

# Using Multi-Dimensional Arrays

Last updated 8/16/23

These slides show some examples of using multi-dimensional arrays

# Multi-Dimensional Arrays

- Example – student scores
  - Read the 8 scores for 10 students from the keyboard and store them in a 2 dimensional array

```
int scores[10][8];
int row;
int col;
for(row = 0; row < 10; row++)
    for(col=0; col < 8; col++)
        scanf("%i", &scores[row][col]);
```

col and row used for clarity  
Could be x and y, foo and boo, ...

notes:

no {} since one line for each for

inner loop – columns (grades)

outer loop – rows (students)

reads all 8 scores for a student  
then goes to the next student

&scores[row][col] refers to a  
single element (address)

note: still need '&' since  
scores[i] is not a pointer (address)  
It is an individual value

# Multi-Dimensional Arrays

- Example - print
  - Print the scores for 10 students from a 2 dimensional array to the console

```
int row;  
int col;  
for(row = 0; row < 10; row++){  
    for(col=0; col < 8; col++)  
        printf("%i", scores[row][col]);  
    printf("\n");  
}
```

notes:

inner loop – columns (grades)  
outer loop – rows (students)  
prints all 8 scores for a student  
then goes to the next student

# Multi-Dimensional Arrays

- Assignment

- Whole arrays **cannot** be used on the right side of an assignment operator

```
int array1[10][8];
```

```
int array2[10][8];
```

```
...
```

```
array2 = array1
```

# Multi-Dimensional Arrays

- Assignment
  - Arrays must be copied element by element

```
int array1[10][8];
int array2[10][8];
...
int row;
int col;
for(row = 0; row < 10; row++)
    for(col=0; col < 8; col++)
        array2[row][col] = array1[row][col];
```

notes: FOR COPY  
order does not matter  
rows or col in outer loop

# Multi-Dimensional Arrays

- Example
  - Multiply all the values in a 2D array by 5

```
int array1[10][20];
int row;
int col;
...
for(row = 0; row < 10; row++){
    for(col = 0; col < 20; col++){
        array1[ row ][ col ] = array1[ row ][ col ] * 5;
    }
}
```