	Computer Science Department/College of Engineering and Computer Science **CSc 20: Programming Concepts and Methodology II** **Lab 4– Java Array (Spring 2018)**

Objective:

In this lab, you are to familiar with Java's array functionalities including:

1. To declare an array, initialize an array, and refer to individual elements of an array.
2. To manipulate a multidimensional array.
3. To output a content of a multidimensional array.

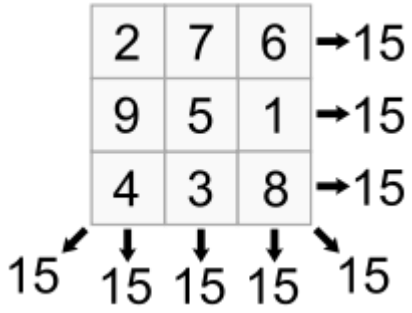
Preparation: (at home)

Write a variable declaration and for loop necessary to create and initialize an integer array named squares that contains the following values:
0 1 4 9 16 25 36 49 64 81 100

Test your work with the following website:
https://www.codestepbystep.com/problem/view/java/arrays/squaredArray

Lab work (in school laboratory):

This lab's objective is to exercise with usages of Java's array features. We would like to apply this learning to determine if a two-dimensional array is a magic square. A magic square is a square matrix in which the sum of every row, every column, and both diagonals is the same. An example below is a magic square:



(source: https://en.wikipedia.org/wiki/Magic_square)

Magic squares have been studied for many years, and there are some particularly famous magic squares. In this exercise you will write code to determine whether a square is magic.

Activities:

1. Copy instructor's MagicSquare.java and TestMagicSquare.java into your working directory. File MagicSquare.java contains the shell for a class that represents a square matrix. It contains headers for a constructor that gives the size of the square and methods to read values into the square, print the square, find the sum of a given row, find the sum of a given column, find the sum of the main (or other) diagonal, and determine whether the square is magic. The read method is given for you; you will need to write the others. Note that the read method takes a Scanner object as a parameter. File MagicSquareTest.java contains the shell for a program that reads input for squares from a file named magicText.txt and tells whether each square is a magic square. Following the comments, fill in the remaining code. Note that the main method reads the size of a square, then after constructing the square of that size, it calls the readSquare method to read the square in. The readSquare method must be sent the Scanner object as a parameter.

The data in magicText.txt is also provided in Canvas.

2. Develop your program according the given pseudo code.

Additional Requirements:

1. Your output should be closely similar to the output of the instructor's sample program (see below).

Sample output:

```
 ----jGRASP exec: java MagicSquareTest

***** Square 1 *****
8 1 6
3 5 7
4 9 2
***** Square 1 *****

Sum of row 0 is: 15
Sum of row 1 is: 15
Sum of row 2 is: 15

Sum of column 0 is: 15
Sum of column 1 is: 15
Sum of column 2 is: 15

The sum of the main diagonal is: 15

The sum of the other diagonal is: 15

Is it a magic square: true
```

Deliverables:

(1) Demonstrate your preparation work from the website https://www.codestepbystep.com/problem/view/java/arrays/squaredArray to your instructor.

(2) Demostrate your running programs, MagicSquareTest.java and MagicSquare.java, to your instructor.

(3) Turn in your programs (program source text) along with the output result in PDF form to to Canvas for grading.