Topic – Arrays and 2D Arrays

- Arrays
  - Data structure which stores a fixed-size sequential collection of elements of the same type.
- ArrayList vs. Arrays
  - Java arrays are of a fixed length. After arrays are created, they cannot grow or shrink, which means that you must know in advance how many elements an array will hold.
  - Array lists are created with an initial size. When this size is exceeded, the collection is automatically enlarged. When objects are removed, the array may be shrunk.
  - In Java, there are a variety of ways of creating a new array. One way is to first create the array and then fill it:

```java
Die[] dice = new Die[5]; /*Declaring an array variable dice, creating an array, and assigning the reference of the array to Die */

//Die dice[] = new Die[5] also works but not preferred way

for(int i = 0; i < 5; i++) {
    System.out.print(dice[i] + " ");
}
```

- We now have Dice[0], Dice[1], Dice[2], Dice[3], Dice[4] (Note: index starts with 0)
- Notice that the array was created but the slots were not filled with Die objects. Instead, it is full of the keyword null. We need to create 5 new Die objects and add them to the array.

```java
for(int i = 0; i < 5; i++) {
    dice[i] = new Die();
}
```

- Visual Example: double[] myList = new double[10];

- Note that you have to specify the array size when you create it. Java arrays are fixed size and cannot be increased or decreased!
- Arrays can have size up to 2 billion (more precisely, \(2^{31} - 1\)).
To save space, the Java language creators added a second way to create an array, for when you know ahead of time all the values to be placed in the array initially:

```java
int[] values = new int[]{3,1,4,1,5,9,2,6,5,3,5,8,9,7,9};
int[] values = {3,1,4,1,5,9,2,6,5,3,5,8,9,7,9};
```

Let's now use arrays

```java
public static void main(String[] args) {
    Die[] dice = new Die[5];
    for(int i = 0; i < 5; i++)
        dice[i] = new Die();

    for(Die d : dice) //foreach loop
        d.roll();
    for(int i = 0; i < 5; i++)
        System.out.println(dice[i].getFaces());
}
```

What do you do if you created an array of size 10 and then, in the middle of the program, an 11th item needs to be added to the array? Use ArrayList

You can also read:
https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html

2D Arrays

What if you need bigger arrays than 2 billion? One way is to have arrays of arrays:

```java
int[][] A = new int[3][4]; //3 rows and 4 columns
```

This creates an array with 3 slots, each of which is an array of length 4 [draw a picture with pointers]

```java
int[][] A = { { 1, 0, 12, -1 },
            { 7, -3, 2, 5 },
            { -5, -2, 2, -9 } ,
        };
```
- Each element in a multi-dimensional array is itself an array.
- Let's play with this matrix. What do the following lines of code do?

```java
int[][] matrix = new int[5][4];
matrix[0][1] = 3;
for (int j = 0; j < 4; j++)
    matrix[0][j] = 1;

// nested arrays
for (int i = 0; i < matrix.length; i++)
    for (int j = 0; j < matrix[0].length; j++)
        matrix[i][j] = i + j;
```

- We don't need all inner arrays to be the same length:
  ```java
  int[][] m = new int[2][];
m[0] = new int[]{1, 2, 3};
m[1] = new int[]{1, 2};
  ```

Disclaimer: Notes adapted from previous CS 231 lecture materials at Colby College.