象已创建。"<<endl;}
    virtual ~SortArray();
};
SortArray::~SortArray()
{
cout<<"SortArray 类对象被撤销。"<<endl;
}
void SortArray::Sort()
{
//请自行编写 Sort 函数的代码，将输入的整数从小到大排序。
}
//并将主函数修改为：
int main()
{
SortArray s(5);
s.Input(); //输入整数
s.Display("显示排序以前的："); //显示排序以前的整数
s.Sort(); //进行排序
s.Display("显示排序以后的："); //显示排序以后的整数
return 0;
}

```

(3) 声明一个类 ReArray 继承类 MyArray，在该类中定义一个函数，具有将输入的整数进行倒置的功能。

【提示】

在第 (1) 步的基础上可增加下面的参考框架：

```

class ReArray:public MyArray{
public:
    void Reverse();
    ReArray(int leng);
    virtual ~ReArray();
};

```

请自行编写构造函数、析构函数和倒置函数 ReArray，以及修改主函数。

(4) 声明一个类 AverArray 继承类 MyArray，在该类中定义一个函数，具有求输入的整数平均值的功能。

【提示】

在第 (1) 步的基础上增加下面的参考框架：

```

class AverArray:public MyArray{
public:
    AverArray(int leng);
    ~AverArray();
    double Aver();
};

```

};

请自行编写构造函数、析构函数和求平均值函数 `Aver`（求解整数的平均值）以及修改主函数。

(5) 声明一个 `NewArray` 类，同时继承了类 `SortArray`, `ReArray` 和 `AverArray`，使得类 `NewArray` 的对象同时具有排序、倒置和求平均值的功能。在继承的过程中声明 `MyArray` 为虚基类，体会虚基类在解决二义性问题中的作用。

实验步骤：

- (1) 启动 Visual C++6.0 开发环境；
- (2) 创建一个项目；
- (3) 建立 C++源程序文件；
- (4) 编辑 C++源程序文件内容；
- (5) 建立并运行可执行程序；
- (6) 关闭工作空间

实验六：面向对象程序设计基础---多态性

实验目的：

- (3) 理解掌握静态多态---运算符重载函数的定义及应用
- (4) 理解掌握动态多态---虚函数的定义及应用

实验环境：

(1) 硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）

(2) 软件：操作系统为 Windows 7，工具软件为 Visual C++6.0

实验内容：

一、写出以下程序的输出结果

```
#include <iostream>
using namespace std;
class Words{
public:
    Words(char *s)
    {
        str=new char[strlen(s)+1];
        strcpy(str,s);
        len=strlen(s);
    }
    void disp();
    char operator [] (int n);
private:
    int len;
    char *str;
};
char Words::operator [] (int n)
{
    if (n<0||n>len-1)
    {
        cout<<"数组下标越界!\n";
```

```

        return ' ';
    }
    else
        return *(str+n);
}
void Words::disp()
{
    cout<<str<<endl;
}
int main()
{
    Words word("This is C++ book.");
    word.disp();
    cout<<"第 1 个字符: ";
    cout<<word[0]<<endl;
    cout<<"第 16 个字符: ";
    cout<<word[15]<<endl;
    cout<<"第 26 个字符: ";
    cout<<word[25]<<endl;
    return 0;
}

```

二、编程题

编写一个程序，定义抽象基类 **Container**，有它派生出 3 个派生类：**Sphere**（球体）、**Cylinder**（圆柱体）、**Cube**（正方体）。用虚函数分别计算几种图形的表面积和体积。

三、编程题

编写一个程序，其中设计一个时间类 **Time**，用来保存时、分、秒等私有数据成员，通过重载操作符“+”实现两个时间的相加。要求将小时范围限制在大于等于 0，分钟范围限制在 0~59 分，秒钟范围限制在 0~59 秒。

【提示】时间类 **Time** 的参考框架如下：

```

class Time{
public :
    Time(int h=0,int m=0,int s=0);    //构造函数
    Time operator+(Time&); //运算符重载函数，实现两个时间的相加

```



```
void disptime(string); //输出时间函数
private:
int hours;                //小时
int minutes;              //分钟
int seconds;              //秒钟
};
```

实验步骤:

- (1) 启动 Visual C++6.0 开发环境;
- (2) 创建一个项目;
- (3) 建立 C++源程序文件;
- (4) 编辑 C++源程序文件内容;
- (5) 建立并运行可执行程序;
- (6) 关闭工作空间

实验七：面向对象程序设计基础---模板与异常处理

实验目的：

- (1) 理解模板概念，练习掌握函数模板定义及其应用。
- (2) 练习掌握类模板定义及其应用。

实验环境：

- (1) 硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）
- (2) 软件：操作系统为 Windows 7，工具软件为 Visual C++6.0

实验内容：

一、分析并调试下列程序，写出运行结果并分析原因。

(1)

```
#include<iostream>
using namespace std;
template<typename T>           //函数模板
T max(T x,T y)
{return x>y?x:y;
}
int max(int a,int b)           //非模板函数
{return a>b ?a:b;
}
double max(double a,double b) //非模板函数
{return a>b ?a:b;
}
int main()
{cout<<"max(3,7) is "<<max(3,7)<<endl ;
return 0 ;
}
```

(2)

```
#include<iostream>
using namespace std;
int max(int a,int b)           //非模板函数
{return a>b ?a:b;
}
double max(double a,double b) //非模板函数
{return a>b ?a:b;
}
int main()
```

```
{cout<<“max(‘3’,’7’) is ”<<max(‘3’,’7’)<<endl ;  
return 0 ;  
}
```

二、编写一个求任意类型数组中最大元素和最小元素的程序，要求将求最大元素和最小元素的函数设计成函数模板。

三、编写一个程序，使用类模板对数组元素进行排序、倒置、查找和求和。

【提示】设计一个类模板：

```
template<class Type>  
class Array{  
...  
};
```

具有对数组元素进行排序、倒置、查找和求和功能，然后产生类型实参分别为 int 型和 double 型的两个模板类，分别对整型数组与双精度数组完成所要求的操作。

四、编写一个程序，求输入数的平方根。设置异常处理，对输入负数的情况给出提示。

实验步骤：

- (1) 启动 Visual C++6.0 开发环境；
- (2) 创建一个项目；
- (3) 建立 C++源程序文件；
- (4) 编辑 C++源程序文件内容；
- (5) 建立并运行可执行程序；
- (6) 关闭工作空间

实验八：面向对象程序设计基础---C++的流类库与输入输出

实验目的：

- (1) 掌握 C++格式化的输入输出方法。
- (2) 掌握重载运算符“<<“和”>>”的方法。
- (3) 掌握磁盘文件的输入输出方法。

实验环境：

- (1) 硬件：个人微机（配置不低于：CPU 为 P4，主频 1.6G，内存 256M，硬盘 40GB）
- (2) 软件：操作系统为 Windows 7，工具软件为 Visual C++6.0

实验内容：

一、下面给出打印九九乘法表的程序，但程序中存在错误。请上机调试，使得此程序运行后，能够输出如下所示的九九乘法表。

```
*  1  2  3  4  5  6  7  8  9
1  1
2  2  4
3  3  6  9
4  4  8 12 16
5  5 10 15 20 25
6  6 12 18 24 30 36
7  7 14 21 28 35 42 49
8  8 16 24 32 40 48 56 64
9  9 18 27 36 45 54 63 72 81
```

```
//test7_1_1.cpp
#include<iostream>
#include<iomanip>
using namespace std;
int main( )
{
    int i,j;
    cout<< “*”;
    for(i=1;i<=9;i++)
        cout<<i<< “ ”;
    cout<<endl;
    for(i=1;i<=9;i++)
    { cout<<i;
      for(j=1;j<=i;j++)
        cout<<i*j;
```

```

    }
    return 0;
}

```

二、下面的程序用于统计文件 xyz.txt 中的字符个数，请填空完成程序。

```

//test7_2_1.cpp
#include<iostream>
#include<fstream>
using namespace std;
int main( )
{
    char ch;
    int i=0;
    ifstream file;
    file.open("xyz.txt",ios::in);
    if( ① )
    {
        cout<< "xyz.txt cannot open"<<endl;
        abort();
    }
    while(!file.eof( ))
    {
        ②
        i++;
    }
    cout<< "文件字符个数: " << i << endl;
    ③
    return 0;
}

```

三、重载运算符“<<”和“>>”，使其能够输入一件商品的信息和输出这件商品的信息。商品的信息有编号、商品名和价格。假如商品类 **Merchandise** 的框架如下：

```

class Merchandise{
public:
    Merchandise( );
    ~Merchandise( );
    friend istream& operator>>(istream& in, Merchandise& s);// 输入一件商品
        的信息
    friend ostream& operator<<(ostream& out, Merchandise& s);//输出这件商
        品的信息
private:
    int no;
    char* name;
    double price;
}

```

```
};
```

要求实现该类，并编写以下的 **main** 函数对该类进行操作。

```
int main( )
```

```
{
```

```
    Merchandise mer;
```

```
    cin>>mer;
```

```
    cout<<mer;
```

```
    return 0;
```

```
}
```

四、编写一个程序，将两个文本文件连接成一个文件，然后将此文件中所有小写字母转换成大写字母，并打印出来。

实验步骤：

- (1) 启动 Visual C++6.0 开发环境；
- (2) 创建一个项目；
- (3) 建立 C++源程序文件；
- (4) 编辑 C++源程序文件内容；
- (5) 建立并运行可执行程序；
- (6) 关闭工作空间