08.05 Virtual Lecture Notes: Two-Dimensional Array

A two-dimensional array can be used to manipulate data. For example, imagine if you had a table of student test scores. Each row would be a different student and each column a different test score. The table may look something like this:

```
            Test  Scores
Student       1    2    3    4
      1      98  100   75   89
      2     100   95   97   85
      3      67   77   79   80
      4     100   99   95   87
      5     100   95   90   85
```

You can create a two-dimensional array with five rows and four columns using the following statement:

```
int[][] testScores = new int[5][4];
```

Each position within the table is initialized to 0. Run the demo program and view the output. You would declare a two-dimensional array like this if you plan to read in the values for the table from a text file or calculate the values later using a method that traverses the array and populates it.

If you want to preload the array with values, then you create the array this way:

```
int[][] testScores = { { 98, 100, 75, 89},
                       {100,  95, 97, 85},
                       { 67,  77, 79, 80},
                       {100,  99, 95, 87},
                       {100,  95, 90, 85} };
```

To access a value within the array, the row and column index values are used. For instance, `testScores[0][3]` represents the value in the first row, last column which is 89.

Now that we have our array, we can manipulate however we want. For example, let's say the instructor wanted to be able to curve a test by adding points to everyone's score; you could use a method like the one below to allow the user to specify which test number and the value of the curve.

```
   public static void curveTest(int[][] scores, int testNum, int
value)
   {
     int col = testNum - 1;
     for(int row = 0; row < scores.length; row++)
     {
        scores[row][col] += 5;
     }
   }
```

Note that this only modifies one test score by using a fixed value for the column value and then using the loop to go through the table, row by row.

Often, you will need a method to traverse your two-dimensional array, such as when printing out the table of test scores. You can do that using two nested for loops. The outer loop cycles through the rows and the inner loop the columns.

```
for(int row = 0; row < scores.length; row ++)
{
   System.out.printf("%5d   ", row + 1);
   for(int col = 0; col < scores[row].length; col++)
   {
     System.out.printf("%5d", scores[row][col]);
   }
   System.out.println();
}
```

The `printScores` method walks through the array one row at a time. The outer loop steps through the table row by row. When on a particular row, the inner loop steps through each column.

Using the `Test2D.java` demo program, step through it to make sure you understand how to work with two-dimensional arrays.

Print