CSC 101 Programming Assignment #1 01/16/16

Due date – **Friday** **10/7/16 at Noon**

This project is to be done by a pair (2) of students. No groups of three or one without the instructor’s permission. The phrase “I like to work alone” is not an acceptable reason for not having a partner. You will need to learn to work together. Make sure that both of you understand all aspects of the assignment. Note: If the work is not evenly done, this will no doubt be seen during exams.

**Project # 1 Implementation of DoubleArraySeq class**

Your task is to create the class that is discussed in chapter 3, section 3.3 of your textbook. The assigned project in the back of the chapter is #4 on page 169. You must also add the six methods suggested in the problem which are described below. Plus, you are to add the methods **equals** and **toString** to the class. Lastly you must create a Test class that allow the user to create sequences and manipulate them using your class. This is also described below.

Along with this document in the assignments folder should be a file called **DoubleArraySeq.java** which contains the template of the class along with all written specifications for the methods (except for the additional eight) You should grab a copy of this file to work off of.

The eight additional methods that you need to add should be named the following:

**void addFront( double )**

A method to add a new element at the front of the sequence and make it the current element.

**void removeFront( )**

A method to remove the element at the front of the sequence.

If there is a next element, make that element the current element.

Throw an IllegalStateException if the sequence is empty

**void addEnd ( double )**

A method to add a new element at the end of the sequence and make that element the current element

**void setCurrentLast( )**

A method that makes the last element of the sequence the current element

Throw an IllegalStateException if the sequence is empty

**double getElement ( int n )**

a method that returns the nth element of the sequence, and make current element to the nth element(n does not represent the array location)

Throw an IllegalStateException if the sequence is empty, or if n is greater than the sequence size

**void setCurrent ( int n )**

a method that makes the nth element become the current element

Throw an IllegalStateException if the sequence is empty, or if n is greater than the sequence size

**boolean equals ( Object )**

a method that returns true if sequence is the same length and order and data, current element could be different.

**String toString ( )**

Outputs all elements in order separated by a space

Throw an IllegalStateException if the sequence is empty

As you add each of these methods you need to also add the specifications in the Javadoc comments for each new method.

The Test program will declare two sequence objects, sequenceA and sequenceB. Use each of the constructors once, capacity of second one is 5. Then the program will have a menu output to the screen with the following choices:

1. Print the sequences of A and B
2. Report the capacity of A and B
3. Report if A and B are equal
4. Change active sequence (default is A)
5. Add a number to the front of a sequence
6. Add a number before a given number(user inputted)
7. Add a number after a given number(user inputted)
8. Add a number to the end of a sequence
9. Add sequence B to end of A
10. Delete a number at a certain index
11. Delete the first number from the sequence
12. Display a number at a certain index
13. Display the last element in the sequence
14. Trim extra memory from both A and B
15. Create a clone sequence and show
16. Create a new sequence using concatenate of B and A and show
17. Quit

These choices will allow the user and you to test your class well. For menu items that work on only one of the two sequences, you will use the active sequence. This can be changed at any point by picking choice 4. For any choices that change a sequence you should automatically show the new sequence. After each choice is finished the menu should pop up again to reshow the choices. In some conditions exceptions are thrown based on the state of the sequences, you should be attempting to catch or avoid these to allow the program to run smoothly.

Tips for good grades:

* Use of your programs should be user friendly- I should not have to wonder if the computer is waiting for me to input a value without having been given direction to.
* Programs should begin and end with friendly messages.
* Make sure you use comments where needed and use variable names that make sense, some of your grade will depend on program style as well as the use of your program.
* Update the comments in the class file, to include your names and any new information
* You will lose points for things like not indenting, or naming variables in non-descriptive ways. Do no leave in debugging code, or commented out code.
* I use jGrasp and the java version that is in the lab computers. So make sure that your programs work with this.
* Test your own projects thoroughly before you hand them in.
* Late projects will not be accepted so plan ahead.

Each of these programs must be done in a separate file. Name them **DoubleArraySeq.java, and** **SequenceTest.java**. (The .java is added automatically when you save a java file) This means you should be naming the main class in each file these names as well. If you do not name these files correctly, you will lose points. Use Javadoc to create the documentation for your ADT class.

Hand in electronically – (NOT E-mail!!!)

In S-drive CSC 103 folder:

1. Create folder called **projectone\_firstname\_lastname**
2. Place these two script files named above in folder.
3. Place documention from javadoc for your DoubleArraySeq class in folder