16.216: ECE Application Programming
Fall 2011

Lecture 31: Key Questions
November 30, 2011

1. Describe how to declare, initialize, and access two-dimensional arrays.

2. **Example:** Complete the following program:

```c
#include <stdio.h>
#define NRows 3     // # of rows
#define NCols 4     // # of columns
int main() {
  double x[NRows][NCols] =          // 2-D array
           { {  10,  2.5,    0,  1.5},
             {-2.3, -1.1, -0.2,    0},
             {10.5, -6.1, 23.4, -9.2} };

  int negCnt[NRows] = {0};    // Initialize entire row
                              // count array to 0
  int i, j;                   // Row and column indices

  /* INSERT CODE HERE--Visit every element in array x and
     count the number of negative values in each row */




  // Now print the row counts
  for (i = 0; i < NRows; i++)
    printf("Row %d has %d negative values.\n",
             i, negCnt[i]);

  return 0;
}
```

3. Explain how 2-D arrays are passed to functions.

4. **Example:** Say we have a program that stores student exam scores in a 2-D array:
   - Each row represents an individual student
   - Each column represents one of the 3 exams

Write functions to:
   - Calculate the exam average for each student and store it in a 1-D array that is accessible in the main program
     - Assume all exams have equal weight
   - Calculate the average for each exam and store it in a 1-D array that is accessible in the main program
   - Each function takes the same arguments:
     - The 2-D array
     - The # of students in the class
     - The 1-D array that will be used to hold the averages

4 (cont.) Extra space to write functions described previously.