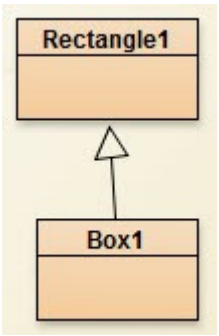
09.01 Virtual Lecture Notes



Extending classes can be demonstrated using the `Rectangle` and `Box` classes.

Look over the `Rectangle` class. Its definition should be familiar to you. It contains two private instance variables for the length and width, a constructor, and two get methods.

Now look at the `Box` class. Notice the class definition statement. We say the `Rectangle` class is extended to create the `Box` class. The first line of the `Box` class definition includes the word `extends`, followed by the name of the `Rectangle` class. It looks like this:

```
public class Box extends Rectangle
{
...
}
```

Now, a `Box` object will automatically inherit both the `getLength` and `getWidth` methods from `Rectangle`, since they are public. The instance variables `length` and `width` are private; therefore, they are not inherited. Box cannot reference them directly. In order to set the values when constructed, `Box` must use a call to the constructor of `Rectangle`, by making a special function call to the superclass. This is a call to the `super` method.

The `Box` constructor will look like this:

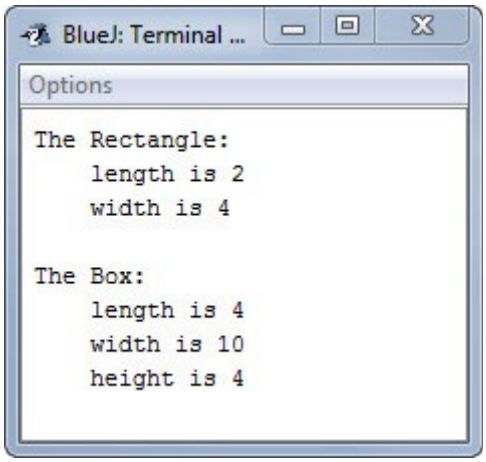
```
public Box(int l, int w, int h)
{
   // call super class
   super(l, w);

   // initialize instance variables
   height = h;
```

```
}
```

A call to `super` must be the first statement to occur in the constructor. Pass any variables it will need. In the case of Box, its call to `super` includes variables `l` and `w`, which are needed by the `Rectangle` constructor to set the value of `length` and `width`.

In addition to extending the `Rectangle` class and having access to its public methods, the `Box` class also contains one instance variable and one method to accommodate the height.

Run the `ShapesTester` program. The output should look similar to the image below:



Make changes and explore the classes to see how they behave. As a challenge, try to write another class such as `Square` or `Cube`. Don't forget the `super` call.

Print