School of Information Technologies  
The University of Sydney

**Objective**
- Understand the two reuse mechanism – delegate vs. inheritance
  - Practicing both options
  - Understand the differences.
- Practice using Adapter pattern.

**Tasks**
Suppose we have a `Rectangle` class with the following API specification:

```java
public class Rectangle extends java.lang.Object

The Rectangle class defines a geometry Rectangle and methods of calculating geometry facts

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Rectangle()</code></td>
<td>default constructor creates a Rectangle of width 4 and height 2</td>
</tr>
<tr>
<td><code>Rectangle(int width, int height)</code></td>
<td>create a Rectangle with given width and height</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int area()</code></td>
<td>get the area of a Rectangle</td>
</tr>
<tr>
<td><code>int getHeight()</code></td>
<td>retrieves the height of a Rectangle</td>
</tr>
<tr>
<td><code>int getWidth()</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>getPerimeter()</code></td>
<td>retrieves the width of a Rectangle</td>
</tr>
<tr>
<td><code>setWidth(int w)</code></td>
<td>set the width of a Rectangle</td>
</tr>
<tr>
<td><code>setWidth(int h)</code></td>
<td>set the height of a Rectangle</td>
</tr>
</tbody>
</table>

1. Create a **Square** class through reusing the **Rectangle** class. You need to practice both delegation and inheritance options.

2. Suppose we want to reuse **Rectangle** class as a **Shape** in the following application. How can you achieve it?

```java
interface Shape{
    int getPerimeter();
    String getName();
    int getArea();
}

class Circle implements Shape{
    int r;
    public Circle(int r){
        this.r = r;
    }

    public int getPerimeter(){
        return (int) 3.14 * 2 * r;
    }

    public int getArea(){
        return (int) 3.14 * r * r;
    }

    public String getName(){
        return "circle:" + r;
    }
}

class ShapeCollection{
    Shape[] shapes = new Shape[20];

    void printArea(){
        for (int i = 0; i < shapes.length; i++){
            System.out.println(shapes[i].getName()+ " : " + shapes[i].getArea());
        }
    }
}
```