15. Inheritance :: Revising the Design of the Square Class (continued)

```
Square

- mSideLen : int
- mX : int
- mY : int

+ Square () :
+ Square (pX : int, pY : int, pSideLen : int) :
+ getSideLen () : int
+ getX () : int
+ getY () : int
+ move (pNewX : int, pNewY : int) : void
+ resize (pNewSideLen : int) : void
+ setSideLen (pNewSideLen : int) : void
+ setX (pNewX : int) : void
+ setY (pNewY : int) : void
```

Employing what we have learned about inheritance, we can now redesign the Square class to take advantage of the existing Rectangle class. Since a Square is a Rectangle we know that if we make Square a subclass of Rectangle that Square will inherit all of the instance variables of Rectangle: mX, mY, mWidth, and mHeight. Square will also be able to call any of the **public** or **protected** methods of Rectangle (which is all of them).

15. Inheritance :: Revising the Design of the Square Class (continued)

Since Rectangle declares mX and mY and provides accessor/mutator methods getX(), get Y(), set X(), and set Y(), I hope you seethat that there is no reason for Square to duplicate those variables and methods. There is also no reason to duplicate the move() method because moving a Square is identical to moving a *Rectangle*. Furthermore, Square will inherit mWidth and mHeight so rather than having Square declare a new instance variable to represent the side length, we can simply reuse mWidth and mHeight, i.e., a Square is a Rectangle for which mWidth always equals mHeight.

```
Rectangle
- mHeight: int
- mWidth: int
- mX : int
- mY : int
+ Rectangle ():
+ Rectangle (pX : int, pY : int, pWidth : int, pHeight : int) :
+ getHeight (): int
+ getWidth (): int
+ getX (): int
+ getY(): int
+ move (pNewX : int, pNewY : int) : void
+ resize (pNewWidth: int, pNewHeight: int): void
+ setHeight (pNewHeight : int) : void
+ setWidth (pNewWidth : int) : void
+ setX (pNewX : int) : void
+ setY (pNewY : int) : void
```

Square

- + Square () :
- + Square (pX : int, pY : int, pSideLen : int) :
- + getSideLen (): int
- + resize (pNewSideLen : int) : void
- + setSideLen (pNewSideLen : int) : void