改为一般虚函数：

```
virtual void area() { }
```

然后仍然在主函数声明一个基类对象

```
Shape shape;
```

编译运行，观察与前一种情况有什么不同？为什么？

小 结

本章介绍了类的多态特性。C++支持的多态可以分为 4 类，即重载多态、强制多态、包含多态和参数多态。本章我们重点学习了包含多态。虚函数和抽象类是实现这种运行过程中多态的关键机制，因此也是我们学习的重点。读者应该在仔细研究本章所列例题的基础上认真实践正确掌握其使用方法。

习 题

一、选择题

1. 在公有派生的前提下，下列说法错误的是（ ）。
   A. 在一个赋值语句中，等号的左边是一个基类对象时，右边可以是一个派生类对象
   B. 若 B 是 A 的派生类，且有默认的构造函数，则语句 A &a=*new B; 是合法的
   C. 在一个返回值为基类指针的函数中，可以返回一个派生类对象的地址
   D. 在调用一个形参类型为派生类引用的函数时，可以将一个基类对象作为实参
2. 下列关于虚函数的表述中正确的是（ ）。
   A. 只有用 virtual 修饰的成员函数才是虚函数
   B. 派生类中覆盖虚函数的重定义函数仍然是虚函数
   C. 对虚函数的调用都是动态绑定的
   D. 使用虚函数主要是为了提高运行效率
3. 下列关于纯虚函数的表述中正确的是（ ）。
   A. 纯虚函数是只有接口没有实现的虚函数
   B. 含有纯虚函数的类不能用于定义对象，因此没有构造函数
   C. 纯虚函数的声明就是在虚函数声明的尾部加上修饰“=0”
   D. 派生类必须实现基类中声明的纯虚函数
4. 下列关于抽象类的表述中正确的是（ ）。
   A. 没有函数成员的类称为抽象类
   B. 含有虚函数成员的类称为抽象类
   C. 含有纯虚函数成员的类称为抽象类
   D. 用 abstract 修饰的类称为抽象类
5. 下列关于子类型的表述中错误的是（ ）。
   A. 如果 B 是 A 的子类型，则 A 也是 B 的子类型

B. 如果B是A的子类型，则B的对象可以初始化A的引用

C. 只有当B公有继承A，B才可能成为A的子类型

D. 如果B是A的子类型，C是B的子类型，则C是A的子类型

6. 有如下程序：

```
#include <iostream>
using namespace std;

class Base
{
public:
    void fun() { cout<<"Base::fun"<<endl; }
};
class Derived : public Base
{
    void fun()
    {
        _______                    //显式调用基类的函数 fun()
        cout<<"Derived::fun"<<endl;
    }
};
```

程序中下划线处缺失的语句是（　　）。

A. fun();　　B. Base.fun();　　C. Base::fun();　　D. Base->fun();

7. 虚函数必须是类的（　　）。

A. 成员函数　　B. 友元函数　　C. 静态函数　　D. 析构函数

本题不做 8. 在下面关于类设计的初步设想中，最有可能构成子类型关系的是（　　）。

A. 将“宠物狗”和“宠物猫”作为“宠物”的公有派生类

B. 将“技术员”和“项目经理”作为“雇员”的公有派生类

C. 将“英语老师”和“历史老师”作为“老师”的公有派生类

D. 将“办公室”和“会议室”作为“房间”的公有派生类

本题不做 9. 在下列各组关于C++多态性的不同描述中，实际上指的是同一种多态性的是（　　）。

A. 重载多态、基于函数覆盖的多态性、基于虚函数的多态性

B. 编译时的多态性、包含多态性、基于虚函数的多态性

C. 通过模板实现的多态性、基于函数重载的多态性、包含多态

D. 参数多态、运行时的多态性、通过模板实现的多态性

10. 有如下程序：

```
#include<iostream>
using namespace std;

class BASE{
    char c;
public:
    BASE(char n):c(n){}
    virtual~BASE(){cout<<c;}
};
class DERIVED:public BASE{
```

```
    char c;
public:
    DERIVED(char n):BASE(n+1),c(n){}
    ~DERIVED(){cout<<c;}
};
int main(){
    DERIVED('X');
    return 0;
}
```

执行上面的程序将输出（　　）。

A. XY　　B. YX　　C. X　　D. Y

11. 有如下程序：

```
#include<iostream>
using namespace std;

class AA{
public:
    virtual void f(){ cout<<"AA";}
};
class BB:public AA{
public:
    BB(){ cout<<"BB";}
};
class CC:public BB{
public:
    virtual void f(){ BB::f(); cout<<"CC";}
};
int main(){
    AA aa,*p;
    BB bb;
    CC cc;
    p=&cc;
    p->f();
    return 0;
}
```

执行上面的程序将输出（　　）。

A. BBAACC　　B. AABBCC　　C. BBAABBCC　　D. BBBBAACC

12. 有如下程序：

```
#include<iostream>
using namespace std;

class XX{
protected:
    int k;
public:
    XX(int n=5):k(n){}
    ~XX(){ cout<<"XX";}
    virtual void f()const=0;
};
```

```
inline void XX::f()const{ cout<<k+3;}
class YY : public XX {
public:
    ~YY(){ cout<<"YY";}
    void f()const{ cout<<k-3; XX::f();}
};
int main(){
    XX &p=*new YY;
    p.f();
    delete &p;
    return 0;
};
```

执行上面的程序将输出（　　）。

A. 28XX　　B. 28YYXX　　C. -33XX　　D. -33XXYY

二、填空题

1. 含有纯虚函数的类称为_____。

2. 已知一个类有一成员函数，其原型是：

```
Name excute(int, const char *)const;
```

现准备将之改为纯虚函数，其原型应改为_____。

3. 下面程序中的 Number 是一个抽象类，程序的输出是

```
67 11
```

程序中有缺失语句，请补充完整。

```
#include <iostream>
using namespace std;
class Number{
protected:
    int val;
public:
    Number(int i=0):val(i){}
    ________________
};

class HexNumber:public Number{
public:
    HexNumber(int i=0):Number(i){}
    void show()const{ cout<<hex<<val<<' '; }
};

class DecNumber:public Number{
public :
    DecNumber(int i=0):Number(i){}
    void show()const{ cout<<dec<<val<<' '; }
};

void Print(_________){ n.show( );}

int main( ){
```

```
    DecNumber d(67);
    Print(d);
    HexNumber h(17);
    Print (h);

    return 0;
}
```

4. 已知下面程序的输出如下：

```
This is class A
Who am I?
This is class C
```

程序中有缺失语句，请补充完整。

```
#include <iostream>
using namespace std;

class A{
public:
    virtual void printMe()const{ cout<<"This is class A"<<endl; }
};

class B:public A{
public:
    _____________________________
};

class C:public B{
public:
    void printMe()const{ cout<<"This is class C"<<endl; }
};

void show(_____________){ a.printMe();}

int main(){
    A a; show( a );
    B b; show( b );
    C c; show( c );

    return 0;
}
```

5. 下面程序的输出是_____。

```
#include<iostream>
using namespace std;

class BaseClass{
private:
    char *base;
public:
    BaseClass();
    ~BaseClass();
    virtual void Print() const{ cout << base ; }
```

```
};

class DerivedClass:public BaseClass{
private:
    char *derived;
public:
    DerivedClass();
    ~DerivedClass();
    virtual void Print() const{ cout << derived;}
};

BaseClass::BaseClass():base(new char[7]){
    memcpy(base, "base? ", 7);
    cout << "BASE? ";
}

BaseClass::~BaseClass(){
    delete [] base;
    cout << "BASE! " ;
}

DerivedClass::DerivedClass():derived(new char[10]){
    memcpy(derived, "derived? ",10);
    cout << "DERIVED? ";
}

DerivedClass::~DerivedClass(){
    delete [] derived;
    cout << "DERIVED! ";
}

int main(){
    BaseClass *basePtr = new DerivedClass;
    basePtr->Print();
    delete basePtr;

    return 0;
}
```

6. 下面程序的输出是_____。

```
#include<iostream.h>
class North{
private:
    double angle;
public:
    North (double a = 0) : angle(a){}
    virtual void Print()const{ cout << angle << endl; }
};

class East:virtual public North{
public:
    East(double a):North(a){}
};
```

```
class West:virtual public North{
public:
    West(double a):North(a){}
};

class South:public East,public West{
public:
    South():East(90),West(-90){}
};

int main(){
    South s;
    s.Print();
    return 0;
}
```

7. 实现编译时的多态性的机制称为_____，实现运行时的多态性的机制称为_____。

8. 在C++中，编译时的多态性是通过_____和模板体现的，运行时的多态性是通过_____体现的。

9. 有一种特殊的虚函数，重定义时不要求同名，这种虚函数是_____。

10. 一个纯虚函数声明的最后3个字符是_____。