# Programming Project 3 (Due 11-1-2016 at 2:00 PM)

Write a JAVA program to solve the following problem. Your program should properly compile and run. Your code MUST follow the documentation style used in your textbook. You need to upload into Moodle the following:

- From this point on, each of your projects is expected to be composed of multiple source code files. Thus to organize these files, all of them have to be part of a single package. Your package name MUST match the directory containing your code files. Finally, you need to export your package as a .JAR file then upload this single file into Moodle. If you are not familiar with how to export your code into a .JAR file, please check the following link. **Make sure that you include your source code files (check the appropriate check box) while exporting the .jar file**.
  http://help.eclipse.org/neon/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2Ftasks%2Ftasks-33.htm

**Very Important Note: Do NOT upload any other files into Moodle; just ONE file with the extension ".jar" that is it.**

**Problem Statement:**
The following method does not appear to be working properly if all data are negative numbers. You are asked to write a test driver class that will test this method in order to identify the issue with the code. The driver class will first read a data (from an input file called "**inputData.txt**") into an integer array then call the method with different input parameters to test its functionality.

```java
/** Finds the target value in array elements x[start] through x[last].
    @param x array whose largest value is found
    @param start first subscript in range
    @param last last subscript in range
    @return the largest value of x[start] through x[last]
    @pre first <= last
 */
public int findMax(int[] x, int start, int last) {
    if (start > last)
        throw new IllegalArgumentException("Empty range");
    int maxSoFar = 0;
    for (int i = start; i < last; i++) {
        if (x[i] > maxSoFar)
            maxSoFar = i;
    }
    return maxSoFar;
}
```

Please do the following in this order:

1. In your code, you also need to include a brief description of your debugging plan as a series of comments that proceeds the method definition.
2. In order to properly debug the given method, you need to make sure that you have several print statements at various execution phases to ensure a better insight of how the given code work. After finishing the debugging the method and fixing the problem, **you need to leave ALL debugging statements in, but execute them only when the global constant TESTING is true**.
3. Before uploading your code. Make sure that it works properly for ANY combinations of input parameters and for all boundary conditions.

**(End of Project)**