CSC103 Fall 2013

**Sara Wexler**

**Assignment #2**

**This assignment is to be done by a group of two persons**

You can change partners. You are not required to stay with the same partner you worked with on project one.

**Do programming project 4 from chapter 3 on page 169**

The class is called **DoubleArraySequence**

Provide the additional methods as stated in the book with the following names:

1. **void addFront** **( double num)**

A method to add a new element at the front of the sequence

And make it the current element

1. **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 exception if the sequence is empty

1. **void addEnd ( double num)**

A method to add a new element at the end of the sequence

And make that element the current element

1. **void currentLast**

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

Element

Throw an exception if the sequence is empty

1. **double retrieveElement ( int i)**

a method that returns the ith element of the sequence, and make current element to the ith element

Throw an exception if the sequence is empty, or if i is greater then

the sequence size

1. **void setCurrent ( int i)**

a method that makes the ith element become the current element

Throw an exception if the sequence is empty, or if i is greater then

the sequence size

**Highlight all new code in the DoubleArraySeq class**

Outline of Java Source Code for this class can be found on the web page:

<http://www.cs.colorado.edu/~main/edu/colorado/collections/DoubleArraySeq.java>

This file contains only blank implementations ("stubs").

2. **Test your program:**

Write a class named **SequenceTest** which will test the methods in

the class **DoublerraySeq.** This SequenceTest class should have instance variables:

sq1 – the original sequence,

sq2 – the second sequence to be created,

sq3 – for the cloned sequence.

And at least the following methods (you may need more, it depends on your design):

* **menu**

**public void menu ( String st);**

The method creates a menu with the following options:

1. Create a sequence

2. Delete a number

3. Delete the first number from the sequence

4. Add a number before another number

5. Add a number after a number

6. Add a number to the end of the sequence

7. Display a number at a certain index

8. Display the last element in the sequence

9. Replace a number with another number

10. Append another sequence to the first sequence

11. Create a clone sequence

12 Print the sequence

13. Quit

This method has a string parameter **st**.

This st string will have the option to select from the menu, along with the other numbers that are related to that option, examples:

I. 1- 20 30 40 50

Which means the option is 1 – to create a sequence from the

following numbers: 20 30 40 50

II. 4 -5 40

Which means, option 4 –Add number 5 before the number 40

## 

## Each line of input should have an output with the explanation what is required to be done

## Example1.:

Input Line: 1- 20 30 40 50

Will create the following output::

**Input line: 1 – 20,30.40,50**

**Create a sequence**

**-------------------------**

**The sequence : 20 30 40 50**

**Number of elements: 4**

**The current element: 50**

( look for sample of output below)

## **Call the method printSequence after each operation**

* **createSequence**

**public createSquence (String st)**

This method returns an object of the class **DoublerraySeq.**

The method will parse the string parameter, create a token for each number in the string and will add it to the sequence, one number at a time.

* **printSequence**

**A** method to display **:**

* a comment what is being done; example: insert number 5 after the number 520
* The sequence,
* the number of elements in the sequence
* the current element.

You should determine the parameters for the method

* **Find( double num)**

**public int find ( double num)**

A method to find element num in the sequence, if the element was found then the method returns the index, else the method returns -1.

1. Create another file called: **Lab2.java** which will a from the input file

(look bellow) and will call **Sequence Test**. This file will have main in

it.

**Turned In :**

1. Use **closeable envelope / folder** **marked clearly outside with names, class section and assignment #**
2. All source code written for the assignment;
3. **Write your** **names at the top of the source code** and **the date**
4. Highlight all new code in the DoubleArraySeq class
5. Hard copy of **test cases** run
6. Disk with all files relevant to the Assignment (.java and .class files).

**Due Date :** at the beginning of class on **Monday Oct 14 ( week 7)**

Use this guideline for checking your program over before handing it in:

* **Documentation ( similar to the Sample documentation on the M drive)**
* Hard copy of **source code** **stapled together**
* **UML Diagram** of the classes
* **Test cases** stapled together
* You name in a comment at the top of each file
* Due Date in a comment **under** your name
* Program Description after the Date
* Proper indentation throughout your code
* Descriptive identifiers
* Constants defined as needed
* Comments:

comments at the beginning of program explaining the purpose of the

program   
 comments in the program to clarify code

generating javadoc

if you give poor comments or not enough you will loose 20%

* Properly Labeled Disk- neatly print Your Name, CSC103, Wexler
* Disk with all files relevant to the Assignment (including executable).
* File names are as given above
* Proper indentation throughout your code
* Formatted Output - Correct spelling and capitalization is consistent.
* Correct Output

**Grading criteria**

1. - if the project is working good, get correct output, and there are sufficient comments
2. - if the project is working good, but there are some problems with the

code, or some of the output is incorrect

or the comments are not sufficient

2 - if the code compile but does not run

1 - if the does not compile

**Input file for testing Lab2**

1-

1 - 100,400,200,700,500

2 – 400

4 - 3,700

5 - 5,200

3

6 - 25

7 - 3

8

1 - 10,20,30

10

11

9 - 700, 7

12 - 1

12 - 2

**The output should look like:**

Input line: 1-

Create a sequence

-------------------------

Exception – no data was created

Input line: 1 - 100,400,200,700,500

Create a sequence

-------------------------

The sequence : 100 400 200 700 500

Number of elements: 5

The current element: 500

Input line: 2 – 400

Delete a number

-------------------------

The sequence : 100 200 700 500

Number of elements: 4

The current element: 200