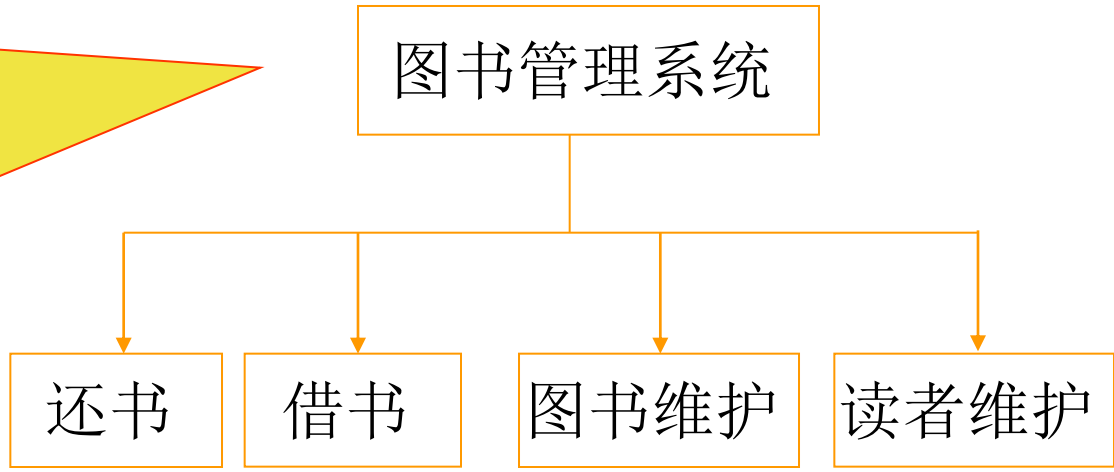
第7讲　输入/输出流

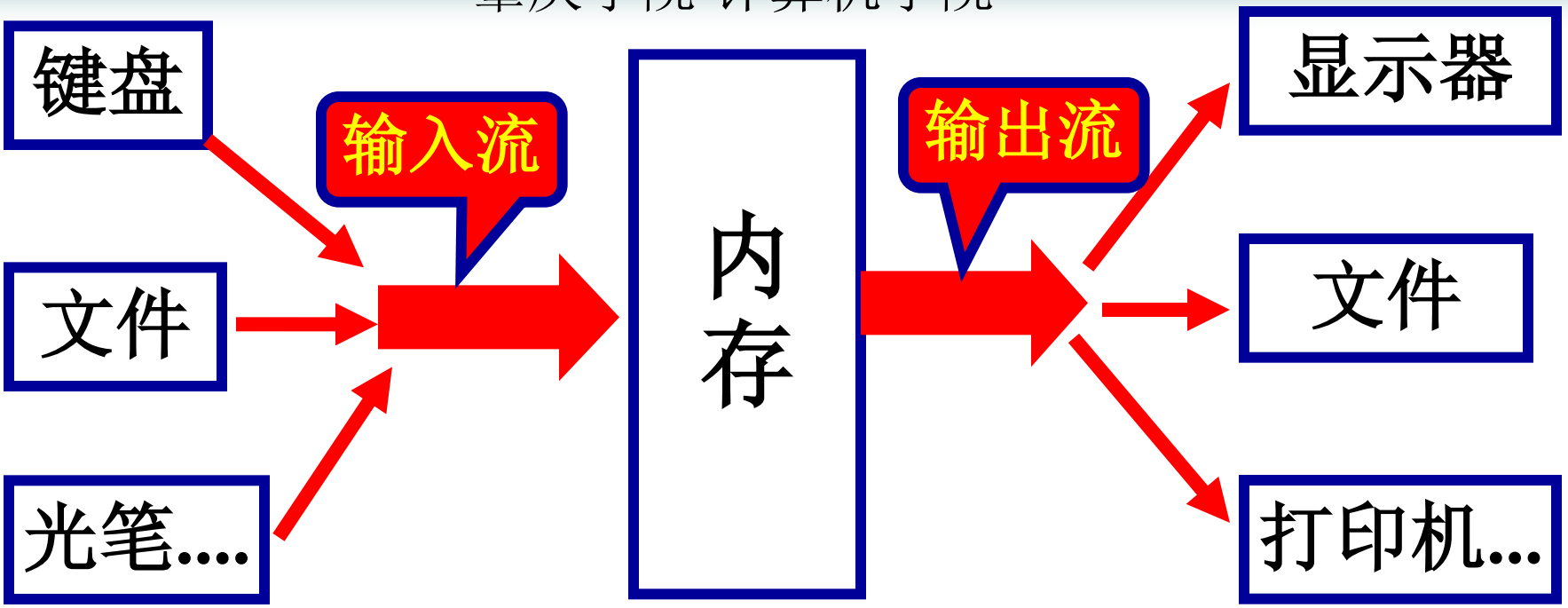
[课程设计例：问题描述]

- 设计一个图书管理系统,该图书管理系统分为借书,还书,图书维护和读者维护4个部分,如下图所示.

不使用数据库，如何将对象的数据从键盘输入显示在屏幕上，并保存在文件中？或者从文件中读取对象的数据显示在屏幕上？



图书管理系统示意图



编译系统已经以运算符或函数的形式做好了对标准外设（键盘、屏幕、打印机、文件）的接口，使用时只需按照要求的格式调用即可。

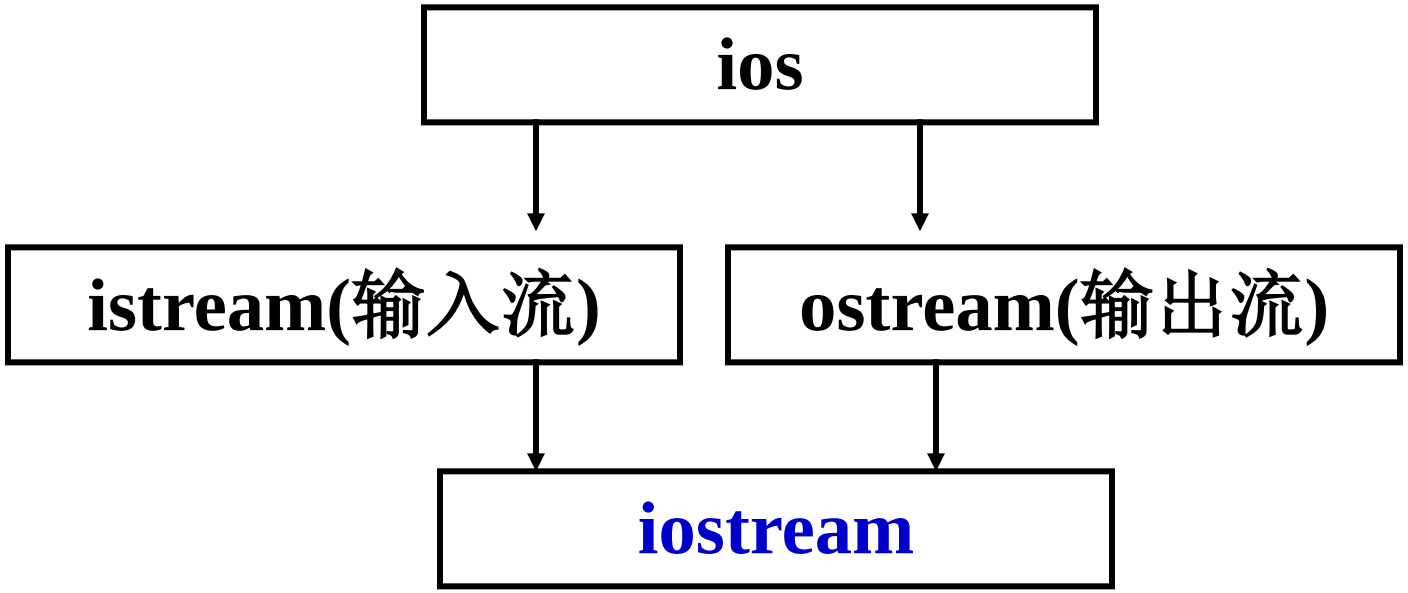
cin>>x;　cout<<x;

cin.get(ch);

从输入流中接受一个字符，并赋给字符变量ch

输入输出流（I/O Stream）

C++语言的I/O系统向用户提供一个统一的接口，使得程序的设计尽量与所访问的具体设备无关，在用户与设备之间提供了一个抽象的界面:输入输出流。 继承层次如下：



在“iostream.h”中说明

用标准流进行输入/输出时，系统自动地完成数据类型的转换：

对于输入流，要将输入的字符序列形式的数据变换成计算机内部的二进制数后，再赋给变量，变换后的格式由变量的类型确定。

对于输出流，将要输出的数据变换成字符串形式后，送到输出流（或文件）中。

如何输入输出对象的成员？

```
class A
{ float x, y;
public:
   A(float a=0, float b=0){  x=a;  y=b;  }
   void Set(float a, float b){ x=a; y=b; }
   void  Show(void){  cout<<x<<'\t'<<y<<endl;   }
};
void main(void)
{  A   a(2,3);
    a.Set(20 , 30);
    a.Show( );
}
```

输入对象数据

输出对象数据

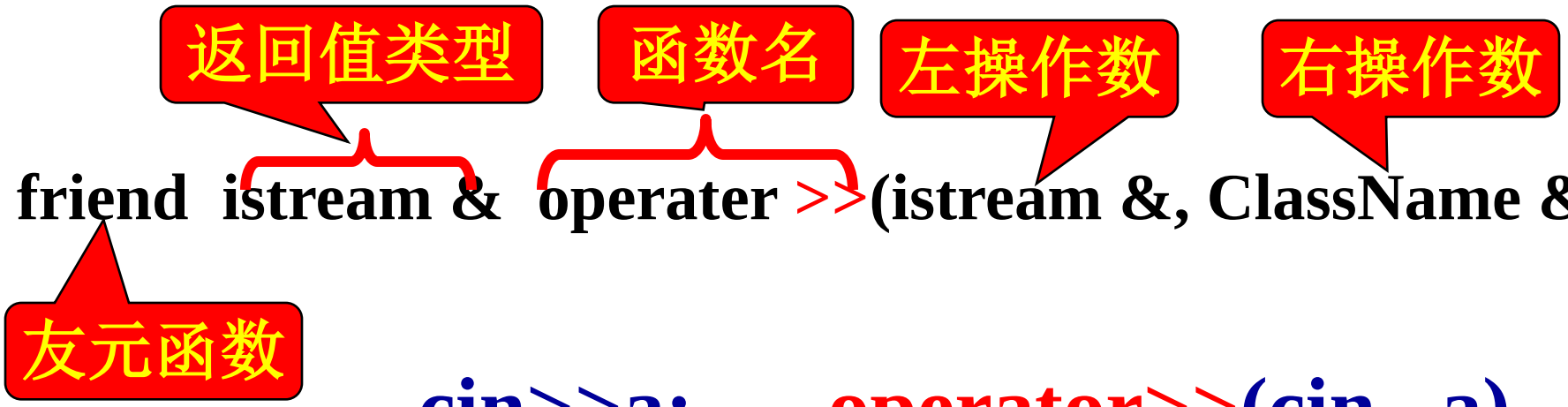
cin>>a;

cout<<a;

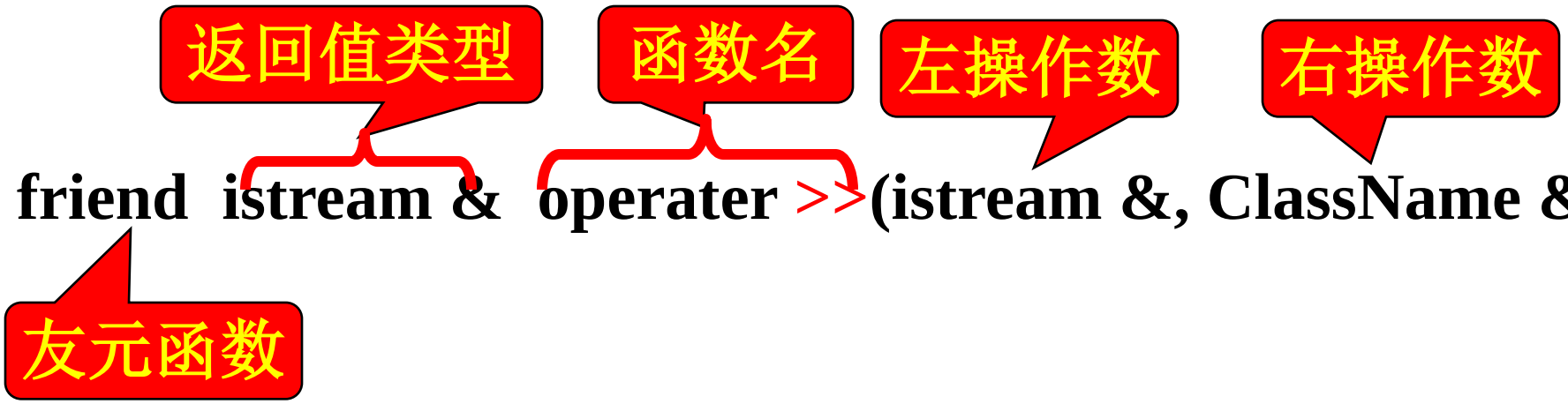
对象不能直接输入输出，需重载运算符

在C++中允许用户重载运算符“<<”和“>>”，实现对象的输入和输出。重载这二个运算符时，将重载这二个运算符的函数说明为类的友元函数。

重载提取运算符的一般声明格式为：



cin>>a;　　operator>>(cin, a)



cin>>a;　　operator>>(cin, a)

返回值类型：类istream的引用，cin中可以连续使用运算符“>>”。

cin>>a>>b;

```
class A
{    float x, y;
  public:
     .....
    friend  istream &  operater >>(istream &, A &);
};
     .....
    A  a;
    cin>>a;
     ....
```

在类中原型说明

在类外定义函数

```
istream &  operater >>(istream &is, A &a)
{    cout<<“ Input a:”<<endl;
       is>>a.x>>a.y;
       return  is;
}
```

重新定义输入流

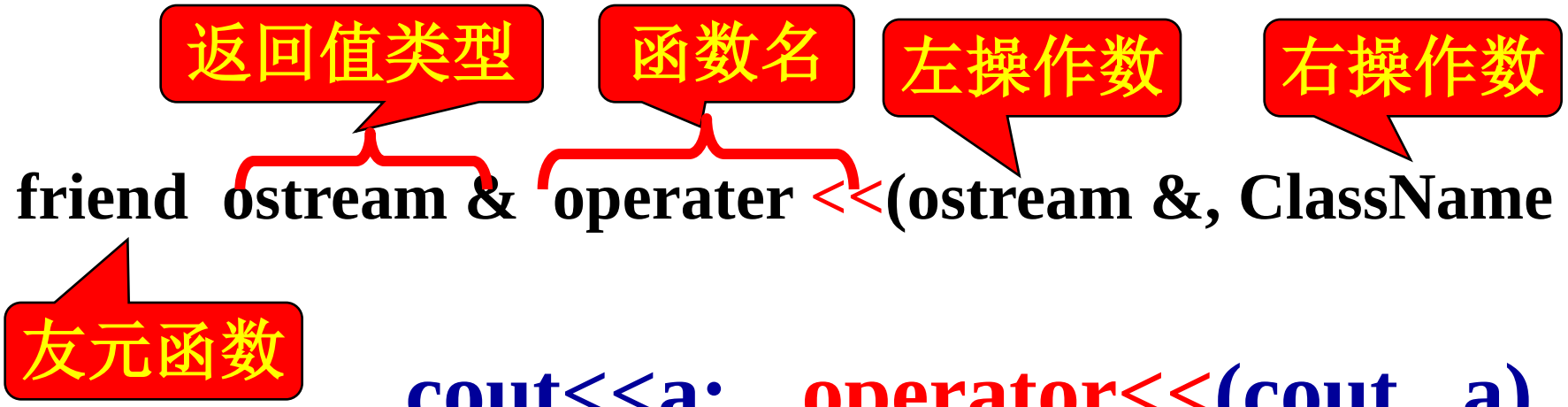
返回输入流

```
class incount{
        int c1,c2;
public:incount(int a=0,int b=0)      {        c1=a;   c2=b;   }
  void show(void){cout<<"c1="<<c1<<'\t'<<"c2="<<c2<<endl;}
   friend istream & operator>>(istream &,incount &);
};
istream & operator>>(istream &is,  incount &cc)
{       is>>cc.c1>>cc.c2;      return is;       }
void main(void)
{       incount x1,x2;
        x1.show ();
        x2.show ();
        cin>>x1;
        cin>>x2;
        x1.show ();
        x2.show ();
}
```

重载输入函数原型说明

重载输入函数定义

重载输出（插入）运算符的一般格式为：



cout<<a;　operator<<(cout, a)

与输入（提取）运算符比较：

friend istream & operater >>(istream &, ClassName &);

将输入流改为输出流。

```
class A
{    float x, y;
  public:
     .....
   friend  ostream &  operater << (ostream &, A &);
};
     .....
    A  a(2,3);
    cout<<a;
     ....
```

在类中原型说明

在类外定义函数

```
ostream &  operater <<(ostream &os, A &a)
{    cout<<“ The object is:”<<endl;
      os<<a.x<<‘\t’<<a.y<<endl;
      return  os;
}
```

重新定义输出流

返回输出流

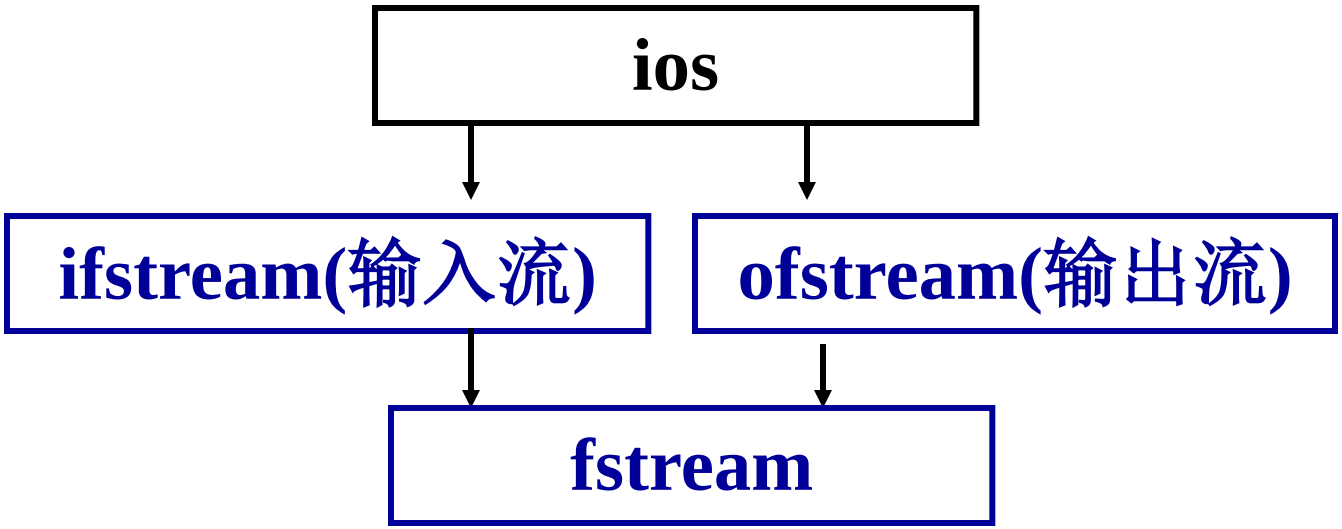
```
class incount{
        int c1,c2;
public:  incount(int a=0,int b=0)  {        c1=a;    c2=b;    }
        void show(void)  {cout<<"c1="<<c1<<'\t'<<"c2="<<c2<<endl;       }
        friend istream & operator>>(istream &,incount &);
        friend ostream & operator<<(ostream &,incount &);
};
istream & operator>>(istream &is,incount &cc)
{       is>>cc.c1>>cc.c2;          return is;}
ostream &operator<<(ostream &os,incount &cc)   //重载cout<<
{os<<"c1="<<cc.c1<<'\t'<<"c2="<<cc.c2<<endl;   return os;}
void main(void)
{
        incount x1,x2;
        cout<<x1<<x2;//调用输出函数
        cin>>x1;          //调用输入函数
        cin>>x2;
        cout<<x1<<x2;
}
```

重载输出函数原型说明

重载输出函数定义

文件流：

C++在头文件fstream.h中定义了C++的文件流类，当程序中使用文件时，要包含头文件fstream.h 。

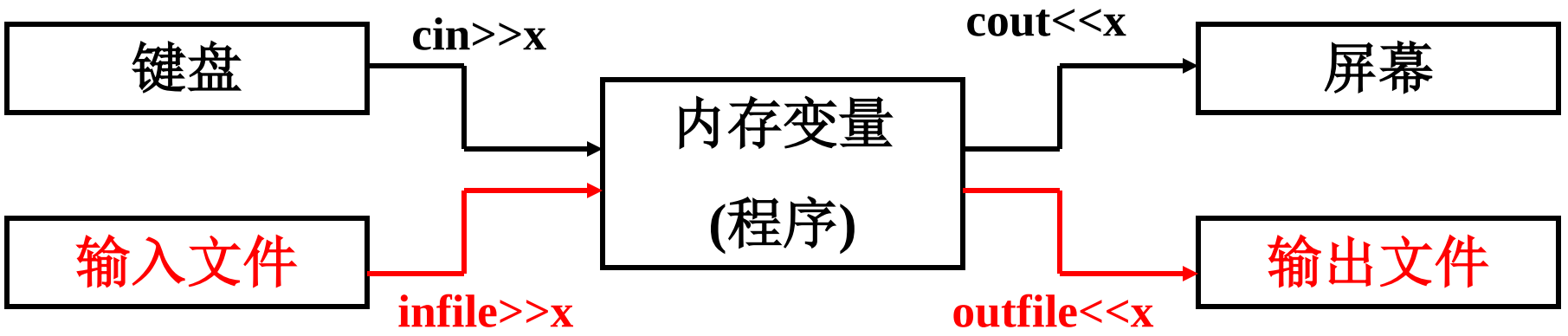


在“fstream.h”中说明

当使用文件时，程序头：#include<fstream.h>

其中定义了各种文件操作运算符及函数。

程序对文本文件的操作与对键盘、显示器的操作比较：



在涉及文本文件的操作时，将输入文件看成键盘，将输出文件看成显示器，格式不变。只需在程序中增加打开与关闭文件的语句。

文件的操作

文件
- 文本文件：
  - 以ASCII表示的文件： 记事本文件，*.cpp等
- 二进制文件：
  - 用二进制形式表示的文件：可执行程序*.EXE等

字符5对应35H,即53　　字符6对应36H，即54

56：ASCII表示为 00110101 00110110，占两字节

38H，对应十进制的56

56：二进制表示为 111000，占六个二进制位

对应以上不同的文件，操作的函数、格式不同

文本文件的操作步骤（分4步）：

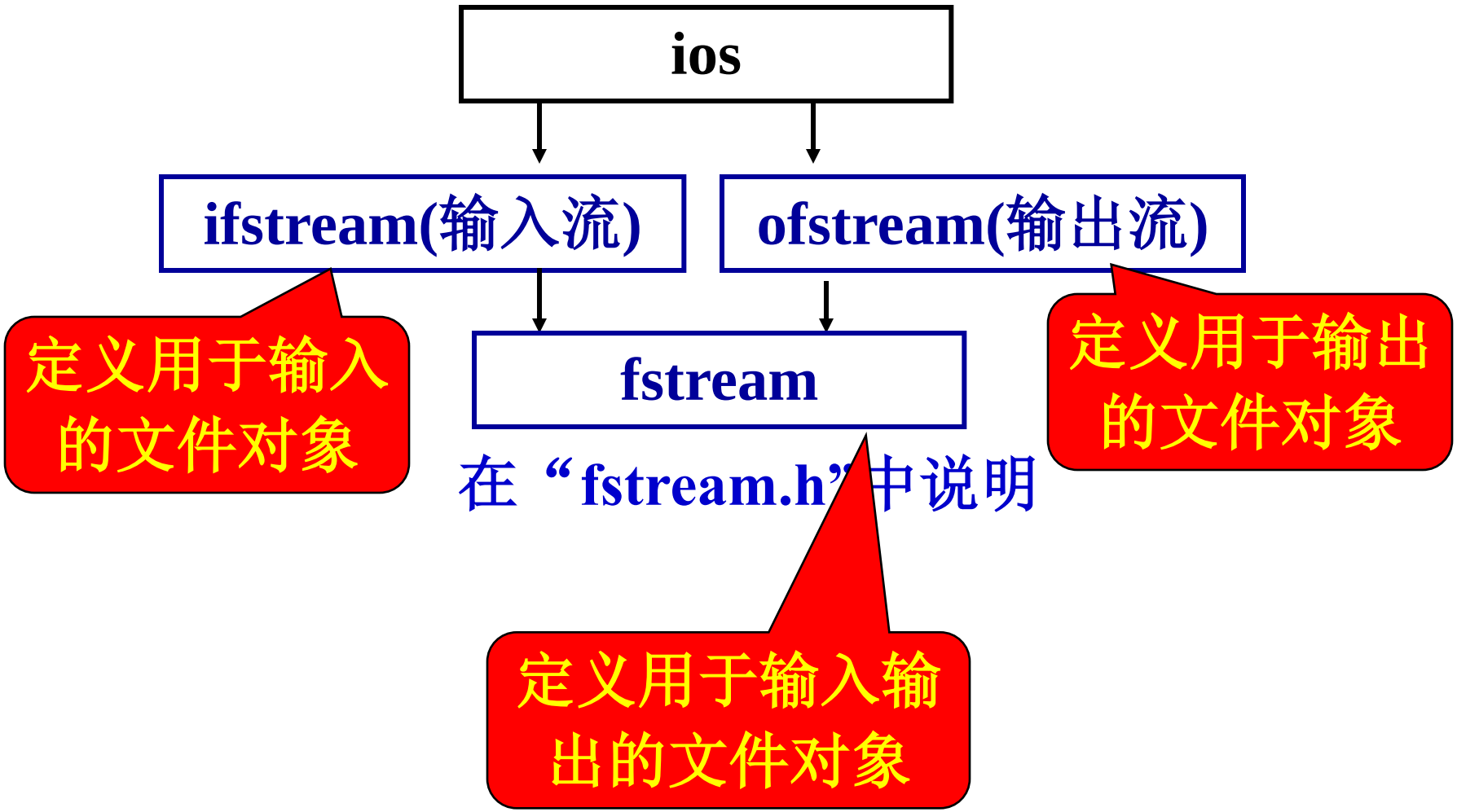
第1步，在程序内定义一个文件类的对象，由该对象与文件发生联系，程序内所有与文件的操作都是对该对象的操作。

```
fstream   infile , outfile;
ifstream    infile;
ofstream   outfile;
```

两个对象，可以联系两个输入输出文件

对象只能联系输入文件

对象只能联系输出文件


ios
ifstream(输入流)
ofstream(输出流)
定义用于输入的文件对象
fstream
定义用于输出的文件对象
在“fstream.h”中说明
定义用于输入输出的文件对象

第2步，用对象打开文件：

```
ifstream  infile;  //定义输入文件类对象
infile.open(“myfile1.txt”);//利用函数打开某一文件
```

打开文件的作用是，使文件流对象与要使用的文件名之间建立联系。

```
ofstream  outfile;  //定义输出文件类对象
outfile.open(“myfile1.txt”);//打开某一文件供输出
```

infile.open(“myfile1.txt”);

打开文件“myfile1.txt”用于输入，并将这个文件与输入文件类对象infile建立联系，今后，在程序中，用到这个文件“myfile1.txt”的地方就用infile来代替。

outfile.open(“myfile2.txt”);

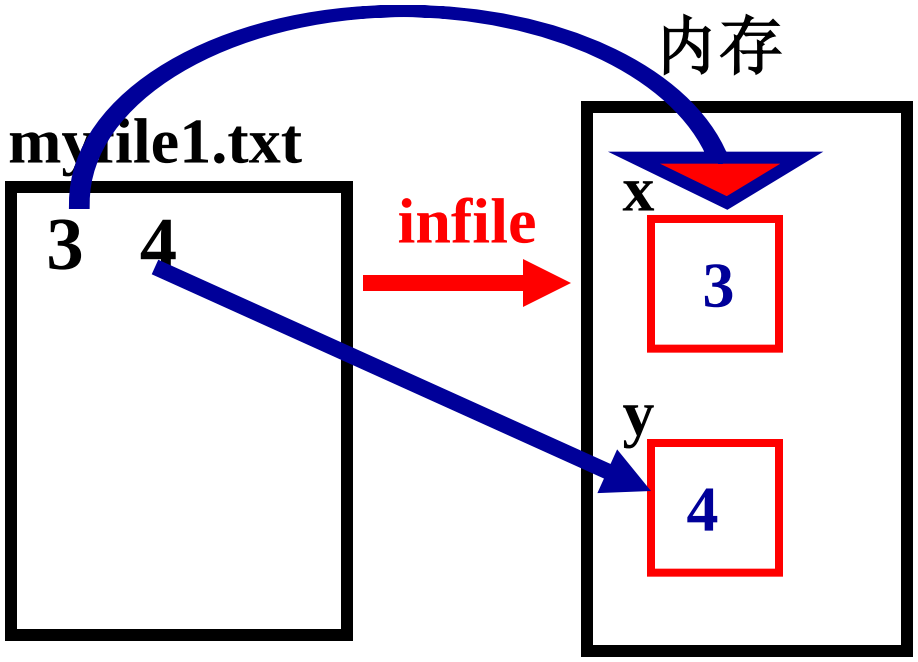
打开文件“myfile2.txt”用于输出，并将这个文件与输出文件类对象outfile建立联系，今后，在程序中，用到这个文件“myfile2.txt”的地方就用outfile来代替。

第3步，从文件中输入输出数据：

将文件类对象看成键盘和显示器即可。

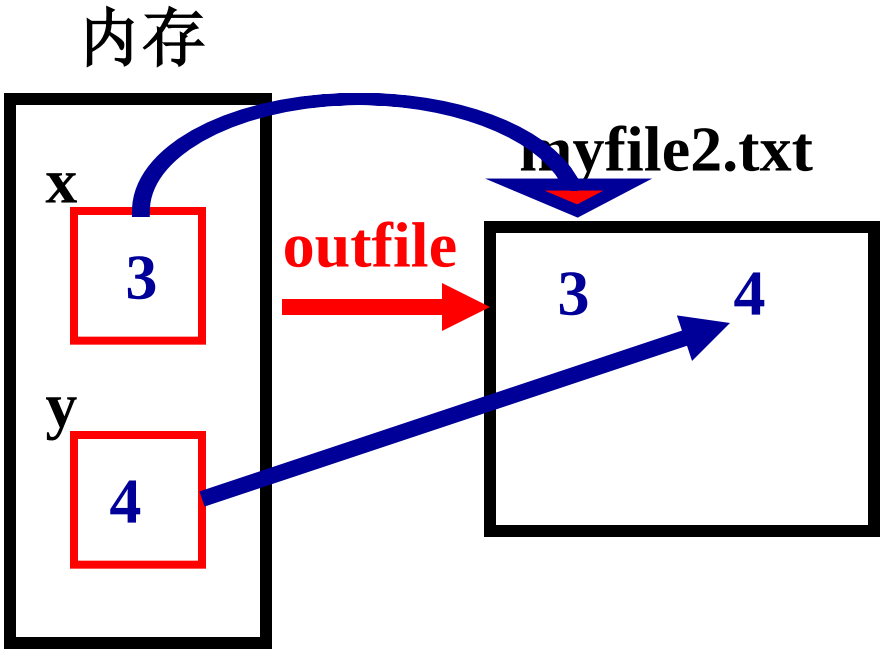
```
ifstream  infile;  //定义输入文件类对象
infile.open("myfile1.txt");//利用函数打开某一文件
float  x , y;
infile>>x>>y;
```

用 infile 代 替 myfile1.txt 进行操作。



```
ofstream  outfile;  //定义输出文件类对象
outfile.open("myfile2.txt");//利用函数打开某一文件
  float x＝3 , y＝4;
outfile<<x<<'\t'<<y<<endl;
```

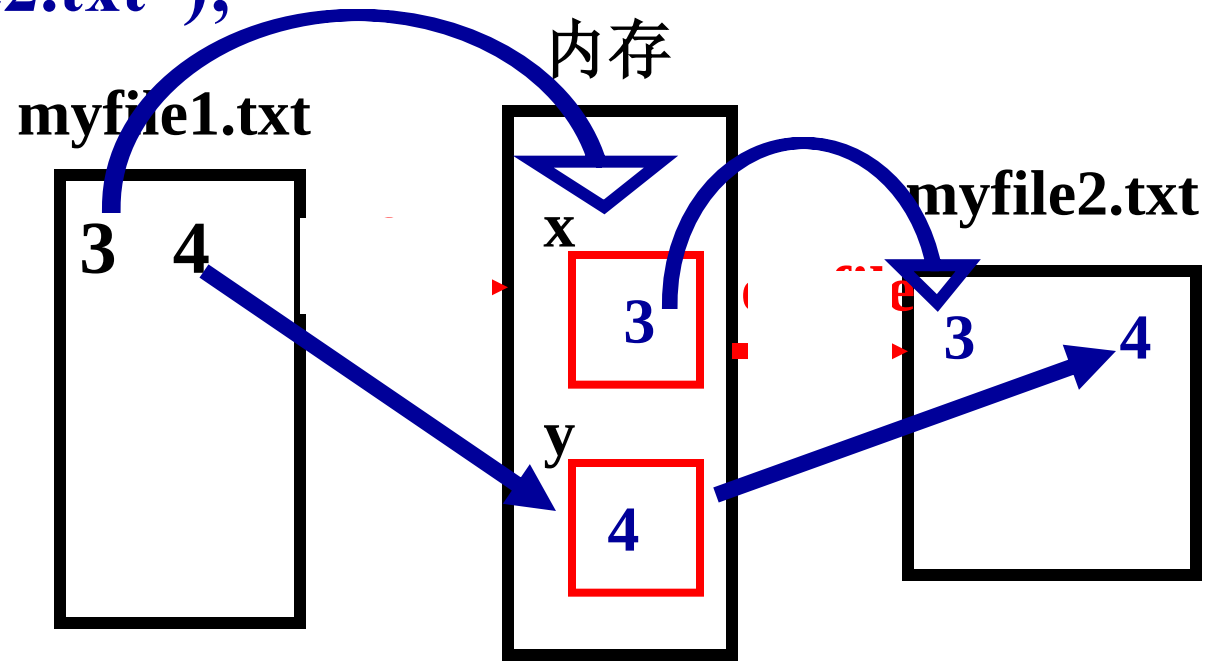
用outfile代替myfile2.txt进行操作。



第4步，用完文件后,使用成员函数关闭文件：

```
ifstream  infile;
ofstream outfile;
infile.open(“myfile1.txt”);
outfile.open(“myfile2.txt”);
float x,y;
infile>>x>>y;
outfile<<x<<‘\t’
<<y<<endl;
infile.close();
outfile.close();
```

```
infile.close();
outfile.close();
```



实例：

注：输入文件“file1.txt”必须存在，且在其中有十个整数

```
#include <fstream.h>
void main(void)
{  int a[10];
   ifstream   infile;      //定义输入文件类
   ofstream outfile;     //定义输出文件类
   infile.open(“file1.txt”);      //打开一个输入文件“file1.txt”
   outfile.open(“file2.txt”);    //打开一个输出文件“file2.out”
   for(int i=0;i<10;i++)
        infile>>a[i];//将“file1.txt”中的十个整型数输入到a[i]中
    for(i=0;i<10;i++)
         outfile<<a[i]<<‘\t’;//将a[i]中的十个数输出到文件“file2.txt”中
   infile.close();//关闭输入文件
   outfile.close();//关闭输出文件
}
```

当用类fstream定义文件对象时，该对象既能定义输入文件对象，又能定义输出文件对象，所以打开文件时，必须在成员函数open()中的参数中给出打开方式（读或写）。

```
fstream    pfile1,pfile2;//定义了两个文件类的对象
pfile1.open("file1.txt",  ios::in);//pfile1联系到 "file1.txt",用于输入
pfile2.open("file2.txt", ios::out);//pfile2联系到 "file2.txt",用于输出
char  ch;
pfile1>>ch;     //输入
pfile2<<ch;     //输出
pfile1.close();
pfile2.close();
```

在打开文件后，一般都要判断打开是否成功。若打开成功，则文件流对象值为非零；若打开不成功，其值为0。

```
ifstream    pfile1,pfile2;//定义了两个文件类的对象
pfile1.open("file1.txt",  ios::in);
pfile2.open("file2.txt", ios::out);
if (!pfile1)
 {cout <<"不能打开输入文件：file1.txt"<<'\n';  exit(1);}
if (!pfile2)
 {cout <<"不能打开输出文件：file2.txt"<<'\n';  exit(1);}
```

若为0，打开文件操作失败

打开输入文件时，若文件不存在，不会建立新文件。

打开输出文件时，若文件不存在，则建立新文件；若文件存在，则删除原文件的内容，使其成为一个空文件。

涉及到字符串的文件读写

用构造函数打开文件

```
char ch, str[300];
ifstream infile("myfile1.txt");
ofstream outfile("myfiel2.txt");
```

从键盘输入一个字符：cin.get(ch);

从文件输入一个字符：infile.get(ch);

向显示器输出一个字符：cout.put(ch);

向文件输出一个字符：outfile.put(ch);

从键盘输入一行字符：cin.getline(str,300);

从文件输入一行字符：infile.getline(str,300);

从文件中输入一字符或一行字符，当输入至文件尾时，函数返回值为0，可以据此来判断循环结束。

例： 实现两文件的拷贝的程序

```
void main(void)
{       char filename1[256],filename2[256];
        cout<<"Input source file name:  ";
        cin>>filename1;
        cout<<"Input destination:  ";
        cin>>filename2;
        ifstream infile(filename1);
        ofstream outfile(filename2);
        char ch;
        while(infile.get(ch))
                outfile.put(ch);
        infile.close();
        outfile.close();
}
```

输入文件(源文件)名

输出文件(目的文件)名

用构造函数打开文件

从源文件中读取一个字符，至文件尾停止循环

将该字符输出至目的文件

关闭文件

```
void main(void)
{       char filename1[256],filename2[256];
        char buf[300];
        cout<<"Input source file name:  ";
        cin>>filename1;
        cout<<"Input destination:  ";
        cin>>filename2;
        fstream infile,outfile;
        infile.open(filename1,ios::in);
        outfile.open(filename2,ios::out);
        while(infile.getline(buf,300))
                outfile<<buf<<endl;
        outfile.close();
        infile.close();
}
```

输入文件(源文件)名

输出文件(目的文件)名

用函数打开文件

从源文件中读取一行字符，至文件尾停止循环

将该行字符输出至目的文件

关闭文件

二进制文件的读写操作：分2步。

第1步，打开二进制文件。

```
fstream   infile,outfile;
infile.open("inf1.dat",  ios::in| ios::binary);
outfile.open("outf1.dat",  ios::out| ios::binary);
```

二进制文件

文件名

输入方式打开

文件名

输出方式打开

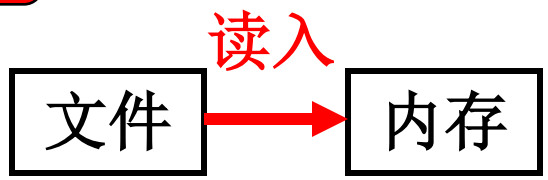
二进制文件

第2步：读写二进制文件。

输入函数：

一次读入的字节数

infile.read(char *, int)



输入文件对象名

数据进入的内存地址

地址要强制转换成字符型指针

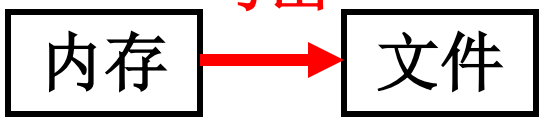
```
int i;
infile.read((char *)&i, sizeof(int));//从文件中输入一个整型数到i
int a[10];
infile.read((char *)a, 10*sizeof(int));//从文件中输入十个整型数到a
```

输出函数：

一次输出的字节数

outfile.write(char *, int)

写出



输出文件对象名

要输出的数据在内存中的地址

```
int  i=4;
outfile.write((char *)&i, sizeof(int));//向文件输出一个整型数i
```

地址要强制转换成字符型指针

```
int  a[10]={0,1,2,3,4,5,6,7,8,9};
outfile.write((char *)a,  10*sizeof(int));//向文件输出一个整型数组a
```

判断二进制文件是否读到文件尾？

infile.eof( )

当到达文件结束位置时，该函数返回一个非零值；否则返回零。

```
fstream   infile;
infile.open("data1.dat",ios::in|ios::binary);
if(!infile)
{   cout<<"Open Error!\n";  exit(1);  }
char  str[300];
while(!infile.eof())
    infile.read(str, 300);
```

判断打开是否出错

判断是否读到文件尾

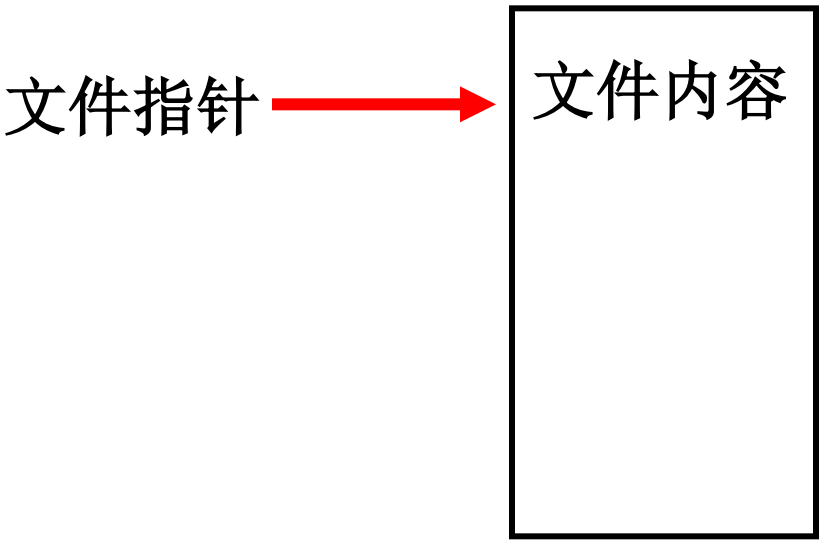
实例：

```
void  main(void )
{char  filename1[256],filename2[256];
char  buff[4096];
cout <<"输入源文件名:";  cin  >>filename1;
cout <<"输入目的文件名:";  cin  >>filename2;
fstream  infile,outfile;
infile.open(filename1,ios::in | ios::binary);
outfile.open(filename2,ios::out | ios::binary);
int n;
while (!infile.eof()){              //文件不结束，继续循环
        infile.read(buff,4096);          //一次最多读4096个字节
        n=infile.gcount();               //取实际读的字节数
        outfile.write(buff,n);           //按实际读的字节数写入文件
        }
infile.close();
outfile.close();
}
```

文件指针：

打开文件时，文件指针位于文件头，并随着读写字节数的多少顺序移动。

可以利用成员函数随机移动文件指针。



随机读取二进制文件

infile.seekg(int);//将文件指针移动到由参数指定的字节处

infile.seekg(100);//将文件指针移动到距离文件头100个字节处

infile.seekg(int, ios::_dir);

移动的字节数

相对位置

_dir: **beg:** 文件头
cur: 当前位置
end: 文件尾

infile.seekg(100, ios::beg);//移动到距文件头100个字节

infile.seekg(-100, ios::cur);//移动到距当前位置前100个字节

infile.seekg(-500, ios::end);//移动到距文件尾前500个字节

```
void  main(void )
{  ofstream  outfile(“data1.dat”,ios::out| ios::binary);
    int  i;
    for(i=5;i< 1000;i+=2 )
          outfile.write((char*)&i,sizeof(int)); //将奇数写入文件
     outfile.close();//关闭文件
     ifstream  f1(“data.dat”,ios::in| ios::binary);
        int x;
   f1.seekg(20*sizeof(int));
        //将文件指针移到距文件头第20个整数的位置
   for(i=0;i<10;i++)
   {f1.read((char *)&x,sizeof(int)); //依次读出第20~29个奇数到x
            cout<< x<< ‘\t’;
    }
    f1.close();
}
```

以读的方式打开原文件

设在缺省目录下有文件file1.txt，文件中内容为：

1　　2　　3　　4　　5　　6

执行下述程序后，程序的输出是___21___。

```
void main(void)
{
        fstream f1;
        int tmp, sum=0;
        f1.open("file1.txt",ios::in);
        while(f1>>tmp)
                sum+=tmp;
        f1.close( );
        cout<<sum<<endl;
}
```

如果令A,B,C,D,……，X,Y,Z这26个英文字母，分别等于百分之1,2,……,24,25,26个数值，那么我们就能得出：

HARD WORK 8+1+18+4+23+15+18+11=98％

(努力工作)

KNOWLEDGE （知识） 96%

LOVE（爱情） 54% **LUCK（运气） 47%**

计算一下MONEY 72% **ATTITUDE 100%**

作业：先看
一个有趣的结论

正是我们对待工作、生活的态度能够使我们的生活达到100%的圆满！

作业

- **上述问题：从file1.txt中读入任意单词，输出单词及对应的百分比到file2.txt**