CSC 103 Programming Assignment #2 09/28/16

Due date – **Fri October 28th by 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 # 2 Implementation of UnboundedInt class**

Your task is to create the class that is discussed in chapter 4, programming project 10 on page 249 of your textbook. You must use a linked list of integers to store the numbers. The use of the **IntNode** class (unaltered) that we discussed in class from chapter 4 is expected. There is a fresh copy of this class in with this assignment. The **UnboundedInt** class will contain four instance variables, the number of Nodes, link to front of list, link to back of list, and a cursor which will transverse the list pointing to any nodes.

Although you can store the representation of your numbers in either way, it makes more sense if you do the lower value terms at the front of the list. For example, to represent the number 12,453,075 you would put a 75 in the first Node, then a 453 in the second and a 12 in the third. This order will help you when you are attempting to add or multiply two numbers of unknown length.

Your class must have the following methods at minimum:

**Constructor(String)**

This constructor will take a string of digits and turn it into an unboundedInt object

**UnboundedInt add (UnboundedInt )**

A method that adds the current UnboundedInt with a passed in one. The return is a new UnboundedInt.

**UnboundedInt multiply (UnboundedInt )**

A method that multiplies the current UnboundedInt with a passed in one. The return is a new UnboundedInt.

**void start( )**

set the cursor to the front of the list

**void advance( )**

move the cursor along the list

Throw an IllegalStateException if the cursor is null

**int getNodeValue ( )**

a method that returns the integer value of the Node that is pointed to by the cursor.

Throw an IllegalStateException if the cursor is not pointing to a Node

**void addEnd ( int ) -optional method (helpful)**

A method to add a new element at the end of the sequence , used for building up each higher term in a single sequence. (i.e. adding a new IntNode to the linked list)

**UnboundedInt clone( )**

a method that returns a copy of the original structure

**boolean equals ( Object )**

a method that returns true if linked list represents the same numerical number as the input parameter. False otherwise. Overrides method in Object class.

**String toString ( )**

creates a string of all elements in order separated by commas, making sure leading zeros are added when needed. (i.e. 12,005,016 or 34,000 )

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 method. You can have more methods than this if needed.

Lastly you must create a Test class that allow the user to create Large Numbers and test your arithmetic using your class.

The Test program will ask the user to input, without commas, two large numbers. You will read them in as strings and then use your constructor to make then into stored **UnboundedInt** objects. Then the program will have a menu output to the screen with the following choices:

1. Display both numbers
2. Input two new numbers
3. Check if numbers are equal
4. Report the sum of the two numbers
5. Report the multiplication of the two numbers
6. Create and output the clone of the numbers(at least one of them)
7. Quit

These choices will allow the user and you to test your class well. After each choice is finished the menu should pop up again to reshow the choices.

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 classes must be done in a separate file. Name them **UnboundedInt.java,** and **LargeNumberTest.java**. (**IntNode.java** should be there as well – but not altered). 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 **projecttwo\_firstname\_lastname**
2. Place these three script files named above in folder.
3. Place documention from javadoc for your UnboundedInt class in folder