# CIS351-ArrayPlay Lab

# Submission Instructions

1. Submit the completed **ArrayPlay.java** file in Blackboard.
2. DO NOT forget to mention your full name in the Java class documentation.

# Objectives

*Students will be able to:*

- Create an array
- Initialize the array elements to an appropriate value
- Use an array to track a sequence of values
- Display the contents of the array

# Key Terms

***array.length***

an attribute of the array that is the number of elements in the array

***initializer list***

a list of values used to instantiate and initialize an array to the values of the items in the list

***ArrayIndexOutOfBoundsException***

an exception that is raised when a program tries to access an element using a subscript that either is negative or is equal to or larger than the number of elements in the array

# Download Materials

- Download **Die.java** which you will need for this lab.
- Download **Die.html** to explore the documentation for this class.

- Download the **cover sheet** which needs to be submitted with the lab.

## Additional Help

Watch the posted video on how Array works in Java.

## Part 1 - Getting Started

You will write this program from scratch.

1. Create a new folder for this lab. Download a copy of Die.java to that folder.

2. Create a new Java file named **ArrayPlay.java**.

3. ArrayPlay will contain a single main method. For each section below, you will add code to the program, not replace prior code. When you are done, this program will provide you with a reference for working with arrays including initializing, printing, and manipulating entries.

## Part 2 - Sequentially accessing an array

NOTE: Wherever you are asked to do something to your array and then print it, you should finish the update step and have a separate loop to perform the printing.

ALSO NOTE: Keep all steps in the same method.

- In the main method, declare and initialize a single array of integers that is 6 elements long.
- Initialize each of the array elements to the value, -1. (use an appropriate loop to do so).
- In another loop, print each element of the array on a separate line as shown:

```
array[0] = -1
array[1] = -1
...
```

- TEST IT!
- Add a loop to change the value of each element of the array to its subscript. For example, myArray[3] should hold the value 3.

- In another loop, print each element of the array in step 5 on a separate line. (You may copy and paste the code from step 3.)
- If you have not seen an ArrayIndexOutOfBoundsException, create one: change your loop so that it reads past the end of the array. Run it. What happens?
- Fix the error from the previous step. Add code to reinitialize your array to all zeros.

## Part 3 - Randomly accessing an array

- Create a new `Die` object.
- Create a loop that will iterate 100 times. In the loop body:
  - Roll the die.

  - Take the result of the roll, map it to the location in the array, and increment that cell of the array. (The die values will be in the range 1 - 6; your array subscripts are 0 - 5. Think about how you will map the die values to subscripts).

    The value of each array element will be a count of the number of times that roll is made. So if 1 comes up three times, array[0] should have the value 3.

- After the loop finishes, print the results of the rolls as they correspond to die faces. In other words, how many times was a 1 rolled?, 2? etc.

  For example, if the array holds {20, 17, 19, 15, 12, 17} you would output:

  ```
  1 was rolled 20 times.
  2 was rolled 17 times.
  ...
  ```

## Part 4 - Working with two arrays

- You will need two double arrays. Declare these and then....
  i. Use an initializer list to initialize **one** of the arrays to 10 double values (of your choice). See the book regarding initializer lists.
  ii. Instantiate the other array to 10 elements, but do not initialize the contents.
  iii. Print the contents of both arrays. Print them side by side on your screen. Label this output,Before.
  iv. Copy the contents of the first array to the second array and again print their contents

side by side. Label this output After.

v. Finally, change the contents of one element of the first array and a different element of the second array. (assign a different number than before).
vi. Again, print the contents of the two arrays side by side. Label this output *After Change*.
vii. Check that the contents of the two arrays are indeed different. If not, try to figure out why they are not and make the appropriate corrections.

## Grading Criteria

**Total points: 10 points**

General Programming Correctness - 1pt
Part 2 - 3pt
Part 3 - 3pt
Part 4 - 3pt

*Farzana Rahman / frahman@syr.edu*