**Introduction to JAVA (Continued) – Part 4**

# D001 — Drawing a Line

The HSA console can be used for graphics as well as text.

When drawing graphics, you must work with the "Cartesian Coordinate System", meaning you must specify (x, y) coordinates when drawing objects. The standard console is 640 (width) x 500 (height) pixels, with (0, 0) being in the top left corner and (639, 499) in the bottom right corner.

The most basic graphic shape that can be drawn is the line. The syntax for drawing a line is: c.drawLine (x1, y1, x2, y2);

This command draws a line from (x1, y1) to (x2, y2), where all four variables are integers. A typical command would look like:

c.drawLine (0, 0, 340, 250);

# Assignment:

Write a program that draws 4 lines in the middle of the screen in the shape of a 200 x 100 pixel rectangle.

Save as "D001.java".

# D002 — Drawing a Rectangle

The syntax for drawing a rectangle is as follows: c.drawRect (x, y, width, height);

...where *x* and *y* define the location of the top left corner.

# Assignment:

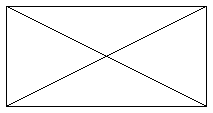
Modify D001 so it uses the drawRect() method instead of four drawLine() methods. The rectangle should be drawn in the same position in both programs.

Save as "D002.java".

# D002a — Drawing a Flag

**Assignment:**

Use the drawLine() and drawRect() methods to draw the following 200 x 100 pixel flag. Be sure the flag is centred on the console.



Save as "D002a.java".

# D003 — Drawing an Oval

The drawOval() syntax is very similar to the drawRect() syntax: drawOval (x, y, width, height);

# Assignment:

Modify D002a so it draws a perfect circle in the middle of the flag. Save as "D003.java".

# D004 — Setting the Colour

To set the colour of any graphics symbol you would first use the setColor() command. A typical setColor() command looks like this:

c.setColor (Color.red);

# Assignment:

Write a quick program that draws various shapes to test this command. Experiment with the colour to determines as many colours as possible.

Save as "D004.java".

# D005 — Other Graphic Shapes

Following is a complete list of all console class shapes:

clearRect (int x, int y, int width, int height)

Clears the rectangle to the background color.

copyArea (int x, int y, int width, int height, int deltaX, int deltaY)

Copies the rectangle defined by the upper-left corner (x, y) with width of width and height of height to a position moved by deltaX and deltaY pixels.

draw3DRect (int x, int y, int width, int height, boolean raised) Draws a 3-D rectangle. It appears raised if raised is true.

drawArc (int x, int y, int width, int height, int startAngle, int arcAngle)

Draws an arc. The arc is inscribed in the rectangle defined by the upper- left corner (x, y) with width of width and height of height. It starts at startAngle degrees and goes counterclockwise for arcAngle degrees.

drawLine (int x1, int y1, int x2, int y2) Draws a line from (x1, y1) to (x2, y2).

drawMapleLeaf (int x, int y, int width, int height)

Draws a maple leaf. The maple leaf is inscribed in the rectangle defined by the upper-left corner (x, y) with width of width and height of height.

drawOval (int x, int y, int width, int height)

Draws an ellipse. The ellipse is inscribed in the rectangle defined by the upper-left corner (x, y) with width of width and height of height.

drawPolygon (int[] xPoints, int[] yPoints, int numPoints)

Draws a polygon. The xPoints and yPoints arrays define the coordinates of the array of vertices. numPoints specifies the number of vertices in the polygon.

drawRect (int x, int y, int width, int height)

Draws a rectangle with upper-left corner at (x, y) with width of width and height of height. drawRoundRect (int x, int y, int width, int height, int arcWidth, int arcHeight)

Draws a rectangle with rounded corners with upper-left corner at (x, y) with width of width and height of height. arcWidth and arcHeight are the width and height of the ellipse used to draw the rounded corners.

drawStar (int x, int y, int width, int height)

Draws a star. The star is inscribed in the rectangle defined by the upper-left corner (x, y) with width of width and height of height.

drawString (String str, int x, int y)

Draws the string str at the starting point (x, y). The y coordinate is the base line of the text. fill3DRect (int x, int y, int width, int height, boolean raised)

Draws a filled 3-D rectangle. It appears raised if raised is true. fillArc (int x, int y, int width, int height, int startAngle, int arcAngle)

Draws a filled arc. The arc is inscribed in the rectangle defined by the upper-left corner (x,

y) with width of width and height of height. It starts at startAngle degrees and goes counterclockwise for arcAngle degrees.

fillMapleLeaf (int x, int y, int width, int height)

Draws a filled maple leaf. The maple leaf is inscribed in the rectangle defined by the upper-left corner (x, y) with width of width and height of height.

fillOval (int x, int y, int width, int height)

Draws a filled ellipse. The ellipse is inscribed in the rectangle defined by the upper-left corner (x, y) with width of width and height of height.

fillPolygon (int[] xPoints, int[] yPoints, int numPoints)

Draws a filled polygon. The xPoints and yPoints arrays define the coordinates of the array of vertices. numPoints specifies the number of vertices in the polygon.

fillRect (int x, int y, int width, int height)

Draws a filled rectangle with upper-left corner at (x, y) with width of width and height of height.

fillRoundRect (int x, int y, int width, int height, int arcWidth, int arcHeight)

Draws a filled rectangle with rounded corners with upper-left corner at (x, y) with width of width and height of height. arcWidth and arcHeight are the width and height of the ellipse used to draw the rounded corners.

fillStar (int x, int y, int width, int height)

Draws a filled star. The star is inscribed in the rectangle defined by the upper-left corner (x, y) with width of width and height of height.

*Source:* [*http://www.holtsoft.com/java/hsa\_package.html#Con-Graphics*](http://www.holtsoft.com/java/hsa_package.html#Con-Graphics)

There is no assignment for this lesson.

# D006 — Drawing a Flag II

**Assignment:**

Using the relevant methods presented in D004 and D005, draw a Canadian flag.



Save as "D006.java".

# D007 — Drawing a Landscape

**Assignment:**

Draw a landscape containing a house with a chimney, a front door, two windows, a horizon line, a tree with leaves and branches, clouds, and puffs of smoke coming from the chimney.

Use appropriate colours for the various shapes (green grass, blue sky, etc.). Save as "D007.java".