



# Multi-Dimensional Arrays

CS2263 – Systems Software Development

1

## Learning Outcomes

At the conclusion of this lecture students should be able to:

- Demonstrate how to reference elements of multi-dimensional arrays
- Explain how multi-dimensional arrays can be passed to functions



2

## Defining Arrays

- To define an array:

```
type arrayName[size][size];
```

- For example

```
1. int id[1000][10];
2. int location[1000][1000][1000];
3. #define STUDENTS 10000
   #define TESTS 10;
   double scores[STUDENTS][TESTS];
```

- As in Java, multi-dimensional arrays are “vectors of vectors”



3

## Array Compile-time Initialization

- Dimensions are filled in-order, if no direction

```
int a[2][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

- a[0][0] value is 1
- a[1][0] value is 2
- a[0][1] value is 3
- a[1][1] value is 4

- Most common form:

```
int a[2][3] = {
    {1, 2, 3},
    {4, 5, 6}
};
```

- Can omit only 1 dimension of array when initializing

```
int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```



4

## Processing Idiom

```
1  for(int i=0; i<xN; i++){
2      for(int j=0; j<yN; j++){
3          for(int k=0; k<zN; k++){
4              a[i][j][k] = 0;
5          }
6      }
7  }
```



5

## Multi- Dimensional Arrays and Functions

- Regardless of dimensions, arrays are passed as an address
- N-1 dimensions must be defined in the function declaration
  - `void f1(a[2][3][], int n){...}`
  - Needed for C to interpret the dimensions



6