**Const**

1. 定义圆类，求圆面积及圆周长。将pi定义为常数据成员。将area函数和zhouchang函数以及print函数定义为常成员函数。
2. 在上题基础上，增加一个常对象，并调用函数。

**Static**

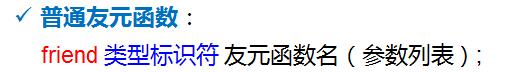
**静态数据成员**

1. 定义CStudent类，在类中添加静态数据成员，保存学生的总数。每创建一个学生对象，学生数加1；学生对象撤销时，学生数减1。

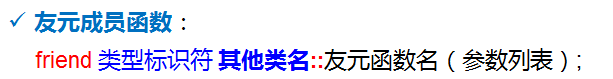
**静态成员函数**

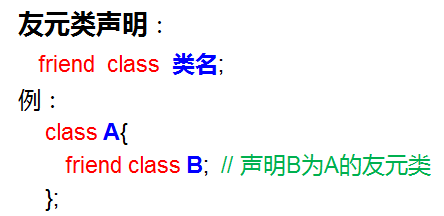
1. 在上题基础上，将静态数据成员变为私有成员，用静态成员函数访问静态数据成员。

**Friend**



**5、定义点类，类中声明一个友元函数pdistance，实现求两点间的距离。在主函数中完成测试工作。**

**6、编写日期类和时间类。日期类中定义一个成员函数display；在时间类中声明display为其友元成员函数，在主程序中予以测试。**



**7、 编写日期类和时间类。其中，日期类是时间类的友元类。编写主函数予以测试。**

**参考答案**

**（3）静态数据成员**

**#include <iostream.h>**

**#include <string.h>**

**class CStudent**

**{public:**

**CStudent(int, char \*);**

**~CStudent();**

**void Print();**

**static int noofStds; //静态数据成员，公共区定义。**

**private:**

**int id;**

**char name[20];**

**};**

**//静态成员的共享性。**

**CStudent::CStudent(int i, char \*na)**

**{**

**id = i;**

**strcpy(name, na);**

**noofStds ++;//创建一个学生，学生数加1**

**}**

**CStudent::~CStudent()**

**{**

**noofStds --;//学生对象撤销时，学生数减1**

**}**

**void CStudent::Print()**

**{**

**cout<<"姓名："<<name<< " ,"<<"学号："<<id <<endl;**

**}**

**int CStudent::noofStds = 0; //静态数据成员初始化，类外de 在类的事项部分。**

**void func()**

**{**

**CStudent s3(10003, "王五" );**

**s3.Print();**

**cout << "学生数为："<< CStudent::noofStds << endl;**

**}**

**void main()**

**{ //静态成员可以直接引用：类名或对象引用。**

**CStudent s1(10001, "张三" );**

**s1.Print();**

**cout <<"学生数为："<< CStudent::noofStds << endl; //通过类名引用静态数据**

**CStudent s2(10002, "李四" );**

**s2.Print();**

**cout << "学生数为："<< s2.noofStds << endl; //通过对象名引用静态数据**

**func();**

**cout << "学生数为："<< CStudent::noofStds << endl;**

**}**

**（4）静态成员函数**

**#include <iostream.h>**

**#include <string.h>**

**class Cmystring //定义一个字符串类**

**{**

**public :**

**Cmystring(char \* s) //定义构造函数，初始化字符串**

**{**

**len=strlen(s);**

**contents=new char[len+1];**

**strcpy(contents,s);**

**}**

**Cmystring(Cmystring & str) //拷贝构造函数**

**{ len=str.len;**

**contents=new char[len+1];**

**strcpy(contents,str.contents);**

**}**

**static int set\_total\_len() //求字符串的总长度**

**{ total\_len+= len;**

**return total\_len;**

**}**

**static int set\_total\_len(Cmystring str)**

**{**

**total\_len+= str.len;**

**return total\_len;**

**}**

**~Cmystring() //析构函数**

**{**

**delete []contents;**

**}**

**private:**

**//定义一个静态变量，用来存放所有字符串的总长度**

**static int total\_len;**

**int len; //字符串长度**

**char \* contents; //字符串的内容**

**};**

**int Cmystring:: total\_len=0; //静态数据成员初始化**

**void main()**

**{**

**//创建对象str1，并初始化**

**Cmystring str1("this is the first string");**

**//计算总长度，并输出**

**cout<<str1. set\_total\_len()<<”\n”;**

**//创建对象str2，并初始化**

**Cmystring str2("this is the second string");**

**cout<<str1. set\_total\_len()<<”\n”; //计算总长度，并输出**

**}**

5、#include<iostream.h>

#include<math.h>

class Cpoint

{

private:

float X,Y;

public:

Cpoint(float,float);

friend float pdistance(Cpoint &start,Cpoint &end);

};

Cpoint::Cpoint(float x,float y)

{

X=x;

Y=y;

}

float pdistance(Cpoint &start,Cpoint &end)

{

float d;

d=sqrt((end.X -start.X )\*(end.X -start.X )+(end.Y -start.Y )\*(end.Y -start.Y ));

return d;

}

void main()

{

Cpoint p1(1,1);

Cpoint p2(5,5);

cout<<pdistance(p1,p2)<<endl;

}

6、#include<iostream.h>

class Ctime;

class Cdate

{

private:

int year,month,day;

public:

Cdate(int y,int m,int d)

{

year=y;

month=m;

day=d;

}

void display(Ctime &t);

};

7、class Ctime

{

private:

int hour,minute,second;

friend void Cdate::display(Ctime &t);

public:

Ctime(int h,int m,int s)

{

hour=h;

minute=m;

second=s;

}

};

void Cdate::display(Ctime &t)

{

cout<<year<<" "<<month<<" "<<day<<endl;

cout<<t.hour <<t.minute <<t.second <<endl;

}

void main()

{

Cdate d(2015,4,10);

Ctime t(12,00,00);

d.display (t);

}

#include<iostream.h>

class Cdate;

class Ctime

{

private:

int hour,minute,second;

friend Cdate;

public:

Ctime(int h,int m,int s)

{

hour=h;

minute=m;

second=s;

}

};

class Cdate

{

private:

int year,month,day;

public:

Cdate(int y,int m,int d)

{

year=y;

month=m;

day=d;

}

void display(Ctime t)

{

cout<<year<<" "<<month<<" "<<day<<endl;

cout<<t.hour <<" "<<t.minute <<" "<<t.second <<endl;

}

};

void main()

{

Cdate d(2015,4,10);

Ctime t(12,00,00);

d.display (t);

}