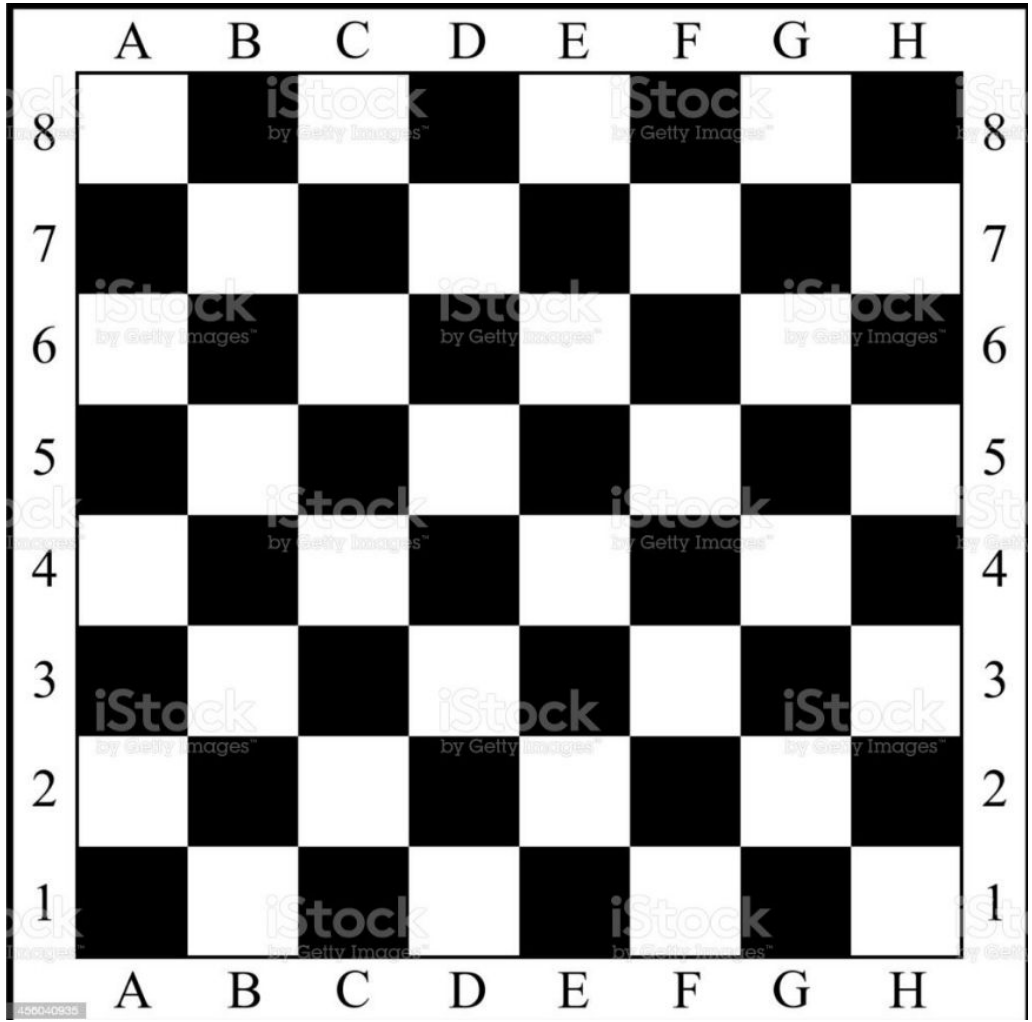
Lecture 14

2D Arrays



How to represent a chess-board using Arrays?

-

2D Array Declaration and Initialization

```
int intArray[][] = {
                    { 1,  2,  3,  4  },
                    { 5,  6,  7,  8  },
                    { 9,  10, 11, 12 },
                    { 13, 14, 15, 16 },
                    { 17, 18, 19, 20 },
                  };
```

- `<data-type>[][] <array-name> = new <data-type>[# of rows][# of columns];`
- `int[][] chessBoard = new int[8][8];`

Initialize a 2D-Array using for-loop

```
public static void main(String[] args) {
    // Take the number of rows and columns from the input
    Scanner input =new Scanner(System.in);
    int rows = input.nextInt();
    int cols = input.nextInt();

    // Declare Array
    int[][] numbers =new int[rows][cols];

    // Initialize Array
    for(int i = 0; i < rows; i++) {
        for(int j = 0; j < cols; j++) {
            numbers[i][j] = input.nextInt();
        }
    }
}
```

Initialize a 2D Array using for-loop

```
// Initialize Array
for(int i = 0; i < rows; i++) {
    for(int j = 0; j < cols; j++) {
        numbers[i][j] = i+j;
    }
}
```

Access Elements of 2D Array

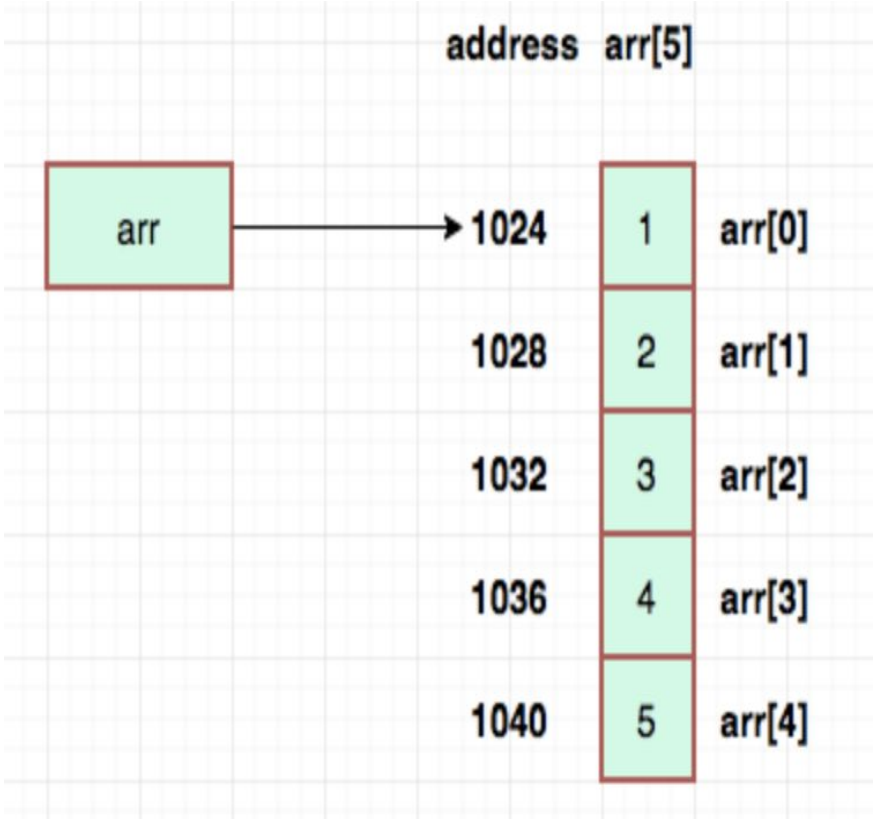
```
for(int i = 0; i < numbers.length; i++) {
    for(int j = 0; j < numbers[0].length; j++) {
        System.out.print(numbers[i][j] + ",");
    }
    System.out.println();
}
```

numbers.length = length of rows

numbers[0].length = length of columns

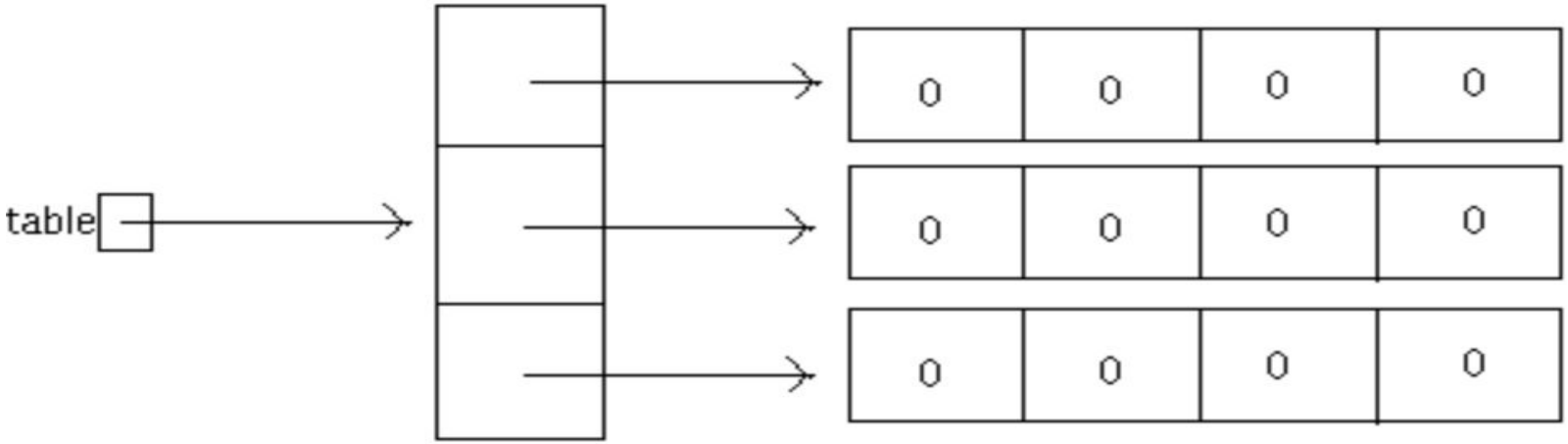
Arrays and Memory

- int[] studentMarks = new int[7];
  - Allocates contiguous memory of 7 Integers and the starting address is stored in the variable studentMarks.
  - studentMarks is a reference variable
  - Arrays are non-primitive data-types
  - int[] arr = {1,2,3,4,5};



2D Arrays and Memory

Memory diagram for `int[][] table = new int[3][4];`



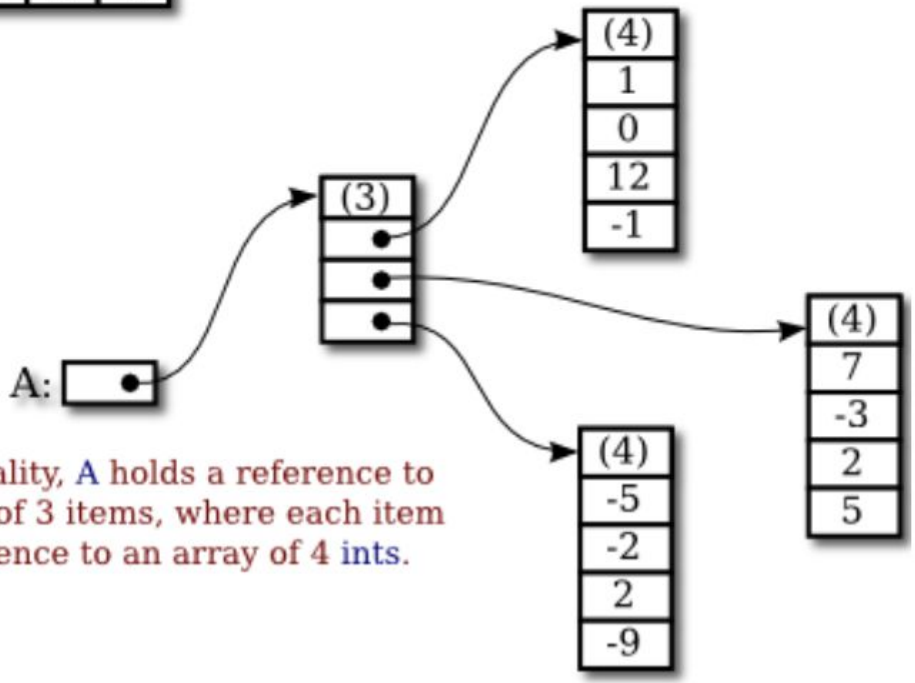
array whose elements are references to objects of type `int[]`

Each row is held in an array whose elements are of type int.

If you create an array A = new int[3][4], you should think of it as a "matrix" with 3 rows and 4 columns.

A:

1	0	12	-1
7	-3	2	5
-5	-2	2	-9



But in reality, A holds a reference to an array of 3 items, where each item is a reference to an array of 4 ints.

Find The Maximum Element in an Array

```
// {4,7,1,2,17,21,5} then return 21
// {1,2,8,3,6,9} then return 9
// Array can't have negative numbers
// Array can't have 0 also
public static int findMaximumNumber(int[] arr) {
    // 1 unit space for maximumNumber variable
    int maximumNumber = 0; // This is executed 1 time

    // 1 unit space for variable i
    for(int i = 0; i < arr.length; i++) { //
        if(arr[i] > maximumNumber) { // This is executed N times where N is the length of the input array
            maximumNumber = arr[i];
        }
    }

    return maximumNumber; // This is executed 1 time
}

// Time Complexity = N + 1 + 1 == O(N) + O(1) + O(1) == O(N) Linear
// Space Complexity = 1 + 1 = O(2) == O(1) Constant
```

Find the Maximum Element in a 2D Array

Given an array check if it’s sorted in ascending order