Unit Five Lecture Four:

Topic 1: Two Dimensional Arrays (Matrix)

A two dimensional array is an array consisting of both rows and columns of values. When declaring a two dimensional array, you must specify how many rows and columns you need.

```
For example:   final int ROWS = 2;
               final int COLUMNS = 3;
               int[][] matrix = new int[ROWS][COLUMNS];
```

This line of code creates the following two dimensional array.

```
matrix[0][0]     matrix[0][1]     matrix[0][2]

    0                0                0

matrix[1][0]     matrix[1][1]     matrix[1][2]

    0                0                0
```

Like one dimensional arrays, it is an error to try to access nonexistent positions. A first index < 0 or >= 2 or a second index < 0 or >= 3 will cause the program to terminate.

To change a value in the matrix array, you need to specify which position in the array you want to access. The following code will change all values in the matrix array.

```
     int c = 1;

     for (int j=0; j<ROWS; j++)
     {
          for (int k=0; k<COLUMNS; k++)
          {
               matrix[j][k] = c;
               c++;
          }
     }
```

The matrix array now looks like this...

```
matrix[0][0]     matrix[0][1]     matrix[0][2]

    1                2                 3

matrix[1][0]     matrix[1][1]     matrix[1][2]

    4                5                 6
```

In Java, you are allowed to list all elements that you want to include in the two dimensional array. They must be enclosed in braces and separated by commas. Each new set of braces indicates a new row of data.

```
For example:   double[][] array2D = {{8.3, 5.9},
                                     {4.8, 3.2},
                                     {6.1, 2.4}};
```

The Java compiler counts how many rows and columns you want in your array, allocates an array of the correct size, and fills it with the elements that you specify. The *new* command is not used in this situation.

The two dimensional array, array2D, would look like this...

```
array2D[0][0]     array2D[0][1]

    8.3               5.9

array2D[1][0]     array2D[1][1]

    4.8               3.2

array2D[2][0]     array2D[2][1]

    6.1               2.4
```

Each row of this two dimensional array is really a one dimensional array and has an instance variable length, which you can access to find out the size of a row (# of columns). A 2 dimensional array also has an instance variable length, which will give you the number of rows.

```
For example:   array2D[0].length = 2
               array2D.length = 3
```

Topic 2: The JFrame

The JFrame is a Java application equivalent to a JApplet. Both the JFrame and the JApplet use a Container to hold all of the user interface components. In a Java application, this Container is filled in the default constructor or a method called from inside the default constructor. In a Java applet, the Container is filled in public void init().

The following code shows you how you could print "Generation 0" of the Game of Life to a JTextArea in a Container attached to a JFrame.

```
import javax.swing.JFrame;
import javax.swing.JTextArea;
import java.awt.Container;
import java.awt.Font;

public class GameOfLife extends JFrame
{
      private JTextArea output;
      private String board[][] = {{" ", " ", " ", " ", " ", " "},
                                  {" ", " ", "*", " ", " ", " "},
                                  {" ", " ", " ", "*", " ", " "},
                                  {" ", "*", "*", "*", " ", " "},
                                  {" ", " ", " ", " ", " ", " "},
                                  {" ", " ", " ", " ", " ", " "}};

      public GameOfLife()        // default constructor
      {
            Container container = getContentPane();
            output = new JTextArea();
            output.setFont(new Font("Monospaced", Font.PLAIN, 10));
            output.setText("Generation 0\n");
            container.add(output);

            PrintMatrix();

            setSize(200, 500);    // sets size of JFrame
            setVisible(true);     // makes JFrame visible
      }

      public void PrintMatrix()
      {
            for (int j=0; j<6; j++)
            {
                  output.append("|");
                  for (int k=0; k<6; k++)
                  {
                        output.append(board[j][k] + "|");
                  }
                  output.append("\n");
                  output.append("-------------\n");
            }
      }
      public static void main(String args[])
      {
            GameOfLife application = new GameOfLife();  // calling default
                                                        // constructor

            application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            // closes application when you close the JFrame
      }
}
```

Assignment U5A4: The Game of Life