1、验证下面程序，理解并掌握虚函数。

#include<iostream.h>

class A

{

public:

virtual void fun()

{

cout<<"Base A"<<endl;

}

};

class B1:public A

{

public:

void fun()

{

cout<<"Derived B1"<<endl;

}

};

class B2:public A

{

public:

void fun()

{

cout<<"Derived B2"<<endl;

}

};

void main()

{

A a1,\*p;

B1 b1;

B2 b2;

p=&a1;

p->fun();

p=&b1;

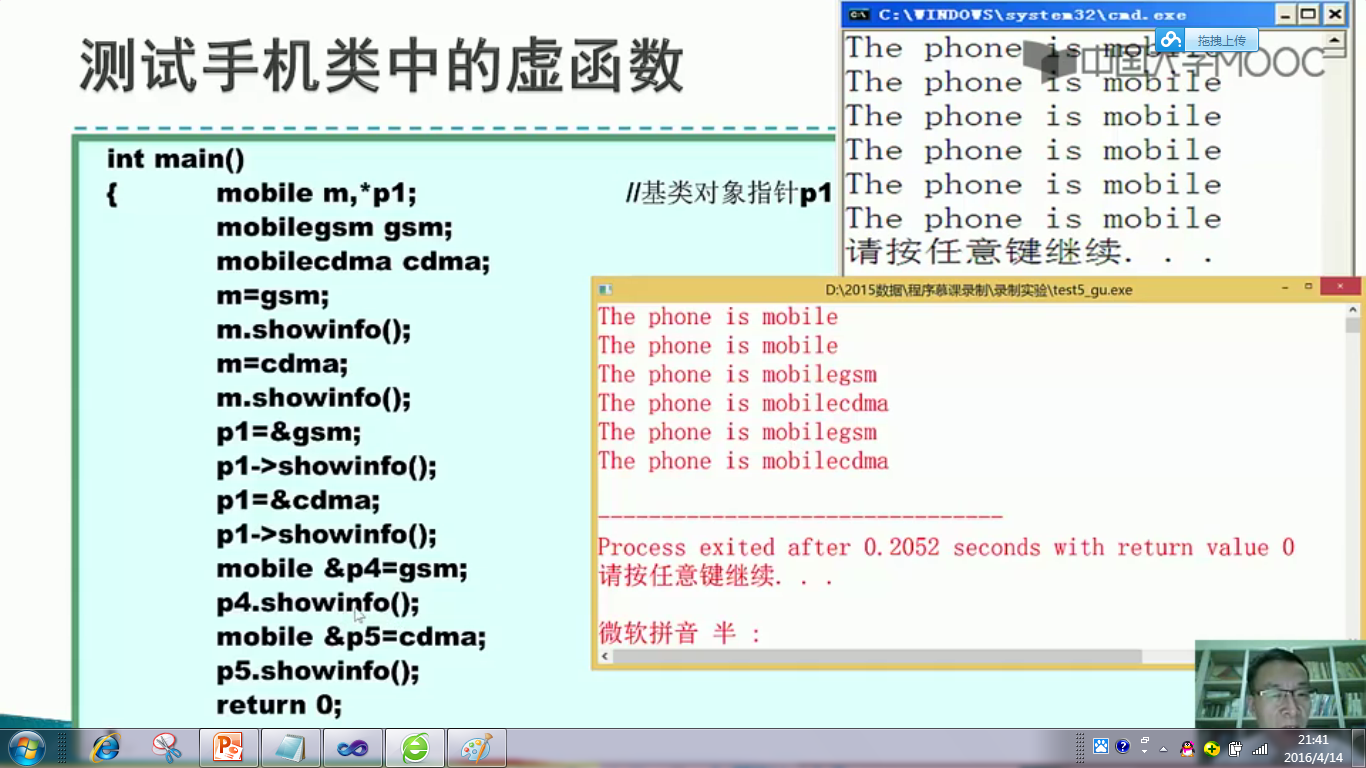
p->fun();

p=&b2;

p->fun();

}

2、定义两个类，一个为student类，再派生一个Gstudent类。定义一个虚函数分别实现输出“a student”和“a graduate student”。（分别用基类指针和基类引用两种方法验证）

3、完成手机虚函数测试。

4、有一个名为shape类，它有两个派生类：circle和square.

测试程序如下：

void main()

{

shape \*p1,\*p2;

p1=new circle(3,3.5,2);

p2=new square(4.5,5,6.2,8);

p1->print();

p2->print();

delete p1,p2;

}

要求：（1）设计3个类。分别完成以下功能：给出圆心坐标和半径，求圆周长；给出正方形的中心坐标和一个顶点坐标，求该正方形的周长。

（2）在shape类中定义纯虚函数print()，在circle类和square类中重载该函数求周长。

【参考代码】

2、

#include<iostream>

using namespace std;

class student

{

public:

virtual void print ()

{

cout<<"a student"<<endl;

}

};

class gstudent:public student

{

public:

void print()

{

cout<<"a graduate student"<<endl;

}

};

int main()

{

gstudent s4;

student s1,\*s2,&s3=s4;

s2=&s4;

s1.print ();

s2->print();

s3.print ();

}

4、

#include<iostream.h>

#include<math.h>

const double pi=3.1415;

class shape

{

protected:

float x,y;

public:

shape(float a,float b)

{

x=a;

y=b;

}

virtual void print()=0;

virtual ~shape()

{

}

};

class circle:public shape

{

private:

float r;

public:

circle(float a,float b,float c):shape(a,b)

{

r=c;

}

void print()

{

cout<<pi\*2\*r<<endl;

}

~circle()

{}

};

class square:public shape

{

private:

float d1,d2;

public:

square(float a,float b,float c,float d):shape(a,b)

{

d1=c;

d2=d;

}

void print()

{

cout<<4\*(fabs(d1-x)+fabs(d2-y))<<endl;

}

~square()

{}

};

void main()

{

shape \*p1,\*p2;

p1=new circle(3,3.5,2);

p2=new square(4.5,5,6.2,8);

p1->print();

p2->print();

delete p1,p2;

}