**Shapes Application**

**Step 1**

Compile and run the code that can be found in the folder  [http:\\ecenotes\ce4208\Assignment2](file:///\\ecenotes\ce4208\Assignment2)

Checkout and try to understand the code.

**Step 2**

Extend the code above to generate random shapes such as lines, ovals and rectangles to randomly draw rectangles and ovals. Create classes MyRect and MyOval (you may need to consult the class files from **step 1** to find the correct shape descriptions).

Both of these classes should include x1, y1, x2, y2 cordinates, a color, and a Boolean flag to indicate if the shape should be filled.

Declare a constructor in each class with arguments for initialising all the instance variables.

To help draw the rectangles and ovals each class should provide methods getUpperLeftX, getUpperLeftY, getWidth and getHeight that calculate the upper-left coordinate x-coordinate, upperLeftX, getWidth, geteHeight.

The upper-left x-coordinate is the smaller of the two x-coordinates, the upper-left y-coordinate is the smaller of the two y-coordinate values. The width is the absolute value of the difference between the two y-coordinate values.

Class Drawpanel, which extends JPanel and handles the creation of the shapes, should declare three arrays, one for each shape type, The length of each array should be a random number between one and five. The constructor of class DrawPanel will fill each of the arrays with shapes of random position, size and shape.

In addition, modify all three shape classes to include the following:

1. A constructor with no arguments that sets all the coordinates on the shape to 0, the colour of the shape to Color.BLACK, and the filled property to false (MyRectangle and MyOval only).
2. *Set* methods for the for the instance variables in each class. The methods that set a coordinate value should verify that the argument is greater than or equal to zero. If not set the coordinate to zero.
3. *Get* methods for the instance variables each class. Method **draw** should reference the coordinates by the **get** methods rather than accessing them directly.
4. You will need to use random number generators to create the shapes.

Note: this is not a trivial exercise you may need help to implement the code.

The next assignment will cover of serialisation and polymorphism.

**Due date: 28th February 2011**