Applets are java programs that can run embedded in a web page. 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:

* Notice on the first line that it extends from **Applet** and in the body it overrides the **paint** method**.**
* There is no *main* method. When an applet is displayed on a web page, the paint method is automatically invoked, 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) require parameters that specify the upper left-hand corner of the shape (using the coordinate system described in 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 the HTML file that instructs the Web browser to run the applet (remember applets are 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 the HTML document is as follows:

<html>

<h1>Here is my Applet</h1>

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

</applet>

</html>

# Project Instructions

Make a project to run this Applet as follows:

1. Create a Java project in Eclipse (this time it needs to be a Java project). Create a class called RandomShapes and paste the java above into it.
2. Now create an HTML web page in the project (click new, then file). Call it shapes.html. Save it.
3. Right click on the new file, and open with the text editor (open with / text editor).
4. Paste the HTML above into it and save it.
5. Now open the web page using the browser outside of Eclipse (right-click the file, open with, system editor.) I don’t think the Applet will work on the web browser embedded in Eclilpse.
6. Now try running the Applet in Eclipse, like a normal java project. This won’t use the HTML, but it will run it in a special window called an appletviewer. You may need to adjust the size of the Applet window to see the rectangle.
7. In the program change the x and y variables both to 0. Rerun the program (or bring it up in a web browser again). (The Applet Viewer is generally less trouble when making lots of changes than using the browser, you might have to stop/restart the browser to see the changes to the Applet). What happened to the rectangle?
8. Now change the width to 200 and the height to 300. Run it again to see how this affects the rectangle.
9. Change x to 400, y to 40, width to 50 and height to 200. Test the program to see the effect.
10. Modify the program so the position and size of the rectangle so that x is a random value between 0 and PAGE\_WIDTH, and y is a random value between 0 and PAGE\_HEIGHT (use these constant identifiers).

Modify the assignment statements to assign width and height random values between 50 and MAX\_SIZE + 50.

Save and run the program to test the changes.

1. 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.
2. 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.
3. After testing your program in the Applet Viewer, bring it up again in your browser. Now keep pressing the refresh button. Each time you do the program executes again, computing new random values for the positions and sizes of the shapes.