# Lab Assignment 7\_1 - Arrays

## Part 1:

1. For this program, you will be adding methods and statements into existing code to create a program that uses overloaded methods and constructors.
2. Create a new Java project called **Lab7\_1A** and a class named **Lab7\_1A**
3. Create a second class called **ArrayProcessor.**
4. Open the files **Lab7\_1A.txt** and **ArrayProcessor.txt** attached to this assignment and replace the code in your classes with their code.
5. Look for all the **//\*To Do** comments in both classes and follow those instructions. Add your name and the program’s purpose at the top of Lab7\_1A.  
     
   **Note:** There are lines in the classes that are just documentation; don’t delete or change them. Simply look for the To Do lines and follow their instructions.

## Part 2:

1. Create a new Java project called **Lab7\_1B** and a class named **Lab7\_1B**.
2. Create a second class called **ArrayExaminer**.
3. In the ArrayExaminer class declare the following instance variables:
   1. String named textFileName
   2. Array of 20 integers named numArray (Only do the 1st half of the declaration here:   
      int [] numArray; )
   3. Integer variable named largest
   4. Integer value named largestIndex
4. Add the following methods to this class:
   1. A constructor with one String parameter that
      1. does the second half of declaring numArray

(numArray = new int [20];).

* + 1. Sets textFileName equal to the parameter
    2. Sets the other instance variables to 0.
  1. A void method named **fillArray** that will read 20 integers from a text file (textFileName) and fill the array with the values.  
     It should also print the array all on one line. (Hint – use the Arrays.toString)
  2. void method named **findLargest** (no parameters) that reads through numArray and sets the instance variable, **largest**, equal to the largest item in the array, and sets **largestIndex** equal to its index  
     Print the value of **largest** (with a label).
  3. boolean method named **goingUp** (no parameters) that
     1. returns a true if all the values in numArray from the beginning up to the **largestIndex** position are increasing, and false otherwise. (Example of increasing [1,2,6,9,23] )
  4. boolean method named **goingDown** (no parameters) that
     1. returns a true if all the values in numArray from **largestIndex** position until the end are decreasing, and false otherwise. (Example of decreasing [31,12,6,2,1])
  5. void method named **isPeak** (no parameters) that
     1. calls findLargest
     2. if goingUp and goingDown are both true, then print that the array has a peak. Otherwise print that it does not have a peak.

1. Back in the main method/class.
   1. Declare and instantiate an ArrayExaminer object named array1 and send “Lab7\_1B1.txt”) as the parameter to its constructor.
   2. Call fillArray for array1
   3. Call isPeak for array1
   4. Declare and instantiate an ArrayExaminer object named array2 and send “Lab7\_1B2.txt”) as the parameter to its constructor.
   5. Call fillArray for array2
   6. Call isPeak for array2