PROJECT #1: 100 points, CSE 274 – Fall 2019
Java Review (Array, Interface, Class, JUnit Test, Driver)

Outcomes:

- Use an array to implement a simple container.
- Write and understand Javadoc documentation.
- Follow a set of specifications for class implementation, including constructors and methods.
- Write a driver class to test a class and its methods.
- Use JUnit library to test code.
- Test code for edge cases, corner cases, and typical cases.
- Format and comment source code that adheres to a given set of formatting guidelines.

General description:

You will implement a simple container (like the one we worked together in the lab). You are required to write a Java class called, ArrayUtility which implements an interface called UtilityInterface. The container (ArrayUtility) stores multiple Integer objects in an array and defines the methods from the UtilityInterface to manipulate the array. The methods that need to be implemented are documented in the file UtilityInterface.html (given).

You are required to write 1 Java interface and 3 Java classes for this project:

1. **UtilityInterface**.java: A Java interface that describes the utility operations on an array of objects. You are required to write this interface following the given java documentation written in UtilityInterface.html. You have to write Javadoc comments for the interface and its methods. You can use/copy the comments written in the UtilityInterface.html file for the interface and methods. It has the following definition:

   ```
   public interface UtilityInterface<T> {

   }
   ```

2. **ArrayUtility**.java: A Java class that stores multiple Integer objects (using an array) and implements the interface UtilityInterface. It has the following definition:

   ```
   public class ArrayUtility implements UtilityInterface<Integer> {
           …
           …
   }
   ```

   It has the following instance variables and constructors:

   a. private Integer [] array: An array to store multiple objects of the Integer class.
   b. private int last: An integer index which always points to the first available location in the array. It starts from 0.
   c. public static final int DEFAULT_CAPACITY: An integer constant which denotes the default capacity of the array. The default capacity is 10.
   d. public ArrayUtility(): The default constructor which initializes the array with the default capacity.
   e. public ArrayUtility(int capacity): A parameterized constructor which initializes the array with the specified capacity. It receives the capacity as a parameter.
   f. Defines all the methods from the UtilityInterface interface.

3. **ArrayUtilityDriver**.java: A Java class that has a main method and creates objects of ArrayUtility class and tests its methods. You have to test all methods thoroughly. Handle edge cases, making sure that your methods behave correctly when (for example), you try to add an element to a full array, or you try to remove an element from an empty array or an element that doesn't exist.
4. **TestArrayUtility**.java: A Java class that uses JUnit library to test methods from the ArrayUtility class. You have to write separate JUnit test methods for each method you test from the ArrayUtility class. You should use JUnit library to thoroughly test your code, otherwise you will get 0 in this section. Again, handle edge cases, making sure that your methods behave correctly when (for example), you try to add an element to a full array, or you try to remove an element from an empty array or an element that doesn't exist.

Prerequisite:

Before beginning your work, you should already have

- Install and setup JDK.
- Install and setup Eclipse.

Submitting:

- You will submit Java source code files: UtilityInterface.java, ArrayUtility.java, TestArrayUtility.java, and ArrayUtilityDriver.java are required, but you may write and include other Java source code files as needed. If ArrayUtility.java depends on other classes you have written, those other source code files must also be submitted.
- All Java files for this assignment should be in the default package. That is, you should not put any package statements in any of the files you submit for this assignment.
- Submit your work before the due date in the appropriate folder on the Canvas site. You should only add your .java files to the submission. Do not add your entire Eclipse project folder. Only add the .java source code files. You can zip the files and submit one zip file if there are multiple java files in your project.
- If you name your file, your class, or your methods incorrectly, or if any of your methods or constructors have the wrong parameter types or return types...anything that breaks my tester, you will lose at least 40% of the value of the assignment, and may end up with a score of zero.

General requirements:

- Use Javadoc style comments for your UtilityInterface and ArrayUtility and any other supporting classes that are meant for reuse. There is no need to use Javadoc style comments for a tester or driver class.
- **At the top of the comments of ArrayUtility.java, include the following (failure to include this will result in a 5-point deduction):**
    - A section listing which parts of your program work correctly
    - A section listing which parts are not working correctly. For those, explain what's wrong.
- Format your code so that it is readable using generally accepted guidelines for formatting source code.

PROJECT #1: 100 points, CSE 274 – Fall 2019
Java Review (Array, Interface, Class, JUnit Test, Driver)

- Follow good programming practices. Don't make code more complicated than it needs to be. Break larger methods into smaller parts. A method should not be longer than 25-30 lines of code.

Requirements for the ArrayUtility class:

- Do not create any other public methods or constructors than those specified here. Do not add any public data members.
- Use the default package for this class (do not add any package statements)

Grading Rubric:

Requirement	**Max score**
Interface: UtilityInterface	
Declares the UtilityInterface interface with Javadoc comment	2 points
Lists all methods specified in the UtilityInterface.html with appropriate Javadoc comments (14 methods). If you've never written Javadoc comments before, here is an 8-minute tutorial created by Norm Krumpe to help you understand: https://www.youtube.com/watch?v=6XoVf4x-tag	8 points
Class: ArrayUtility	
Declares the ArrayUtility class (implements UtilityInterface) and the instance variables (and constant).	2 points
Constructors	4 (2 points each)
Defines the methods from the UtilityInterface correctly (14 methods)	42 (3 points each)
Class: ArrayUtilityDriver : Implements a main method and tests all the methods of ArrayUtility class thoroughly. Handle edge cases, making sure that your methods behave correctly when (for example), you try to add an element to a full array, or you try to remove an element from an empty array or an element that doesn't exist.	14 (1 point each)
Class: TestArrayUtilityDriver Tests all the methods of ArrayUtility class thoroughly using JUnit library. You have to write separate JUnit test methods for each method you test from the ArrayUtility class (14 methods excluding the constructors). Handle edge cases, making sure that your methods behave correctly when (for example), you try to add an element to a full array, or you try to remove an element from an empty array or an element that doesn't exist.	28 (2 points each)

Deductions may be made for programming that is not consistent with general standards of good programming as learned in CSE 174 and 271. This includes methods that are too long, logic that should be cleaned up, poor formatting of code, and so on.