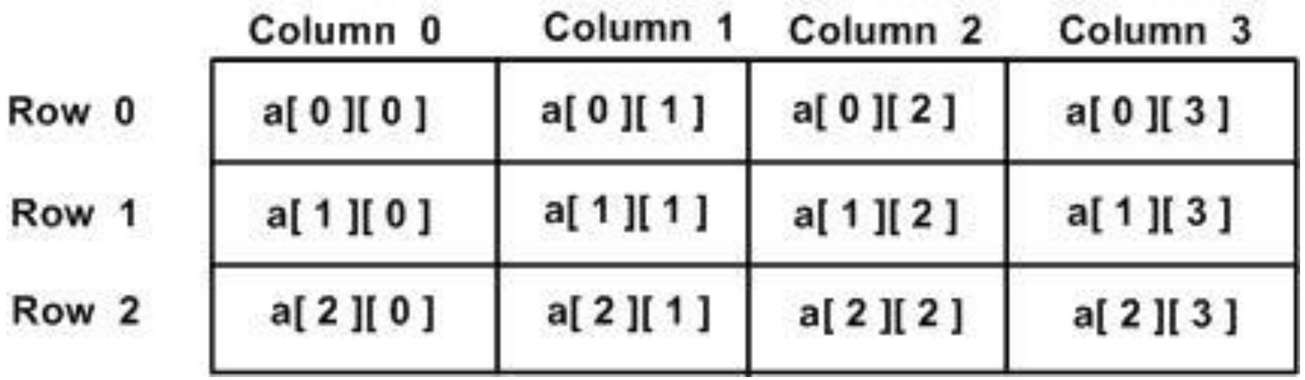
# Class 21

2D Arrays

# 1D arrays v 2D arrays

| | 1D example | 2D example |
|---|---|---|
| Declaration | int nums[5]; | int grades[2][3]; |
| Initialization | int nums[5] = {1, 2, 3, 4, 5}; | int grades[2][3] = {{1, 2, 3}, {4, 5, 6}}; |
| Assign value to specific element | nums[2] = 15; // stores 15 in nums at index 2 | grades[0][2] = 10; // stores 10 at row index 0, column index 2 |
| Reference a specific element | cout << nums[1]; | cout << grades[1][1]; |
| Process array | Single for loop with counter initialized to 0 | Nested for loops, with row and column counters initialized to 0 |

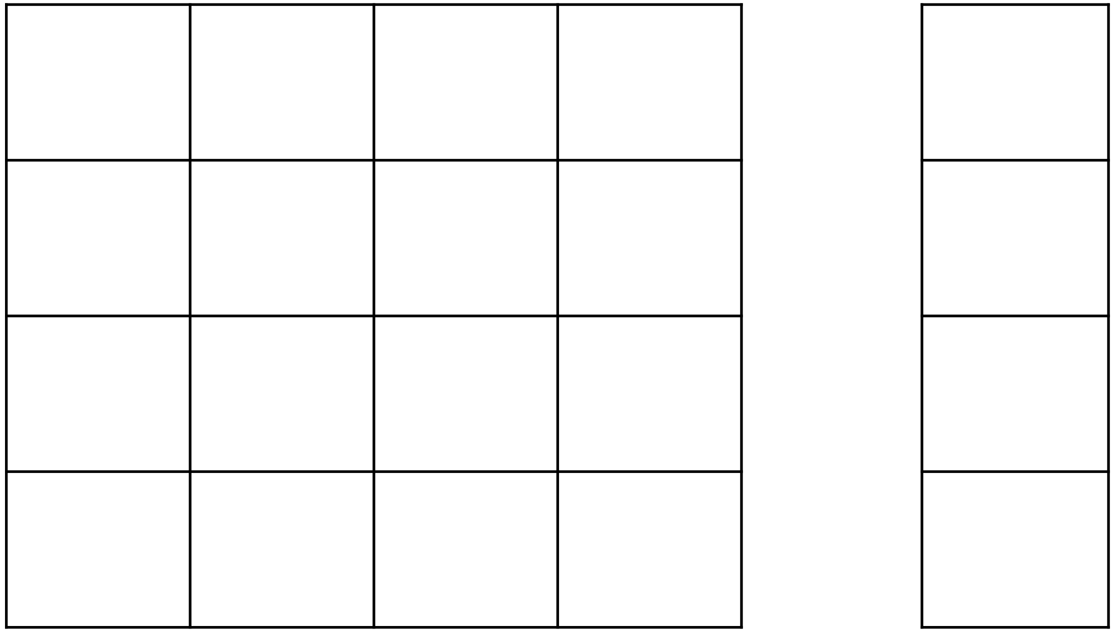
# Accessing elements in 2D array



| | Column 0 | Column 1 | Column 2 | Column 3 |
|---|---|---|---|---|
| Row 0 | a[ 0 ][ 0 ] | a[ 0 ][ 1 ] | a[ 0 ][ 2 ] | a[ 0 ][ 3 ] |
| Row 1 | a[ 1 ][ 0 ] | a[ 1 ][ 1 ] | a[ 1 ][ 2 ] | a[ 1 ][ 3 ] |
| Row 2 | a[ 2 ][ 0 ] | a[ 2 ][ 1 ] | a[ 2 ][ 2 ] | a[ 2 ][ 3 ] |

```
// read in values from user
int arr[3][4];
for(int row = 0; row < 3; row++){
   for(int col = 0; col < 4; col++){
      cin >> arr[row][col];
   }
}
```

# 2D arrays summarized in 1D arrays



```
int a[4][4];
// initialize a to random numbers
for(int row = 0; row < 4; row++){
   for(int col = 0; col < 4; col++){
      a[row][col] = rand()%10 + 1;
   }
}
int b[4] = {0};
// compute sum of each row of a and store in b
for(int row = 0; row < 4; row++){
   for(int col = 0; col < 4; col++){
      b[row]  += a[row][col];
   }
}
```

# Example 1

Goal: Fill a 3 x 3 integer array with numbers from the user. Compute average of all numbers in array. Print all numbers greater than average.

Plan:

- Declare 3 x 3 integer array
- Declare sum int variable and initialize to 0
- Declare average double variable and initialize to 0
- Using nested loops, read in values from user
- Using nested loops, compute sum of all values entered by user
- Divide sum by count of values in array and store result in average variable
- Using nested loops, print all array values greater than average

# Example 2

Goal: Fill a 5 x 5 array with random two-digit integers. Print the array. Count the number of odd numbers in each row. Print count of odd numbers in each row, and print all odd numbers in each row.

Plan:

- Declare 5 x 5 integer array
- Declare 1D int array with capacity 5
- Initialize all elements of 1D array to 0
- Using nested loops, fill 5 x 5 array with random numbers.
- Using nested loops, print the array in table format (rows and columns)
- Using nested loops, count the number of odd integers in row r and store the count in 1D array at index r
- Using nested loops, print the odd numbers in each row to the monitor

# Example 3

Goal: Fill a 5 x 5 array with random integers between 1 and 100 (inclusive). Print array. Compute the average of each row. Print all numbers in each row that are greater than the row average.

Plan:

- Declare 5 x 5 integer array
- Declare 1D double array with capacity 5
- Declare rowSum variable
- Set rowSum = 0 at start of each outer loop
- Using nested loops, print the array
- Using nested loops, sum the values in each row
- At conclusion of each inner loop, divide rowSum by number of elements in row and store result in 1D array at index r
- Using nested loops, print the numbers in row r that are greater than the average stored in the 1D array at index r