Roman Numeral Calculator

Design and implement a Roman numeral calculator frame.

Your program will read a series of Roman arithmetic numeral operations from a data file and will print the result of the operations, also as a Roman number.

The user may specify the filename. If the filename is not entered, the data file *romanNumeral.txt* will be used.

Your program should also check for errors in the input, such as illegal digits or arithmetic operators and take appropriate actions when these are found. The arithmetic operators that your program should recognize are +, -, \*,/ and %.

One way of approaching this problem is to convert the Roman numbers into integers, perform the required operation, and then convert the result back into a Roman number for printing.

**The Java Classes**

Your program should consist of two classes – a Java frame called **RomanNumeralCalculatorFrame** and a **RomanNumeral** class.

The **RomanNumeral** class is described in the **Roman Numeral.docx** file.

The **RomanNumeralCalculatorFrame** should accept the filename and display the list of the calculations when the "Calculate Now!" button is clicked.

**The Data File**

Each line of the data file represents one arithmetic operation and contains three fields separated by a space:

The first Roman Numeral operand

The operator

The second Roman Numeral operand

For example, the *romanNumeral.txt* file could contain the following:

MCCXXVI + LXVIIII

MCCXXVI - LXVIIII

X \* V

CC / L

MDCLXVI % XVI ­­

**Error Handling**

### Your program should do error checking on the keyboard and file input. Be sure to provide test cases for errors in the data. Errors in the data from the keyboard should be displayed in a dialog box and the user should be given the opportunity to reenter the data. Errors in the file data should result in that record being skipped and a message being displayed in the text area.

**The Program Output**

For each operation the report should show the Roman and integer values for the two operands and the result in the form of an arithmetic statement.

Set the font in the text area to monospaced:

**resultDisplay.setFont(new Font("Monospaced", Font.PLAIN, 12));**

A sample run of the program using the data file above is shown on the next page.

