CSCI 230 HW#3

Collaboration policy: **Individual Assignment**
Total Points: **100**

Source Code

In the Dropbox, the only Java file provided in the zip archive is:

- `ArrayList.java`

Furthermore, in this assignment the same `List` interface provided in HW#2 (i.e. the `DoublyLinkedList` assignment) will be used. Under no circumstances are you allowed to modify or create a new `List` interface. You must `List` interface **as is**.

You **may only** modify the `ArrayList` class. In particular, in the `ArrayList` class you **may only** modify the methods listed in **Part 1**, and under no circumstances are you allowed to remove, add, or modify any other line of code in this class (this include instance variables, class variables, constants, etc.).

Lastly, you **may not** change the package structure! Specifically, `edu.cofc.csci230` cannot be removed or modified. If a solution is submitted with a different package structure, it will not be graded, no exceptions.

Part 1

In the `ArrayList` class please fully implement the methods listed below:

- `public void add(int index, AnyType t) throws IndexOutOfBoundsException, NullPointerException`
- `public void set(int index, AnyType t) throws IndexOutOfBoundsException, NullPointerException`
- `public AnyType remove( int index ) throws IndexOutOfBoundsException`
- `public AnyType get(int index) throws IndexOutOfBoundsException`
- `public void clear()`
- `private void grow()`
- `private void shrink()`
- `public void trimToSize()`
- `public void ensureCapacity( int minCapacity )`

In each method above, there is a TODO comment - this is where you add your code. Please note (and testing hint): the functionality of the methods are identical to the ones in the `List` interface and `ArrayList` class defined in the Java API.

Part 2

The provided `ArrayList` class has a main method. In the main please add test cases that demonstrate you have fully evaluated the operational correctness of the methods implemented in **Part 1**. To receive full credit, these test cases **must** be included.

Submission

Create a **zip** file that **only** includes the completed `ArrayList.java` file. **In the zip file please do not include any folders (this includes the package folders) or any other project files**. If you have any questions about the submission policy, you must resolve before the due date. Lastly, please plan appropriately, asking questions the day the assignment is due (within 12 hours) is too late. Please try to resolve any questions at least 2 days before the due date.

The name of the zip file must be your last name. For example, *ritchie.zip* would be correct if the original co-developer of UNIX (Dennis Ritchie) submitted the assignment. Submissions that use RAR compression algorithm (i.e. file that has the .rar extension), or any other compression algorithm other than zip, will not be accepted. Only assignments submitted in the correct format will be accepted (no exceptions).

Please submit the zip file (via OAKS) to the Dropbox setup for this assignment by the due date. You may resubmit the zip file as many times as you like, Dropbox will only keep the newest submission.

Per the syllabus, late assignments will not be accepted – no exceptions. Please do not email Luis or me your assignment after the due date, we will not accept it.

Grading Rubric

ArrayList Compiles	5 points
ArrayList Runs	5 points
Thoroughness of your test cases	5 points
Instructor test cases (17 cases 5 points each)	85 points
	100 points

In particular, the assignment will be graded as follow, if the submitted solution

- Does not compile: 0 of 100 points
- Compiles but does not run: 5 of 100 points
- Compiles and runs with no errors: 10 of 100 points
- Thoroughness of your test cases: 15 of 100 points
- Passes all 17 test cases developed by instructor: 100 of 100 points.