





# Creating a 2-d array

- Declare a reference x for a 2-d integer array
- 2. Create a 2-by-3 integer array y
- 3. Create the following array:
  - 246
  - 813

May 2, 2006

#### Accessing a 2-d array

Given a reference **x** that points to a 2-d **int** array. . .

- 1. What is its height (# of rows)?
- 2. What is x[0]?

May 2, 2006

- 3. What is the length of the first row?
- 4. How to access last element in the second row?

Lecture 27

5. How to access last element in last row?

## Example 1

May 2 200/

Given a 2-d integer array  $\mathbf{x}$ , calculate the sum of all entries in the array. Assume the array is rectangular.

### What if . . .

May 2 2006

- The array is ragged instead of rectangular? Suppose all rows exist but the rows have different lengths.
- Not all rows exist and the existing rows have different lengths?

Lecture 2





# Example 3

May 2, 2006

Given a 2-d array m, re-order the columns such that the column with the highest column sum is the first column. Assume m is rectangular.

How will the code differ from the previous question?

```
public class Array2d {
public static void main(String[] args) {
  int[][] m= new int[][]{new int[]{1,3,5,9},
                         new int[]{2,100},
                          new int[]{2,2,3} };
  int[] rsum= new int[m.length]; //array of row sums
  for (int r=0; r<m.length; r++) {
    //calculate row sum
    //if current row sum is highest, update highSum, highIndex
  }
  //swap row with highest sum into first row
  //Print 2-d array
  for (int r=0; r<m.length; r++){</pre>
    for (int c=0; c<m[r].length; c++)</pre>
      System.out.print(m[r][c]+ " ");
    System.out.println();
  }
}
```

}