1. 单个文件编译

//ex1.cpp

#include <iostream>

using namespace std;

int main()

{cout<<"hello\n";

}

1. 两个文件的编译

//ex2.cpp

#include <iostream>

using namespace std;

int max(int,int);

int main()

{int x=3,y=5;

cout<<"max="<<max(x,y)<<endl;

}

//f1.cpp

int max(int a,int b)

{return((a>=b)?a:b);

}

1. 头文件的使用

//ex3.cpp

#include <iostream>

using namespace std;

#include "my.h"

int main()

{int x=3,y=5;

cout<<"z="<<x\*y\*M<<endl;

}

//my.h

#define M 10

1. 类与对象

//ex4.cpp

#include <iostream>

using namespace std;

class Test

{public:

void init(int x,int y)

{m=x;

n=y;

}

void print()

{w=m-n;

cout<<"m="<<m<<","<<"n="<<n<<","<<"w="<<w<<endl;

}

private:

int m;

int n;

int w;

};

int main()

{Test a;

a.init(68,55);

a.print();

}

将上述程序分为多个文件my1.h、myfun.cpp、ex4-1.cpp

其中：my1.h定义类，myfun.cpp定义类中函数及其它函数，ex4-1.cpp定义main()函数，并调用类对象及其它函数。

//my1.h

class Test

{public:

void init(int x,int y);

void print();

private:

int m;

int n;

int w;

};

//myfun.cpp

#include "my1.h"

#include <iostream>

using namespace std;

void Test::init(int x,int y)

{m=x;

n=y;

}

void Test::print()

{w=m-n;

cout<<"m="<<m<<","<<"n="<<n<<","<<"w="<<w<<endl;

}

void print1()

{cout<<"Ok\n";

}

//ex4-1.cpp

#include <iostream>

using namespace std;

#include "my1.h"

void print1();

int main()

{Test a;

a.init(68,55);

a.print();

print1();

}

g++ ex4-1.cpp myfun.cpp –o ex4-1

./ex4-1观察程序运行结果

1. 静态成员、静态成员函数、构造函数

//ex5.cpp

#include<iostream>

#include<string.h>

using namespace std;

class Stud

{

public:

Stud() {num++; }

void setdata(int n,char \*na,int d)

{no=n;

strcpy(name,na);

deg=d;

sum+=d;

}

static double avg()

{

return sum/num;

}

void disp()

{

cout<<no<<" "<<name<<" "<<deg<<endl;

}

private:

int no;

char name[10];

int deg;

static int sum;

static int num; };

int Stud::sum=0;

int Stud::num=0;

int main()

{

Stud s1,s2,s3;

s1.setdata(1,(char\*)"Li",89);

s2.setdata(2,(char\*)"Chen",78);

s3.setdata(3,(char\*)"Zheng",94);

s1.disp();

s2.disp();

s3.disp();

cout<<"average="<<Stud::avg()<<endl;

}

思考：将上述程序写在两个文件中（my2.h、ex5-1.cpp）

1. 友元

//ex6.cpp

#include <iostream>

#include <string.h>

#include <stdlib.h>

using namespace std;

class student

{private:

char \*name;

float score;

public:

void setname(char \*pname)

{name=new char[strlen(pname)+1];

strcpy(name,pname);

}

void setscore(float pscore)

{score=pscore;

}

char\* getname(){return(name);}

float getscore(){return(score);}

friend int compare1(student& ,student&);

};

int compare(student& x, student& y)

{if (x.getscore()>=y.getscore()) return 1;

else return 0;

}

int main()

{student stu[4],max,min;

int i,i1=0,i2=0;

char str[10];

float temp;

for(i=0;i<4;i++)

{cout<<"please input name:";

cin>>str;

stu[i].setname(str);

cout<<"please input score:";

cin>>temp;

stu[i].setscore(temp);

}

max=min=stu[0];

for(i=1;i<4;i++)

{if(!compare(max,stu[i])) {max=stu[i];i1=i;}

if(compare(min,stu[i])){min=stu[i];i2=i;}

}

cout<<"max="<<max.getscore()<<" "<<"name:"<<stu[i1].getname()<<endl;

cout<<"min="<<min.getscore()<<" "<<"name:"<<stu[i2].getname()<<endl;

}

1. 运算符重载（用友元函数重载，访问成员对象）

//ex7-1.cpp

#include <iostream>

#include <math.h>

using namespace std;

class Triangle

{private:

double a,b,c;

double area;

public:

Triangle(){a=0;b=0;c=0;}

Triangle(double a1,double b1,double c1){a=a1;b=b1;c=c1;}

double GetArea( )

{double s;

s=(a+b+c)/2;

area=sqrt(s\*(s-a)\*(s-b)\*(s-c));

return area;

}

friend double operator+( Triangle&, Triangle&);

};

double operator+( Triangle& m, Triangle& n)

{return m.area+n.area;

}

int main()

{ Triangle A(3,4,5),B(6,8,10);

A.GetArea();B.GetArea();

cout<<A+B<<endl;

}

1. 运算符重载（用友元函数重载，访问成员函数）

//ex7-2.cpp

#include <iostream>

#include <math.h>

using namespace std;

class Triangle

{private:

double a,b,c;

double area;

public:

Triangle(){a=0;b=0;c=0;}

Triangle(double a1,double b1,double c1){a=a1;b=b1;c=c1;}

double GetArea( )

{double s;

s=(a+b+c)/2;

area=sqrt(s\*(s-a)\*(s-b)\*(s-c));

return area;

}

friend double operator+( Triangle&, Triangle&);

};

double operator+( Triangle& m, Triangle& n)

{return m.GetArea( )+n.GetArea( );

}

int main()

{ Triangle A(3,4,5),B(6,8,10);

cout<<A+B<<endl;

}

1. 运算符重载（用成员函数重载，this指针的使用）

//ex7-3.cpp

#include <iostream>

#include <math.h>

using namespace std;

class Triangle

{private:

double a,b,c;

double area;

public:

Triangle(){a=0;b=0;c=0;}

Triangle(double a1,double b1,double c1){a=a1;b=b1;c=c1;}

double GetArea( )

{double s;

s=(a+b+c)/2;

area=sqrt(s\*(s-a)\*(s-b)\*(s-c));

return area;

}

void operator+( Triangle&m)

{cout<< GetArea( )+m.GetArea( )<<endl;}

};

int main()

{ Triangle A(3,4,5),B(6,8,10);

A+B;

}

1. 同时重载“+”和“>“两个运算符

//ex7-4.cpp

#include<iostream>

#include<math.h>

using namespace std;

class Triangle

{private:

int x,y,z;

double area;

public:

Triangle();

double getarea()

{ return area; }

void dispArea()

{cout<<area<<endl; }

friend double operator+(Triangle t1,Triangle t2)

{return t1.area+t2.area; }

friend bool operator>(Triangle t1,Triangle t2)

{return t1.area>t2.area ?true:false; }

};

Triangle::Triangle()

{

int i,j,k;

cout<<"input data:a b c";

cin>>i>>j>>k;

if(i+j>k && i+k>j && j+k>i)

{

double s;

x=i; y=j; z=k;

s=(x+y+z)/2.0;

area=sqrt(s\*(s-x)\*(s-y)\*(s-z));

}

else

{

cout<<"error!"<<endl;

area=x=y=z=0;

}

}

int main()

{

Triangle t1,t2;

cout<<"S1=";t1.dispArea();

cout<<"S2=";t2.dispArea();

if(t1>t2)

cout<<"S1>S2 "

<<t1.getarea()-t2.getarea()<<endl;

else

cout<<"S2>S1 "

<<t2.getarea()-t1.getarea()<<endl;

cout<<"S\_SUM="<<t1+t2<<endl;

}