CSC 212 Programming Assignment # 0
Warm Up
Due date: 03/02/2019

Guidelines:	This is an **individual** assignment. This assignment is optional. The assignment must be submitted to **Web-CAT**

1. We want to implement a generic array class having the following interface:

```
public interface GArray<T extends Comparable<T>> {

  // Return the element at position i
  T get(int i);

  // Set the element at position i
  void set(int i, T e);

  // Changes the size of the array to n. If n is less than the current
     size, the first n elements are kept.
  void resize(int n);

  // Returns the length of the array.
  int length();

  // Executes selection sort (increasing order) on the first n elements
     of the array until at most k swaps are made.
  void selectionSort(int n, int k);

}
```

The method `selectionSort` performs the selection sort algorithm on the n first elements of the array until at most k swaps are made or the n first elements are sorted.

Example 1. *For the array* `a={3, 5, 2, 1, 4}`*:*

- *Calling* `a.selectionSort(4, 2)` *results in* `a={1, 2, 5, 3, 4}`.
- *Calling* `a.selectionSort(4, 8)` *results in* `a={1, 2, 3, 5, 4}`.
- *Calling* `a.selectionSort(4, 0)` *results in* `a={3, 5, 2, 1, 4}`.

Write a class that implements the interface `GArray`. Additionally, complete the implementation of the class `GArrayFactory` which simply creates and returns an object of your implementation of the interface `GArray`.

```
public class GArrayFactory {
  // Create and return an array of size n
  public static <T extends Comparable<T>> GArray<T> getGArray(int n) {
    return null; // Change this.
  }
}
```

2. Write the class `GArrayUtils` that implements the two following methods;

```
public class GArrayUtils {

  // Returns the union of the first n elements of A and the m first
     elements of B. A and B may contain duplicate elements. The output
     should not contain any duplicate elements.
  static <T extends Comparable<T>> GArray<T> union(GArray<T> A, int n,
     GArray<T> B, int m) {
    return null;
  }

  // Returns the intersection of the first n elements of A and the m
     first elements of B. A and B may contain duplicate elements. The
     output should not contain any duplicate elements.
  static <T extends Comparable<T>> GArray<T> intersect(GArray<T> A, int
     n, GArray<T> B, int m) {
    return null;
  }

}
```

1 Deliverable and rules

You must deliver:

1. Source code submission to Web-CAT. You have to upload the following classes in a zipped file:

   - `GArrayFactory`
   - Your implementation of the interface `GArray` (the name of this class is not important).
   - `GArrayUtils`.

   Notice that you should **not upload**:

   - The interface `GArray`.

The submission **deadline** is: **03/02/2019**.

You have to read and follow the following rules:

1. The specification given in the assignment (**class and interface names, and method signatures**) must not be modified. Any change to the specification results in compilation errors and consequently the mark zero.

2. All data structures used in this assignment **must be implemented** by the student. The use of Java collections or any other data structures library is strictly forbidden.

3. This is an individual assignment. Sharing code with other students will result in harsh penalties.

4. Posting the code of the assignment or a link to it on public servers, social platforms or any communication media including but not limited to Facebook, Twitter or WhatsApp will result in disciplinary measures against any involved parties.

5. The submitted software will be evaluated automatically using Web-Cat.

6. All submitted code will be automatically checked for similarity, and if plagiarism is confirmed penalties will apply.

7. You may be selected for discussing your code with an examiner at the discretion of the teaching team. If the examiner concludes plagiarism has taken place, penalties will apply.