16.216: ECE Application Programming

Summer 2012

Lecture 8: Key Questions
August 7, 2012

1. Explain the use of arrays: what an array represents, how to define an array, and how to access values within the array.

2. Explain how the following example works:

```
int main(void)
{
  int x[8];
  int i;

   // get 8 values into x[]
   for (i=0; i<8; i++)
   {
        printf("Enter test %d:",i+1);
        scanf("%d",&x[i]);
   }
}
```

3. What happens if we change the loop condition to `i <= 8`? How can we avoid this problem?

4. **Example:** What does the following program print?

```
int main() {
	int arr[10];
	int i;

	printf("First loop:\n");
	for (i = 0; i < 10; i++) {
		arr[i] = i * 2;
		printf("arr[%d] = %d\n", i, arr[i]);
	}

	printf("\nSecond loop:\n");
	for (i = 0; i < 9; i++) {
		arr[i] = arr[i] + arr[i + 1];
		printf("arr[%d] = %d\n", i, arr[i]);
	}
	return 0;
}
```

5. Explain how to pass arrays to functions.

6. **Example:** Write a function for each of the following:

- Given an array of doubles (`arr`) and the # of elements in the array (`n`), find the average of all array elements

6 (cont.) **Example:** Write a function for each of the following:

- Given an array of ints and the # of elements, find the largest element in the array

- Given an array of test scores (`tests`), the # of elements in the array (`n`), and an amount to scale those scores by (`s`), add `s` to every element in `tests` and print the scaled scores

7. Explain the relationship between pointers and arrays.

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

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

```
#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;
}
```

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

11. **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

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