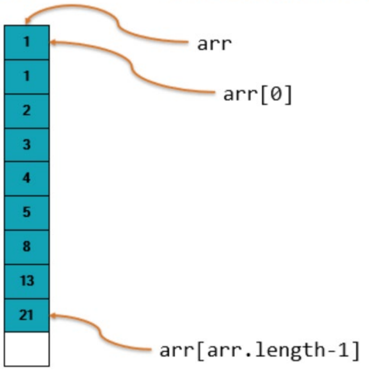
CS163/164 – Worksheet Multidimensional Arrays

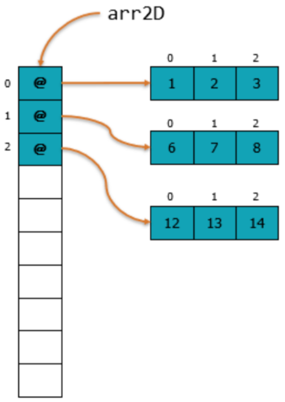
Name:____

Array Review

- Ways to store
  - Variables in order
  - index from 0..N
- Arrays are
  - a type themselves
  - the value of the array
    - reference to memory location!
    - you need an index to access the value!
  - .length gives us total memory allocated
- Arrays can
  - be any size – as long as you allocate it
  - Store any valid type: primitives and objects
  - and store other arrays (since they are a type themselves)!

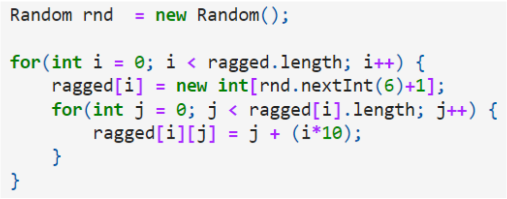
```
int[] arr = {1,1,2,3,5,8,13,21};
```





Multidimensional Arrays

- Number of brackets define number of dimensions
- Declaring the entire array at once
  - int [][] arr2D = new int[3][3];
    - 3 rows, 3 columns
    - fully initialized with zeros
- Irregular/ragged multidimensional array, each column can have a different length
  - int[][] ragged = new int[3][];
    - creates an array of 3 null values
    - requires an int[] to be placed in each spot.



```
Random rnd  = new Random();

for(int i = 0; i < ragged.length; i++) {
    ragged[i] = new int[rnd.nextInt(6)+1];
    for(int j = 0; j < ragged[i].length; j++) {
        ragged[i][j] = j + (i*10);
    }
}
```

CS163/164 – Worksheet Multidimensional Arrays

1. Implement a class named DArrays that has the following methods:

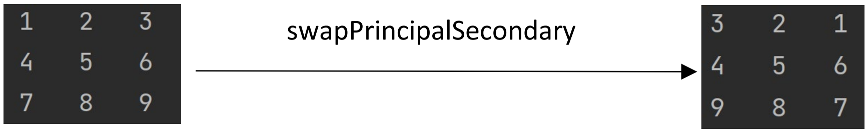
```
/**
 * readMatrix
 * Receives a Scanner in, int row and int col
 * Creates a matrix of row and col dimensions
 * Reads values for each element of the matrix and return the matrix
 * @param in
 * @param row
 * @param col
 * @return int [][]
 */

/**
 * printMatrix
 * Print each element of the matrix using %-4d as formatting
 * pattern to print each element.
 * Tip: use System.out.printf
 * @param matrix [][]
 */

/**
 * readMatrix2
 * Receives a Scanner in and an int row
 * Creates an irregular multidimensional array
 * Reads the number of columns for each row
 * Creates the array of columns for each row considering the
 * number read
 * Reads values for each element of the matrix and return the
 * @param in
 * @param row
 * @return int [][]
 */

/**
 * main
 * Creates a Scanner object
 * Call the methods previously implemented
 * @param args
 */
```

2. Implement the method public static int [] maxEachRow(int [][] matrix) which returns the maximum value in each row of the matrix.

3. Implement the method public static int maxMatrix(int [][] matrix) which returns the maximum value of the matrix.

4. Implement the method public static int[] principalDiagonal(int [][] matrix) which return the values in the principal/main diagonal of a square matrix.

5. Implement the method public static void swapPrincipalSecondary(int [][] matrix) which swaps the values of the principal and secondary diagonals of the matrix.



$$\begin{matrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{matrix} \xrightarrow{\text{swapPrincipalSecondary}} \begin{matrix}3 & 2 & 1\\ 4 & 5 & 6\\ 9 & 8 & 7\end{matrix}$$