

## Using Applets to Draw Shapes

The following is a simple applet that draws a blue rectangle on a yellow background.

```
// *****
// RandomShapes.java
//
// The program will draw two filled rectangles and a
// filled oval positioned randomly on the screen.
// *****

import java.applet.Applet;
import java.awt.*;

public class RandomShapes extends Applet
{
    public void paint (Graphics page)
    {
        // Declare size constants
        final int MAX_SIZE = 300;
        final int PAGE_WIDTH = 600;
        final int PAGE_HEIGHT = 400;

        // Declare variables
        int x, y; // x and y coordinates of upper left-corner of each shape
        int width, height; // width and height of each shape

        // Set the background color
        setBackground (Color.yellow);

        // Set the color for the next shape to be drawn
        page.setColor (Color.blue);

        // Assign the corner point and width and height
        x = 200;
        y = 150;
        width = 100;
        height = 70;

        // Draw the rectangle
        page.fillRect(x, y, width, height);
    }
}
```

**Study the code, noting the following:**

- \* The program imports `java.applet.Applet` because this is an applet, and it imports `java.awt.*` because it uses graphics.
- \* There is no main method -- instead there is a `paint` method. The `paint` method is automatically invoked when an applet is displayed, just as the main method is automatically invoked when an application is executed.

- \* Most of the methods that draw shapes (see the list in Figure 2.18 in your textbook) require parameters that specify the upper left-hand corner of the shape (using the coordinate system described in textbook Section 1.6) and the width and height of the shape. You can see this in the calls to `fillRect`, which draws a rectangle filled with the current foreground color.
- \* This applet will be drawn assuming the window for drawing (the Graphics object - named "page" here) is 600 pixels wide and 400 pixels high. These numbers are defined in constants at the beginning of the program. (They currently have no use but you will use them later). The width and height of the applet are actually specified in an HTML file that tells how to run the applet (remember applets are often executed by Web browsers and Web browsers get their instructions from HTML documents -- note that the code executed by the browser is the bytecode for the program, the `RandomShapes.class` file). The code in an HTML document would be as follows:

```
<html>
<applet code="RandomShapes.class" width=600 height=400>
</applet>
</html>
```

Save the files `RandomShapes.java` and `RandomShapes.html` (save `RandomShapes.html` by right clicking on your mouse then clicking on `Save target As ...`). Notice if the html file was saved as `.html` or `.htm`.

### Now do the following:

1. Open a command prompt window (use `Start -> Run cmd`) and change to the proper directory. Now compile `RandomShapes.java` using **javac**, but do **not** run it with **java**. This is an applet, not a stand alone Java application (it does not have a main method), so it is run with a special program called the Applet Viewer.

2. Within the command window, run the program using the Applet Viewer by typing the command

```
appletviewer RandomShapes.html (or .htm)
```

You are now running the `.html` (or `.htm`) file.

You should see a new window open displaying the rectangle.

3. Now using **Eclipse**, execute `RandomShapes.java` by clicking on `Run -> Run As -> Java Applet` (within Eclipse). Be sure to enlarge the applet window so that all of the output shows on your screen.
4. Now change the `x` and `y` variables both to 0. Save and recompile the program, then run it again. What happened to the rectangle?
5. Now change the width to 200 and the height to 300. Save, recompile and run to see how this affects the rectangle.
6. Change `x` to 400, `y` to 40, width to 50 and height to 200. Test the

program to see the effect.

7. Modify the program so the position and size of the rectangle is random. To do this you need to
  - o Add the command to import the `java.util.Random` class.
  - o Declare and instantiate a `Random` object named `generator`
  - o Modify the assignment statements to assign `x` a random value between 0 and `PAGE_WIDTH` and `y` a random value between 0 and `PAGE_HEIGHT` (use these constant identifiers). Your random values should go up to but not include these maximums.
  - o Modify the assignment statements to assign `width` and `height` random values between 50 and `MAX_SIZE + 50`.Save, recompile, and run the program to test the changes.
8. Now add two more random rectangles -- this only requires duplicating the code you already have, so cutting (or copying) and pasting come in handy. Test the changes.
9. One last touch to the program ... Change the colors for at least two of the shapes so the background and each of the three shapes are different colors (a list of colors is in Figure 2.21 of the text). Also change one of the `fillRect` methods to `fillOval` so the final program draws two randomly positioned and sized rectangles and one oval.