1、定义学生类CStudent，包含数据成员姓名和学号，静态数据成员number保存学生的总数。定义类的构造函数、析构函数，Print成员函数输出学生信息和学生总数，静态成员函数GetTotal用于获得学生总数。

2、

友元函数

例1

#include<iostream.h>

#include<math.h>

class Cpoint

{

private:

float X,Y;

public:

Cpoint(float,float);

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

};

Cpoint::Cpoint(float x,float y)

{

X=x;

Y=y;

}

float distance(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<<distance(p1,p2)<<endl;

}

例2

#include <iostream.h>

class Rectangle

{

private:

float length,width;

public:

Rectangle(float,float);

bool IsRectangle();

friend void Area(Rectangle &R);

friend void Peri(Rectangle &R);

~Rectangle();

};

void Area(Rectangle &R)

{

cout<<"area="<<endl;

cout<<R.length \*R.width <<endl;

}

void Peri(Rectangle &R)

{

cout<<"peri="<<endl;

cout<<(R.length +R.width)\*2 <<endl;

}

Rectangle::Rectangle(float l,float w)

{

length=l;

width=w;

}

bool Rectangle::IsRectangle ()

{

if(length<=0||width<=0)

return false;

else

return true;

}

Rectangle::~Rectangle()

{

cout<<"析构了"<<endl;

}

void main()

{

Rectangle R1(2,3);

if (R1.IsRectangle()==true)

{

Area(R1);

Peri(R1);

}

}

===========================================

友元成员

#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);

};

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);

}

友元类

#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);

}