**CSCI 3302 Programming Assignment 01 (100 Points)**

**Due: Sept 15 8:00 AM**

GITHUB Link: [Program 01](https://classroom.github.com/a/lctCPznv)

Objectives:

* Demonstrate a basic understanding of Java programming.
* Demonstrate basic competency in Java text file IO.
* Demonstrate understanding of the Java [ArrayList](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html) class.
* Demonstrate understanding of using Git/GitHub for assignments.

Assignment Assistance:

* This homework assignment is due prior to the date and time specified above.
* This assignment is restricted to individual effort. You may not receive help from any other person except the instructor, the TA, or the AARC (help from the TA and AARC must be well documented!).
* Any resource used (other than Dr. Becnel or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or in whole from any source.
* If you need help, visit your instructor during his posted office hours. If your schedule cannot accommodate any of these times, then email your instructor to schedule a different time.

Problem Description:

1. Write a Java class called OrderedIntegers in a class file called OrderedIntegers.java.
2. The OrderedIntegers class contains a single attribute; an ArrayList of type Integer named numbers.
3. The OrderedIntegers class contains a constructor and five methods as members of the OrderedIntegers class:
   1. The lone public constructor takes no arguments. The constructor should instantiate the numbers attribute.
   2. A public method called getSize returns the number of integer values stored in the numbers attribute as type int.
   3. A public method called getValue that takes a single argument of type int called index and returns a value from the numbers attribute at that index. If the index is out of the allowable range, then the method throws the OrderedIntegersException (see below).
   4. A public method called addValue that takes a single argument called newNumber of type int. The newNumber is stored in the numbers attribute such that numbers remain in sorted order from lowest value to greatest value.
   5. A public method called removeValue that takes a single argument of type int called index and removes a value from the numbers attribute at that index. If the index is out of the allowable range, then the method throws the OrderedIntegersException (see below).

* 1. A public method called loadFileData. This method takes a single attribute called filename of type String. The method reads a series of integer values from a binary I/O file (NOT a text file) and stores these values in the numbers attribute. The contents of the binary file are not necessarily in sorted order. The attribute numbers is not necessarily empty prior to calling this method, and any data stored in the attribute should not be lost because of this method call. If the file does not exist or is corrupted then the method throws the OrderedIntegersException (see below); this is done by first catching the IOException and then throwing the OrderedIntegerException.

1. To make sure that your implementation is working correctly, you need to create an OrderedIntegerException class as an extension of a RuntimeException. This code is contained within a separate source code file called OrderedIntegersException.java. This exception needs to be thrown if an attempt is made to access an illegal index within the numbers attribute or a problem occurs when reading data from a file. Create a file called OrderedIntegersException.java containing the following code:

**public** **class** OrderedIntegersException **extends** RuntimeException {

**public** OrderedIntegersException(**String** s) {

**super**(s);

}

}

1. Your files OrderedInteger.java and OrderedIntegerException.java should NOT contain a main method. You can include files with testing code; however, these will not be considered for grading. If you wish to include non-working code for insight into your thought process, make sure to contain it within comment blocks and ensure that submission successfully compiles.
2. For testing purposes, you should include the following two methods in OrderedInteger.java  
   A picture containing text, screenshot, font

   Description automatically generated
3. Your program should work in the GitHub codespace (Linux environment) and locally (Windows environment).

Submission:

* Review the Evaluation below to ensure you have met all the requirements.
* Commit electronic copy of OrderedInteger.java and OrderedIntegerException.java to GitHub. Upload a backup copy to D2L.

**Evaluation**

* 1. Project is late or not submitted at all. -100
  2. Project does not run/compile. -50
  3. Project compiles with warnings. -30
  4. Project does not correctly implement the interface. -30
  5. Calculations/output are for requirements incorrect. -10 each
  6. Code is not well organized or properly indented. -5
  7. Code is inadequately commented for readability. -5
  8. Code does not contain the student’s name, course section, -5

and date of submission.

* 1. Code is not submitted to the Github -15