1．类与对象（本题20分）

（1）创建一个点的类CPoint。该类有两个表示坐标的私有属性：x和y。

（2）定义一个无参数的构造函数，把x和y都初始化为0；

（3）定义一个具有两个参数的构造函数，分别用于初始化x和y；

（4）定义一个Display函数，用于输出x和y；

（5）创建主函数，利用两个构造函数创建两个点对象：p1(0,0)，p1(2,3)，并输出两个点的坐标值。

2．继承（本题20分）

（1）定义一个CPerson类，该类包含两个属性：ID和Name；

（2）在类CPerson中定义一个Display函数，用于显示ID和Name；

（3）由CPerson类派生一个CStudent类，该类中增加两个属性：StudentID和Grade；

（4）在类CStudent中定义一个Display函数，用于显示ID、Name、StudentID和Grade；

（5）在主函数中创建“ZhangSan”学生对象，其ID为320802197704102522，Name为张三，StudentID为11301308224，Grade为98。

3．运算符重载（本题20分）

（1）定义一个复数类CComplex，两个属性x和y分别用于表示实部和虚部；

（2）定义一个Display函数，用于输出复数的实部和虚部；

（3）定义两个复数对象的加法，如c1和c2是CComplex定义的对象，定义c1+c2（即运算符重载）；

（4）重载赋值运算符“=”，实现如c3=c1+c2；

（5）在主函数中定义三个复数对象：c1（1,2）、c2（3,4）和c3，c3=c1+c2，计算后输出c3的实部和虚部。

4． 设计平面坐标系中的点类CPoint：（本题20分）

（1） 数据成员有：横坐标double x、纵坐标double y；

（2） 提供两个构造函数：无参构造函数将数据成员初始化为（0，0），有参构造函数利用参数对数据成员赋值；

（3） 提供拷贝构造函数；

（4） 成员函数void moveto(double a,double b)，表示点移动到新的坐标位置（a,b）；

（5） 成员函数void display( )，输出点的坐标；

（6） 友元函数 double distance(CPoint &p1,CPoint &p2)，求两点的距离；（求平方根sqrt函数math.h中）；

（7） 定义主函数，完成点类的测试。

5．设计体积类（本题20分）

（1）定义CVolume类，包含求体积的纯虚函数virtual void display\_volume() =0;

（2）定义球类CSphere，公有继承CVolume，定义私有数据成员求半径double r，定义有参构造函数,重写CVolume类中的void display\_volume()函数，求球的体积；

（3）定义立方体类CCube，公有继承CVolume，定义私有数据成员棱长为double side，定义有参构造函数,重写CVolume类中的void display\_volume()函数，求立方体的体积；

（4）定义主函数，完成上述功能的测试。

1. #include<iostream.h>

class CPoint

{

private:

int x,y;

public:

CPoint()

{

x=0;

y=0;

}

CPoint(int x1,int y1)

{

x=x1;

y=y1;

}

void Display()

{

cout<<x<<' '<<y;

}

~CPoint()

{

cout<<"destroy!";

}

};

void main()

{

CPoint p1,p2(2,3);

p1.Display();

p2.Display();

}

2. #include<iostream.h>

#include<string.h>

class CPerson

{

private:

char ID[19];

char Name[10];

public:

CPerson(char \*id, char \*name)

{

strcpy(ID,id);

strcpy(Name,name);

}

void Display()

{

cout<<ID<<' '<<Name;

}

char \*GetID(){return ID;}

char \* GetName(){return Name;}

};

class CStudent:public CPerson

{

private:

char StudentID[12];

float Grade;

public:

CStudent(char \*id, char \*name,char \*studentid, float grade):CPerson(id, name)

{

strcpy(StudentID,studentid);

Grade=grade;

}

void Display()

{

cout<<GetID()<<' '<<GetName()<<' '<<StudentID<<' '<<Grade<<endl;

}

};

void main()

{

CStudent ZhangSan("320802197704102522","张三","11301308224",98);

ZhangSan.Display();

}

3. #include<iostream.h>

#include<string.h>

class CComplex

{

private:

int x,y;

public:

CComplex(int x1=0,int y1=0)

{

x=x1;

y=y1;

}

void Display()

{

cout<<x<<' '<<y<<endl;

}

CComplex operator+ (const CComplex &c)

{

CComplex t;

t.x=x+c.x;

t.y=y+c.y;

return t;

}

CComplex operator= (const CComplex &c)

{

x=c.x;

y=c.y;

return \*this;

}

};

void main()

{

CComplex c1(1,2),c2(3,4),c3;

c1.Display();

c2.Display();

c3=c1+c2;

c3.Display();

}

4.

using namespace std;

#include<iostream>

#include<math.h>

class CPoint

{

double x,y;

public:

CPoint(double x,double y)

{

this->x=x;

this->y=y;

}

CPoint()

{

x=0;

y=0;

}

CPoint(CPoint &p1)

{

this->x=p1.x;

this->y=p1.y;

}

void moveto(double a,double b)

{

x=a;

y=b;

}

void display()

{

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

}

friend double distance(CPoint &p1,CPoint &p2)

{

double d;

d=sqrt((p1.x-p2.x)\*(p1.x-p2.x)+(p1.y-p2.y)\*(p1.y-p2.y));

return d;

}

};

void main()

{

CPoint p1;

CPoint p2(7,9);

CPoint p3;

p1.display();

p2.display();

p1.moveto(4,5);

p1.display();

cout<<"p1和p2两点之间的距离："<<distance(p1,p2)<<endl;

p3=p1;

p3.display();

}

5. using namespace std;

#include<iostream>

class CVolume

{

public:

virtual void display\_volume() =0;

};

class CSphere:public CVolume

{

double r;

public:

CSphere(double r)

{

this->r=r;

}

void display\_volume()

{

cout<<"球的体积："<<4\*3.14\*r\*r\*r/3<<endl;

}

};

class CCube:public CVolume

{

double side;

public:

CCube(double side)

{

this->side=side;

}

void display\_volume()

{

cout<<"立方体的体积："<<side\*side\*side<<endl;

}

};

void main()

{

CVolume \*aa[2];

aa[0]=new CSphere(2);

aa[1]=new CCube(3);

for(int i=0;i<2;i++)

{

aa[i]->display\_volume();

}

}