PROJECT #2: 96 points – Due Sunday, September 16, by 11:59 p.m.
CSE 274 – Fall 2018

Outcomes:

- Implement an ADT (specifically a resizable array-based implementation of the Set ADT)
- Test a class using the JUnit library

Naming requirements (not following any of these may result in a score of 0):

- The Eclipse project name must be **Project2.**
- You will have write exactly two source code files: **ResizableArraySet.java** and **SetTester.java**
- You will also need to include the Set interface as specified in **SetInterface.java**. Do not modify this file. Your resizable array set must implement this interface.
- You will use the **default package** (this means there should be no package statements in any of your files).

Your assignment is to:

1. In a class named **ResizableArraySet**, implement the abstract data type Set by implementing all of the methods in the interface found in **SetInterface.java**. Your implementation will use a resizable array of int values that doubles in size any time an item is added when the array is already full. Do not use Java's ArrayList. You should simply work with an array, and handle the array resizing on your own.
2. The only instance variables should be the array of int values and a counter for the number of entries in the set.
3. There should be two constructors:
    a. A constructor with no parameters. By default, this should use an array instance variable with a starting size of 10.
    b. A constructor with an int parameter specifying the starting size of the array instance variable.

4. SetTester should thoroughly test the methods and constructors in the ResizableArraySet class. It should utilize the JUnit library to test the ResizableArraySet class. You should comment your testing code on what is being tested, what results are expected, and the actual results. **This should not involve any interaction from the user.** Just run test cases. Comments in your test methods should look something like this:

    // Creating an empty set and adding three items 4 9 2
    // Expecting to see 4 9 2
    // Returns 4 9 2

    // Removing 4 and expecting to see 9 2
    // Returns 9 2

Your test code should not require me to look at your source code. I should know, by running your tester, what is being tested, what results are expected, and what the actual results are. I will be looking for:

- Thoroughness (test all constructors and methods)
- Organization (keep related tests together)
- Readability (use single blank lines in appropriate places to break your code into "chunks" so it's easy to know when one part of your testing is done and the next part begins). The Arrays.toString() is a useful way to display the contents of an array. Feel free to use it to output your results.

Grading Rubric:

Outcome	**Max score**
Constructors implemented correctly	5
Arrays grow and shrink as specified	10
Intersection, union, difference work as specified	24
Remaining methods work as specified. Add and remove methods use the same algorithms we learned for Bag	24
Tester is thorough, organized, and readable (2 points for each test methods, there are 11 methods) Must use JUnit library, otherwise 0	33
Code formatted according to generally accepted standards	0 (deductions only)
Code follows approaches taught in CSE 174 and 271	0 (deductions only)

Submission:

You will submit Java source code files: **ResizableArraySet.java** and **SetTester.java** on Canvas.