

# Assignment 3

## CPS501 –Advanced Programming and Data Structures

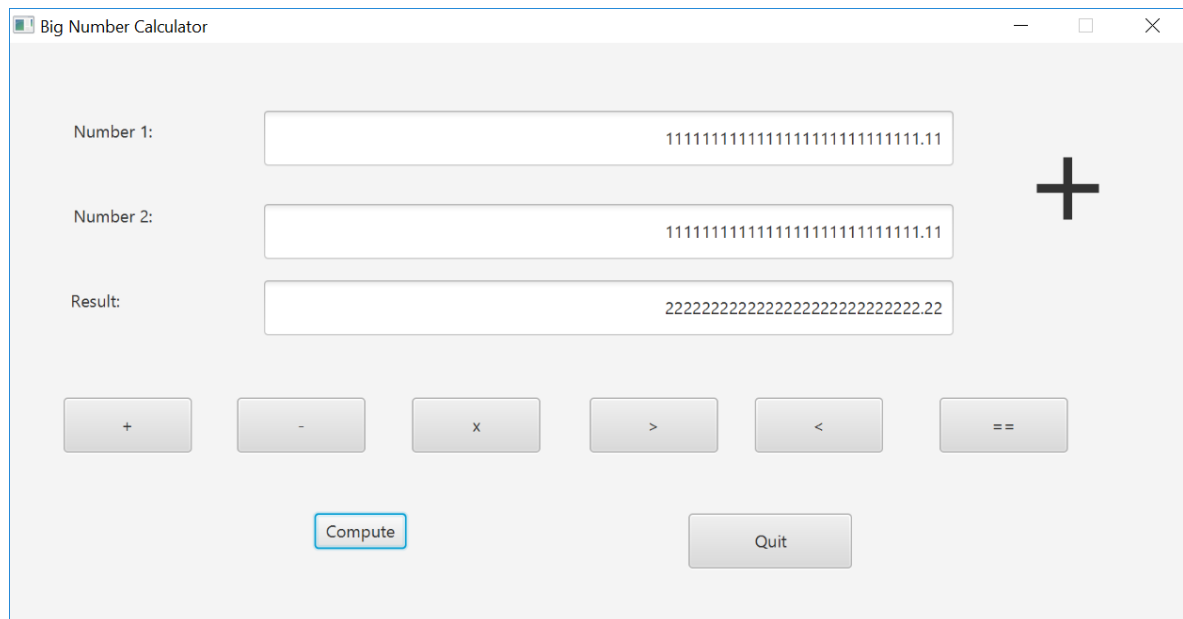
**Deadline: 11/04/2018**

### Requirements

In this assignment, you will solve one practical and interesting problem in programming. By completing the project you will gain valuable hands-on experience in the design, implementation and evaluation of programming algorithms.

### Problem Description

Implement an advanced “Big Number” calculator by using JavaFX GUI. Use TextFields to show the input and output result and Buttons to set the operator (add +, subtract -, multiply x, larger >, smaller <, equal ==). Note that a Big Number can be a floating-point number (0.1 or 1111111111111111111.5). One example of the GUI is shown as below.



Design the GUI by using Scene Builder. The function of each GUI control is described as below.

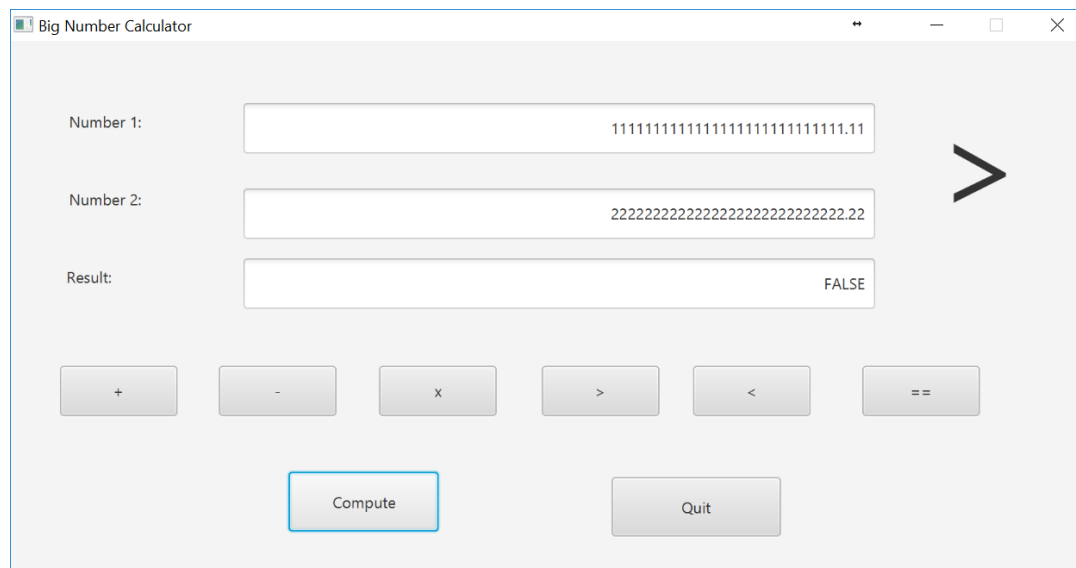
- TextField: receives input and output the result
- Label (on the right corner): display the current operator/comparison
- “+, -, x”: basic operator (add, subtract/multiply)
- “>, <, ==”: larger, smaller, equal comparison
- “Compute”: Compute and display the result to the TextField “result”
- “Quit”: quit the application

Implement the **BigNumber** class which is a **Linked List** (**Singly Linked List** or **Doubly Linked List**). Namely, each digit in the BigNumber is an element in the Linked List.

Using existing classes such as **BigDecimal** or **BigInteger** class is **not allowed** for this assignment. You are also **not allowed** to use the **String** class to implement the **BigNumber**. Only the constructor **public BigNumber(String s)** is allowed.

Please implement all the operators (+, -, x), and comparison (<, >, ==) for the **BigNumber** class. Again, the **BigNumber** can handle the **floating-point** number. The **BigNumber** can handle the large-scale number, for example, 111111111111111111111111111111.11.

Another example is given as below.



How it works:

1. User inputs 111111111111111111111111111111.11 and 222222222222222222222222222222.22 for Number 1 and Number 2, respectively.
2. User also inputs ">" comparison.
3. The label on the right corner is updated to ">".
4. Then, the user clicks "Compute" button.
5. The resulted TextField is updated to "FALSE".

### Hint

The recommended **BigNumber** class is as follows.

```
public class BigNumber
{
    SLL digits; //The linked list is used to store all the digits of the BigNumber.
    int decimalPointPosition; //the position where the decimal point is, e.g.,
    1.2: decimalPointPosition = 1

    public BigNumber(String s){ // This constructor is used to transfer the text (in the
    TextField) to SLL digits, each SLL node will store one digit (in the reverse order), for
    example, 16 is stored as 6 -> 1 -> null
```

```

    BigInteger add(BigInteger n){} //This function is used to compute the addition of the
    current BigInteger with another BigInteger n, For example: 1 6 + 3 4 -> 6 -> 1 -> null + 4-
    >3 -> null = 0 -> 5 -> null

    BigInteger subtract(BigInteger n){} //This function is used to compute the subtraction of
    the current BigInteger with another BigInteger n

    BigInteger multiply(BigInteger n){} //This function is used to compute the
    multiplication of the current BigInteger with another BigInteger n

    String toString(){} //This function is used to print the BigInteger is the right order. If you
    use Singly linked list, the digits are stored in the reverse order. For example: 6 -> 1 -> null
    => "16"
}

```

## What to Submit

1. A well-documented program that implements the problem in the Assignment 3. You must submit your program source code.
2. A well-written, concise project report. It should include: (a) title and names of group members; (b) the analysis of each problem; (c) the issues during the implementation; (d) the solutions to overcome the issues in (c); (e) the contribution of each individual member
3. The powerpoint slides (maximum 20 slides) used in the Assignment grading.

For each group, you must submit the files above in a single zipped folder. Your group will be required to do a presentation in classroom for the grading.

Note: If you cannot submit zipped file to isidore, please change the filename extension to doc or docx and then submit it.

**Submission Due: 11:55pm, November 26, 2018**

## Hint to Multiplication

### First Step

The first step in multiplication is to align the numbers on the right. You will use the numbers from the example: **469 x 32**. You begin with the smaller number. This is the 32. You start with 2. You multiply  $2 \times 469 = 938$  as below. You store **938** to a list.

$$\begin{array}{r}
 11 \\
 469 \\
 \times 32 \\
 \hline
 938
 \end{array}$$

1)  $2 \times 9 = 18$  carry the 1 on top of the 6  
 2)  $2 \times 6 = 12 + \text{the carried } 1 = 13$ , carry the 1 on top of the 4  
 3)  $2 \times 4 = 8 + \text{the carried } 1 = 9$

### Adding a Zero for the Tens Space

Now you need to multiply by the next number to the left of the 2. This is the 3 in 32. Because the 3 is in the tens place you need to hold the tens place by placing a zero in the 1's place before you start multiplying.

$$\begin{array}{r}
 469 \\
 \times 32 \\
 \hline
 938 \\
 0
 \end{array}$$

Place a zero here to hold the 10s place

### Finish Multiplying

Multiply the 3 by the top number (469) and add this number next to the zero. So, you multiply 469 by 4 = 14070 as below. You store **14070** to another list.

$$\begin{array}{r}
 22 \\
 469 \\
 \times 32 \\
 \hline
 938 \\
 14070
 \end{array}$$

1)  $3 \times 9 = 27$ , carry the 2  
 2)  $3 \times 6 = 18 + \text{the carried } 2 = 20$ , carry the 2  
 3)  $3 \times 4 = 12 + \text{the carried } 2 = 14$

After you have multiplied all the numbers on the bottom, you sum two stored lists: 938 and 14070.

$$\begin{array}{r}
 469 \\
 \times 32 \\
 \hline
 1938 \\
 14070 \\
 \hline
 15008
 \end{array}$$

Finally, you have a new list: **15008**.