Lab 4: Java Fundamentals, Part IV

20 pts

Distribute on October 6, 2025
Due October 12, 2025 (Sunday) at 11:59 pm, before 12:00 midnight

Learning Outcomes (CLO vs SO Mapping)

- Utilize Java syntax in fundamental programming algorithms (3)vs(1)
- Recognize and apply the various input and output devices in programming (4)vs(2)
- Recognize and apply the various control structures (5)vs(1)
- Recognize and apply the basic debugging strategies in programming (8)vs(1)

Requirements

With this lab, you gain experience with fundamental Java constructs, including variables, assignment statements, the String and JOptionPane classes, and if-else selection structures.

Preliminaries

1. Create a Java project. The name of the project must be **lab04_<your FirstNameLastName>**. For example, my project would be named lab04_PeterNg.
2. Add a class named **`RomanNumerals`**. Declare the class as your **main class** when the project is set up with Eclipse
3. Make sure you have the usual comment block at the beginning of your class:

```
/*
 * <your name>
 * CS 16000-01 – 02/03, Fall Semester 2025
 *  (Note: Write either 02 or 03, depending on which section your section is.)
 * Lab 4
 *
 */
```

***Exercises (20 pts) (*The GTA must verify your work in the lab to get credit for this lab)**

In this assignment, you solve a conversion problem similar to **Programming Challenges: problem 1 Roman Numerals** of Chapter 3 of your text. Given one of the Roman numerals **I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII, XIII, XIV, XV, XVI, XVII, XVIII, XIX, XX** as **input** your program must determine and display the corresponding decimal digit **1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16. 17, 18, 19, 20.**

Requirements

I. Converter: The task of the Converter is to convert a Roman numeral into a decimal of the int data type. The main() invokes this Converter, which returns a decimal value for a given Roman numeral. This Converter can be constructed as follows:

```
public static int Converter(String roman) {
        //Find the solution for converting a Roman numeral into a decimal.
        //The Converter has to be invoked (i.e., called) from the main() at the end of
        //the problem VIII and before problem IX

        //The Converter translates any given Roman numeral to a decimal and returns
        //this decimal (i.e., an integer value) for the Roman numeral.
        //The body of the Converter is described in Problems I, I.1, I.2, VII, XII, and XIII.

        //The Converter returns an integer value for the given Roman numeral.
        return decimal;}
```

The solution must use a nested **if-else** construct built as follows:

1. The top-level if-else decision is made upon the first character of the Roman numeral, which is always **`'I', 'V', or 'X'`**. That is, divide the given group of Roman numerals into three subgroups according to their first character, **`'I', 'V', or 'X'`**.

```
//initialize decimal to 0;
if (roman.charAt(0) == 'I') {
    … //For I group beginning with I, specified in I.2.
}
else if (roman.charAt(0) == 'V') {
    … //For V group beginning with V, specified in I.2.
 }
else if (roman.charAt(0) == 'X') {
    … //For X group beginning with X, specified in I.2.
}
return decimal;
```

2. Both the **if** and **else** blocks contain deeply nested if-else structures where the input string is compared to the Roman numerals I, II, III, IV, and IX in the first block and V, VI, VII, VIII in the second block, and then to X, XI, XII, XIII, XIV, XV, XVI, XVII, XVIII, XIX, XX in the third block.

```
        //group I
            if (roman.equals("I")) {
                decimal = 1;
            } else if (roman.equals("II")) {
                decimal = 2;
            } else if
            …
            } else if (roman.equals("IX")) {
                decimal = 9;
            }

        //group V
            if (roman.equals("V")) {
                decimal = 5;
```

```
        } else if
        …
        } else if (roman.equals("VIII")) {
                decimal = 8;
        }

//group X
        if (roman.equals("X")) {
                decimal = 10;
        } else if
        …
        } else if (roman.equals("XX")) {
                decimal = 20;
        }
```

II. Input is solicited on a dialog window, as shown in Figure 1.

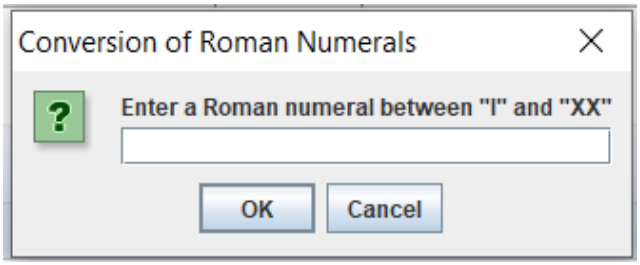

Figure 1

III. Declare a String variable **title** to store the title text, such as "Conversion of Roman Numerals." Declare a String variable **task** to store the input solicitation text, such as "Enter a Roman numeral between "I" and "XX". You must use the variables in the showInputDialog() method call that creates the window. For example, JOptionPane.showInputDialog(null, task, title, JOptionPane.QUESTION_MESSAGE). The input window must follow the template of Figure 1.

```
String title = "Conversion of Roman Numerals";
String task = "Enter a Roman numeral between …;
```

IV. Declare a String variable **roman** to save the input string returned by the window.

```
String roman;
```

V. Use if (roman == null) for the following case: If you press the Cancel button in the message box, as shown in Figure 1, your program should display the message "You pressed the Cancel button!" as shown in Figure 2. Then, after pressing the OK button in the message box shown in Figure 2, the message in Figure 3 is displayed in the message box, as shown in Figure 3, before the program ends. (This is a way to solve cancel messages!)

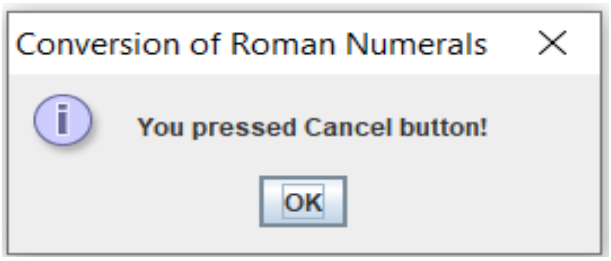

Figure 2

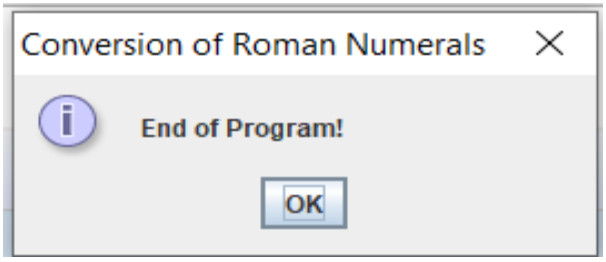

Figure 3

VI. If you pressed OK without entering a Roman numeral in the message box, as shown in Figure 1, your program should display the following message: "You pressed OK without a Roman numeral!" through a message box, as shown in Figure 4. Then, after pressing the OK button in the message box shown in Figure 4, the message in Figure 3 is displayed in the message box, as shown in Figure 3, before the program ends. (This is also a way to solve the error message!)

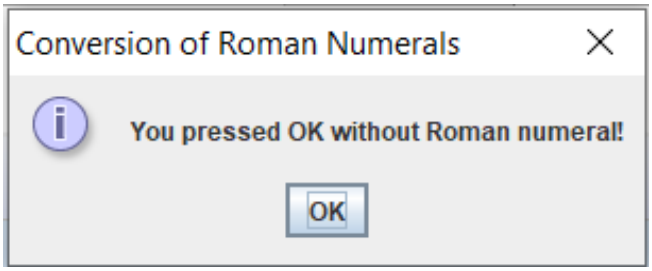

Figure 4

VII. Converter: Declare an int variable **decimal** to store the decimal digit equivalent of the roman numeral and initialize it to 0.

VIII. Allow using lowercase, uppercase, and even mixed characters as input for Roman numerals. The output Roman numerals must be in upper case. Use toUpperCase() for the conversions (Read Chapter 2: 2.9 The String Class, see toUpperCase, page 75, 7th edition).
Then, call the Converter() method using the **'roman'** parameter**.** The Converter() returns the value for the Roman numerals, which is assigned to the variable **decimal**. Thus, in the main(), write the following statement:

decimal = Converter(roman);

The above statement calls the Converter() method and passes a Roman string to it. After that, the Converter returns a value and assigns it to the variable decimal.

IX. The output must be displayed on a message dialog, as shown in Figure 5, and the output window must follow the template; take note of the title line and the output arrangement. Use the String.format() method for the output arrangement, such as "The decimal value for the Roman numeral "VIII" is: ……. 8 ……" in two lines. (Read Chapter 03: Displaying Formatted Output with System.out.printf and String.format, See the String.format on pages 225-227).

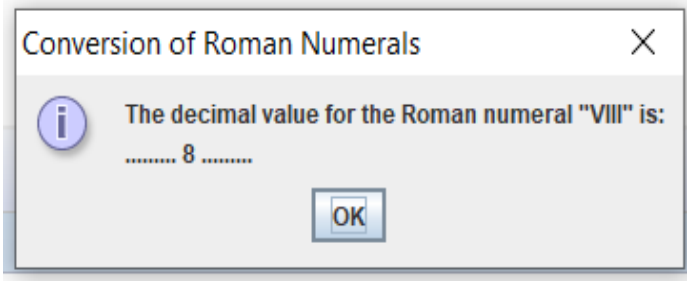

Figure 5

X. We assume that the input string is not empty and not **null** (each triggers a runtime error)

XI. If the input string is none of the ***twenty*** admissible Roman numerals, the message, as shown in Figure 6, is displayed

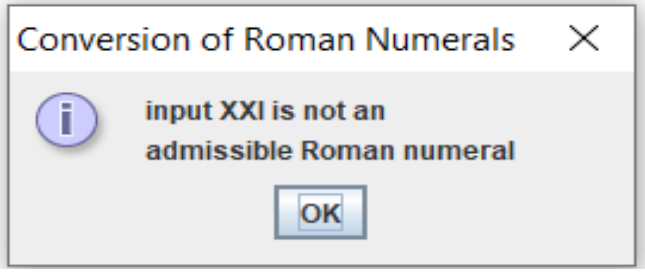

Figure 6

XII. Converter: When you check the first character, use the **==** operator for equality, and do not forget the character operator on the literal, put **`roman.charAt(0) == 'I'`** into the topmost if header**.** (hints)

XIII. Converter: Use the equals method for string comparison (Read Chapter 3: 3.6 Comparing String Objects, pp 142-147, 7th edition). A nested if header should appear as follows. (hints)

```
if(roman.equals("VII")){
```

XIV. The output text (see Figure 5) contains the " " symbols. Use the escape sequence **\"** within the text. (hints)

XV. Use the do-while loop and the showConfirmDialog() method, such as "int yesNo = JOptionPane.showConfirmDialog(null, "Any more Roman Numerals? \nyes or no", title, JOptionPane.YES_NO_OPTION);" asking whether the process should be continued as shown in Figure 7.

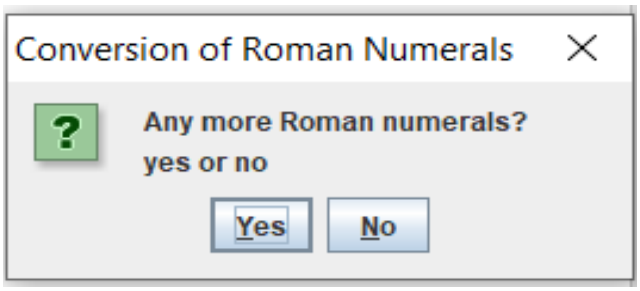

Figure 7

The code for the do-while loop could be:

```
do {
        …

        yesNo = JOptionPane.showConfirmDialog(null,
                                "Any more Roman numerals? \nyes or no", title,
                                 JOptionPane.YES_NO_OPTION);
} while (yesNo == JOptionPane.YES_OPTION);
```

Note that JOptionPane.showConfirmDialog generates an integer value, Yes is 0, and No is 1. JOptionPane.***YES_OPTION*** is equivalent to 0**,** and JOptionPane.***NO_OPTION*** is equivalent to 1**.**

XVI. Finally, before the program ends, a message is displayed through the message box, as shown in Figure 3.

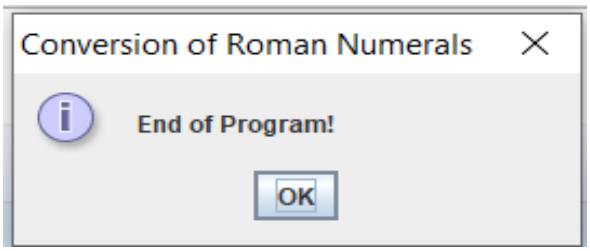

Figure 3

XVII. For the JOptionPane class, you need to use the following two statements in a proper location. Do not excessively state "System.exit(0);" before the end of the main method. The import statement must be placed on top before the "public class RomanNumerals{…}".

```
import javax.swing.JOptionPane;     //needed for JOptionPane class.
```

and

```
System.exit(0);     //required JOptionPane class.
```

XVIII. Your program must be of the following form:

```
import javax.swing.JOptionPane;

public class RomanNumerals_Lab04 {

    public static void main(String[] args) {

        //Declaration block
        …

        do {

        roman = roman.toUpperCase();

        decimal = Converter(roman);

        …
        //ouput

        yesNo = JOptionPane.showConfirmDialog(null,
                                "Any more Roman numerals? \nyes or no", title,
                                JOptionPane.YES_NO_OPTION);
        } while (yesNo == JOptionPane.YES_OPTION);

        …
```

```
            System.exit(0); // required JOptionPane class.

        }// end of main()

        public static int Converter(String roman) {

        //Use nested if to determine the decimal of the Roman numeral referenced by
        //the referenced variable roman.
        …
        return decimal;
        }//end of Converter()

}// end of RomanNumerals_Lab04
```

What to Submit

- upload your zipped project folder to the class website on Brightspace