

**EXAMPLE 62: IN YOUR DERIVED CLASSES, NEVER  
REDEFINE NON-VIRTUAL BASE CLASS METHODS**

EXAMPLE 62: IN YOUR DERIVED CLASSES, NEVER  
REDEFINE NON-VIRTUAL BASE CLASS METHODS

IF YOU DO, MYSTERIOUS BEHAVIOUR  
WILL FOLLOW

```
Rectangle rect;  
Shape * shapePtr = &rect;  
Rectangle * rectPtr = &rect;
```

```
shapePtr->print();  
rectPtr->print();
```

# 1 OBJECT

```
Rectangle rect;
```

```
Shape * shapePtr = &rect;
```

```
Rectangle * rectPtr = &rect;
```

```
shapePtr->print();
```

```
rectPtr->print();
```

# 1 OBJECT

```
Rectangle rect;
```

```
Shape * shapePtr = &rect;
```

```
Rectangle * rectPtr = &rect;
```

```
shapePtr->print();
```

```
rectPtr->print();
```

## 2 POINTERS POINT TO IT..

# 1 OBJECT

```
Rectangle rect;
```

```
Shape * shapePtr = &rect;
```

```
Rectangle * rectPtr = &rect;
```

```
shapePtr->print();
```

```
rectPtr->print();
```

## 2 POINTERS POINT TO IT..

### 1 BASE CLASS POINTER

# 1 OBJECT

```
Rectangle rect;
```

```
Shape * shapePtr = &rect;
```

```
Rectangle * rectPtr = &rect;
```

```
shapePtr->print();
```

```
rectPtr->print();
```

## 2 POINTERS POINT TO IT..

1 BASE CLASS POINTER

1 DERIVED CLASS POINTER

# 1 OBJECT

```
Rectangle rect;
```

```
Shape * shapePtr = &rect;
```

```
Rectangle * rectPtr = &rect;
```

CALL THE SAME METHOD ON BOTH..

```
shapePtr->print();
```

```
rectPtr->print();
```

## 2 POINTERS POINT TO IT..

1 BASE CLASS POINTER

1 DERIVED CLASS POINTER



```
Rectangle rect;  
Shape * shapePtr = &rect;  
Rectangle * rectPtr = &rect;  
CALL THE SAME METHOD ON BOTH..
```

```
shapePtr->print();  
rectPtr->print();
```

DIFFERENT THINGS GET PRINTED.

I am a shape

I am a rectangle



1 OBJECT

```
Rectangle rect;
```

```
Shape * shapePtr = &rect;
```

```
Rectangle * rectPtr = &rect;
```

CALL THE SAME METHOD ON BOTH..

2 POINTERS  
POINT TO IT..

```
shapePtr->print();
```

```
rectPtr->print();
```

DIFFERENT THINGS GET PRINTED.

```
I am a shape
```

```
I am a rectangle
```

CHAOS.

EXAMPLE 62: IN YOUR DERIVED CLASSES, NEVER  
REDEFINE NON-VIRTUAL BASE CLASS METHODS

IF YOU DO, MYSTERIOUS BEHAVIOUR  
WILL FOLLOW

```
Rectangle rect;  
Shape * shapePtr = &rect;  
Rectangle * rectPtr = &rect;
```

```
shapePtr->print();  
rectPtr->print();
```

# IN YOUR DERIVED CLASSES, NEVER REDEFINE NON-VIRTUAL BASE CLASS METHODS

```
class Shape
{
public:
    void print()
    {
        cout << "I am a shape " << endl;
    }
};
```

```
class Rectangle : public Shape
{
public:
    void print()
    {
        cout << "I am a rectangle " << endl;
    }
};
```

## IF YOU DO, MYSTERIOUS BEHAVIOUR WILL FOLLOW