PER SCHOLAS

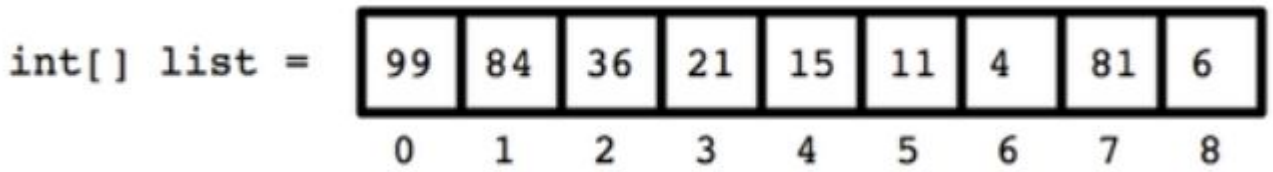
Introduction to Java Programming

Unit 3 - What is Java? Start a Java project in Eclipse IDE, Getting familiar with Java features

2D Arrays

Two Dimensional Arrays

- So far we have studied how to store linear collections of data using a single dimensional array.



- However, the data associated with certain systems (a digital image, a board game, etc.) lives in two dimensions.
- To represent this data in our programs, we need a multi-dimensional data structure, that is, a multidimensional array.

2D Arrays

Two Dimensional Arrays

	Chicago	Boston	New York
Chicago	0	983	787
Boston	983	0	214
New York	787	214	0

```
int[][] distances = {
                      {0, 983, 787},
                      {983, 0, 214},
                      {787, 214, 0}
                    };
```

2D Arrays

Declaring Two Dimensional Arrays

- You can declare a two dimensional array using almost the same syntax you would use to declare a single dimensional array. For example:

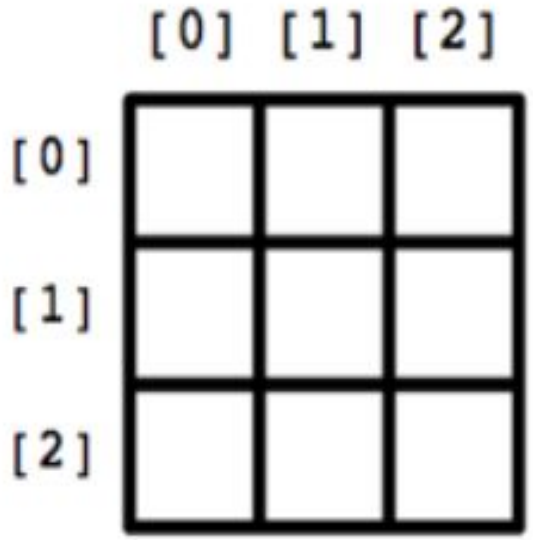
```
7  public class ArraysBuffers{
8      public static void main(String[] args){
9          int[][] myList = new int[5][5];
10     }
11 }
```

- This would create a 5 x 5 matrix of integers, all defaulted to a zero value upon creation. We generally think of the first number as the number of “rows” in your array and the second as the number of “columns”

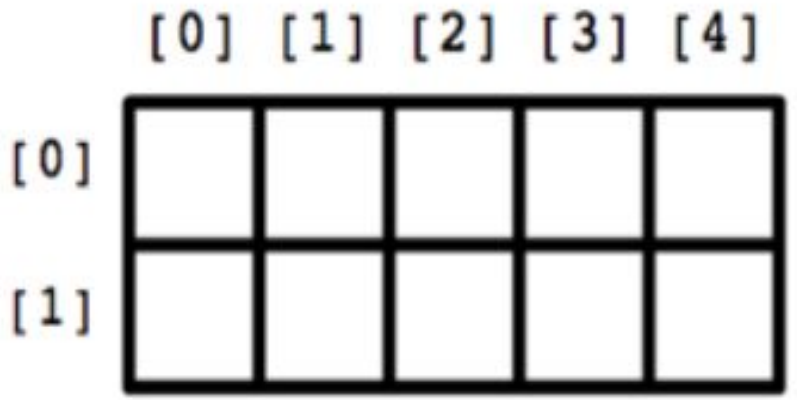
2D Arrays

Declaring Two Dimensional Arrays

```
int[][] a = new int[3][3]
```




```
int[][] a = new int[2][5]
```



2D Arrays

Declaring Two Dimensional Arrays

- You can also create a two dimensional array with pre-specified values by using almost the same syntax as a single dimensional array. For example:

```
 7 public class ArraysBuffers{
 8     public static void main(String[] args){
 9         int[][] list = {
10                 { 1, 2, 3 },
11                 { 4, 5, 6 },
12                 { 7, 8, 9 }
13                 };
14     }
15 }
```

2D Arrays

Accessing Two Dimensional Arrays

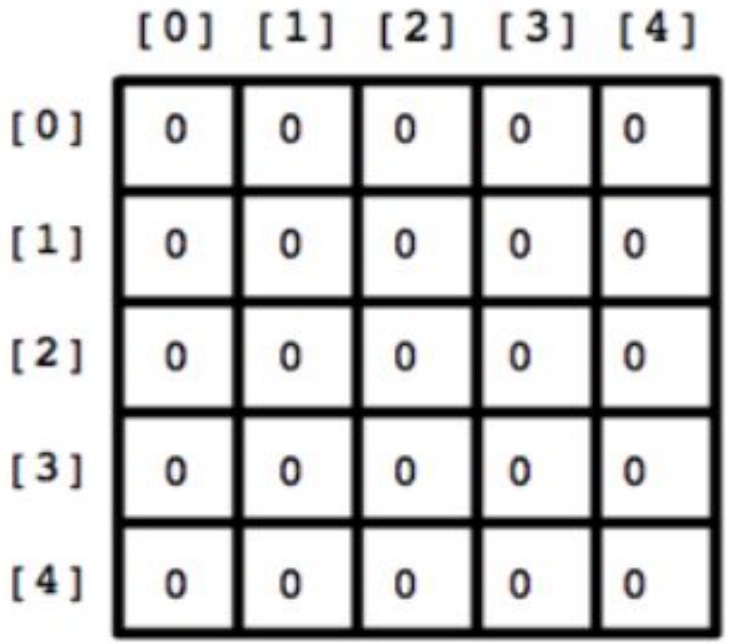
- You can access the two dimensional array values using the same syntax as you would with a single dimensional array. For example, the first element in the first row is at position:
  - myList[0][0]

```
int[][] list = new int[5][5];

list[0][0] = 50;

list[4][4] = 99;

list[2][1] = 11;
```



2D Arrays

Accessing Two Dimensional Arrays

```
int[][] list = new int[5][5];

list[0][0] = 50;

list[4][4] = 99;

list[2][1] = 11;
```

	[0]	[1]	[2]	[3]	[4]
[0]	50	0	0	0	0
[1]	0	0	0	0	0
[2]	0	0	0	0	0
[3]	0	0	0	0	0
[4]	0	0	0	0	0

2D Arrays

Accessing Two Dimensional Arrays

```
int[][] list = new int[5][5];

list[0][0] = 50;

list[4][4] = 99;

list[2][1] = 11;
```

	[0]	[1]	[2]	[3]	[4]
[0]	50	0	0	0	0
[1]	0	0	0	0	0
[2]	0	0	0	0	0
[3]	0	0	0	0	0
[4]	0	0	0	0	99

2D Arrays

Accessing Two Dimensional Arrays

```
int[][] list = new int[5][5];

list[0][0] = 50;

list[4][4] = 99;

list[2][1] = 11;
```

	[0]	[1]	[2]	[3]	[4]
[0]	50	0	0	0	0
[1]	0	0	0	0	0
[2]	0	11	0	0	0
[3]	0	0	0	0	0
[4]	0	0	0	0	99

2D Arrays

Accessing Two Dimensional Arrays

- Two dimensional arrays are really just one dimensional arrays that have been “chained” together. For example:
  - **int[][] list = new int[5][5];**
- Is a list of 5 elements. Each of those elements, however, references another single dimensional array, each of which is 5 elements long as well.
- Moreover, a two-dimensional array is really nothing more than an array of arrays.

2D Arrays

Accessing Two Dimensional Arrays

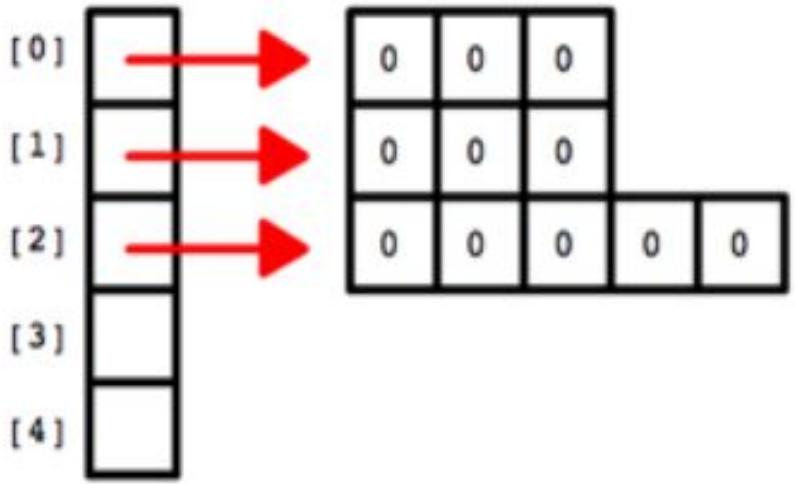
- A 2D array can be initialized with a linear size such as: **int[][] list = new int[5][];**
- Then the arrays inside can be initialize separately and can be of different sizes

```
int[][] list = new int[5][];

list[0] = new int[3];

list[1] = new int[3];

list[2] = new int[5];
```



2D Arrays

Getting the Dimension Lengths

- The length of your “main” array in a 2 dimensional array can be obtained by referencing the following. We usually refer to this as the number of “rows” in the array
  - **list.length**
- The length of each sub-array in a two dimensional array can be obtained by referencing the following. We usually refer to this number as the “columns” in the array.
  - **list[element].length**
- Note that each row in a two dimensional array could have a different number of columns. We sometimes refer to these kinds of arrays as “ragged” arrays

2D Arrays

Getting the Dimension Lengths

```
 7 public class ArraysBuffers{
 8     public static void main(String[] args){
 9         int[][] list = new int[5][];
10         list[0] = new int[3];
11         list[1] = new int[2];
12         list[2] = new int[5];
13         list[3] = new int[7];
14         list[4] = new int[4];
15         System.out.println("Length of Array: "+list.length);
16         System.out.println("Length of the element 3 Array "+list[3].length);
17     }
18 }
```

2D Arrays

Looping Through 2D Arrays

- In order to iterate over all elements in a two dimensional array you will need to maintain two indexes – one for the row and one for the column
- This is usually done by setting up a nested for loop like this:

```
 7 public class ArraysBuffers
 8 {
 9     public static void main(String[] args)
10     {
11         int[][] list = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
12         for (int row = 0; row < list.length; row++)
13         {
14             for (int col = 0; col < list[row].length; col++)
15             {
16                 System.out.print(list[row][col]);
17             }
18             System.out.println();
19         }
20     }
21 }
```

2D Arrays

Looping Through 2D Arrays

- What the values of list after the following program executes?
  - Use the debugger in order to see exactly how this program executes

```
 7 public class ArraysBuffers
 8 {
 9     public static void main(String[] args)
10     {
11         int[][] list = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
12         for (int row = 0; row < list.length; row++)
13         {
14             for (int col = 0; col < list[row].length; col++)
15             {
16                 list[row][col] = list[row][col] * 2;
17             }
18         }
19     }
20 }
```

2D Arrays

Programming Exercise

- Time for a quick assignment:
  - Complete the BabyBlackJack assignment