

Program 6 : 80 points: Swing

Due Friday, 29th April 2016, 11:59 pm

Objectives: The focus of this assignment is the use Swing components to create a simple GUI based program

Program Description:

The purpose of this program will be to provide a simple Binary Calculator.

The user enters the first operand for the calculation using the "0" and "1" JButtons to build a binary number.

The user selects the mathematical operation to be performed using the JComboBox drop-down options.

The user enters the second operand using the "0" and "1" JButtons to build a binary number.

The user then clicks the "Compute" JButton and the results of the calculation are displayed either in Binary or Decimal format, depending on the state of the JRadioButtons, "Binary" and "Decimal".

To perform a new operation, the user must press the "Clear" JButton.

Required Java Files:

A total of two files are required.

BinaryCalculatorDriver.java (provided – no changes needed)

BinaryCalculatorFrame.java

UML DIAGRAM FOR AND DISCUSSION FOR BinaryCalculatorFrame

BinaryCalculatorFrame extends JFrame	
-	TextField txtDisplay
-	JRadioButton rdoBinary
-	JRadioButton rdoDecimal
-	ButtonGroup buttonGroup
-	JButton btnZero
-	JButton btnOne;
-	JComboBox<String> cboOperators
-	JButton btnCompute
-	JButton btnClear
-	
-	String operator
-	Integer operand1
-	Integer operand2
-	Final String [] CHOICES = { "0P", "+", "-", "*", "/", "%" };
<<constructor>> BinaryCalculatorFrame()	

Additional Required Classes:

Additionally, the BinaryCalculatorFrame class must declare two nested, private classes, one for each type of event that may occur:

1. Implement ActionListener (for the btnZero, btnOne, compute, and clear components)
2. Implement ItemListener (for cboOperators component).

The display field does not need a listener since it will only be used for output.

The binary and decimal radio buttons do not need listeners since their states will only be used when the computed result is displayed.

Data Members:

operator – String - represents the operator selected with the following possible values:

- "OP" CHOICES index 0 – No operator selected
- "+" CHOICES index 1 – addition
- "-" CHOICES index 2 – subtraction
- "*" CHOICES index 3 – multiplication
- "/" CHOICES index 4 – division
- "%" CHOICES index 5 – modulus

operand1 – Integer value of the first operand in the calculation

operand2 – Integer value of the second operand in the calculation

Frame Component descriptions:

txtDisplay – JTextField,

- Initial starting text should be "0", with its width to 20. There is a JTextField constructor that accepts the initial String value and field width as an int. Set it as uneditable and its alignment to right justified like this:

```
txtDisplay = new JTextField("0", 20);
txtDisplay.setHorizontalAlignment(JTextField.RIGHT);
txtDisplay.setEditable(false);
```

buttonGroup – Group to include the rdoBinary and rdoDecimal radio buttons

rdoBinary, rdoDecimal – JRadioButtons

- Initialize rdoBinary to true and rdoDecimal to false.
- Add both to buttonGroup

btnZero – JButton, when pressed will append a '0' digit to the end of the value currently displayed in txtDisplay

btnOne – JButton, when pressed will append a '1' digit to the end of the value currently displayed in txtDisplay

cboOperators – JComboBox, List of possible operators.

- List of possible operators (Strings) include:
- "OP" meaning no operator is selected, "+", "-", "*", "/" and "%" representing each of the possible operations.
- When the selection of the box is changed (if it has not been changed to 0 or "OP"), set the 'operator' equal to the corresponding operator from 'CHOICES' depending on the operator selected.
- The current value of the displayed operand should be stored into operand1, the displayed operand cleared out (set to 0) and updated in the txtDisplay.

btnCompute – JButton

- When pressed, perform the arithmetic and display the result in txtDisplay in either binary or decimal format depending on which radio button is currently selected.
- The currently displayed operand will be used as the second operand in the calculation.
- Based upon the currently selected operation calculate the result and sets this as the current value displayed in txtDisplay. Depending on which radio button is selected (either rdoBinary or rdoDecimal) the result should be displayed in the corresponding base.
- Reset the currently selected operator to "OP" (index 0 for cboOperators).
- ERROR HANDLING – If the selected operator is "OP", use a JOptionPane to display instructions:

```
JOptionPane.showMessageDialog(null,
    "Input First Binary Number - Select an Operator - "+
    "Input Second Binary Number - Click Compute",
    "ERROR", JOptionPane.ERROR_MESSAGE);
```

IMPORTANT: DO NOT write methods to do calculations with binary values. Instead, convert the operands into decimal and perform the calculations on the decimal values. The result can then be converted back to binary if required.

The following **static** Integer methods will be extremely helpful:

- String Integer.toString(int) – takes an int and returns a String representation
- int Integer.parseInt(String) – takes a String and converts it into an int
- int Integer.parseInt(String, int) – takes a String and a base (such as base 2 for binary). Converts the String in the provided base and returns its decimal value as an int
- String Integer.toBinaryString(int) – takes a decimal value and returns its binary representation as a String

clear – JButton, resets all values.

- operand1=0, operand2=0, and operator to "OP",
- displayed text to "0"
- selected Operator to "OP".

Other Notes:

- Be sure to import all of the necessary classes
- Name the components as show in the example
- Your window should look exactly like the displayed window shown on the following pages.

Submitting Your Program Files:

1. This assignment cannot be completed on the loki server. The collection of the assignment **WILL NOT** be done automatically.
2. You must zip your project files and submit your zip file to the Blackboard Assignment link for Program 6 on our class Blackboard page by the due date and time.
3. Click on the link below by going to "Blackboard" > "Assignments"



Program 6 – Binary Calculator – due Friday 29th April 2016 at 23:59

Please open this link and uploaded a zipped version of your project files before due date

4. Then go ahead and zip your .java files and upload it in the Program 6 Assignment page as show below before due date.

[Cancel](#)
[Save as Draft](#)
[Submit](#)

ASSIGNMENT INFORMATION

Due Date Friday, April 29, 2016 11:59 PM	Points Possible 80
---	------------------------------

Please open this link and uploaded a zipped version of your project files before due date

ASSIGNMENT SUBMISSION

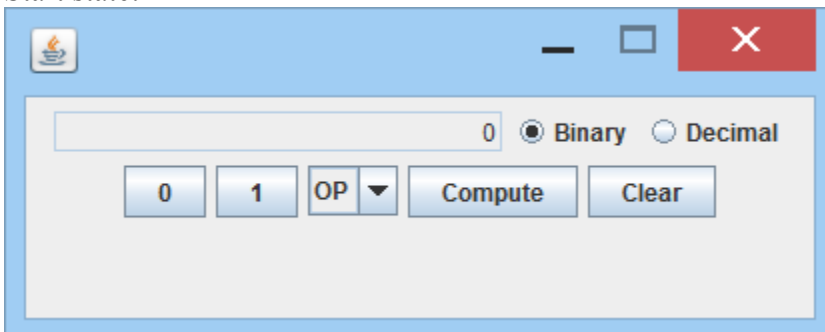
Text Submission [Write Submission](#)
 Attach File [Browse My Computer](#) [Browse Course](#)

ADD COMMENTS

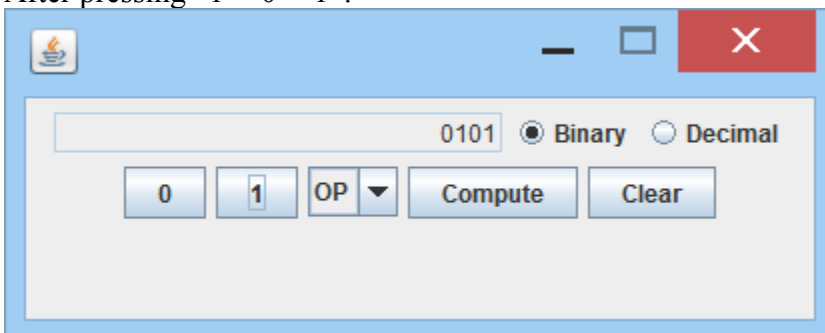
- The time of your submission will be used to determine if your files are submitted late.

Sample Program Execution:

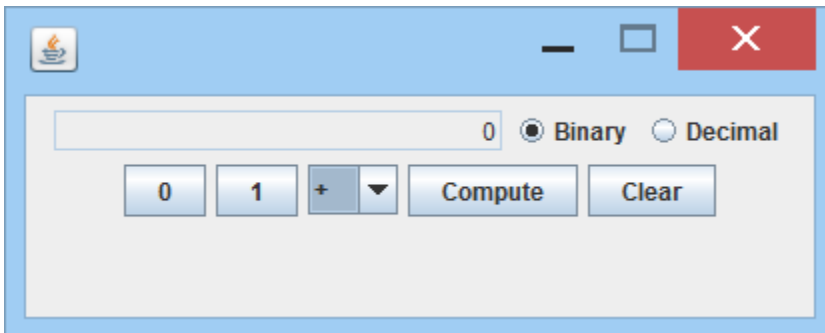
Start state:



After pressing “1” “0” “1”:

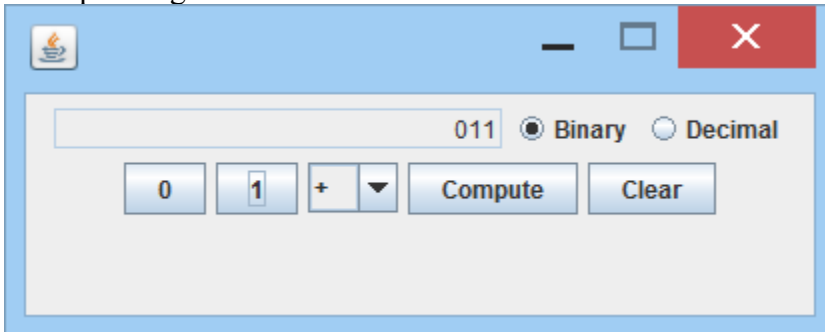


After selecting “+” from the cboOperators combo box:



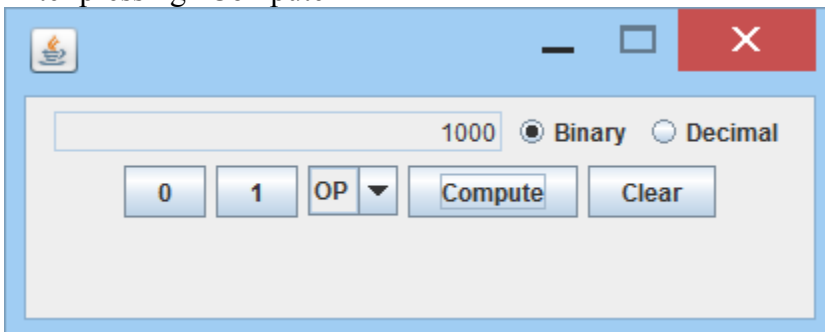
The screenshot shows a calculator application window. The display shows '0'. The 'Binary' radio button is selected. The operator dropdown menu is open, showing a '+' sign. The buttons '0', '1', '+', 'Compute', and 'Clear' are visible.

After pressing “1” “1”:



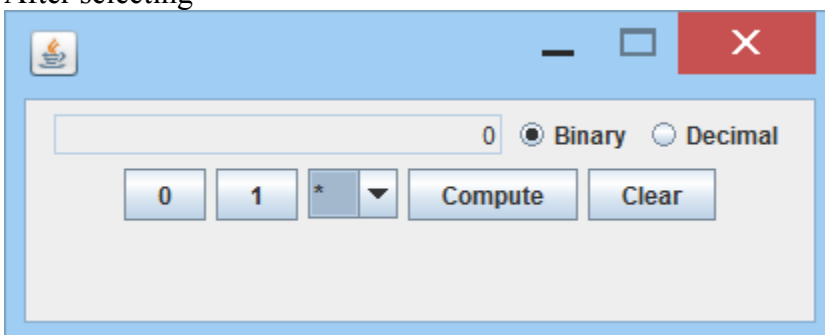
The screenshot shows the calculator application window. The display now shows '011'. The 'Binary' radio button is selected. The operator dropdown menu is still open, showing a '+' sign. The buttons '0', '1', '+', 'Compute', and 'Clear' are visible.

After pressing “Compute”



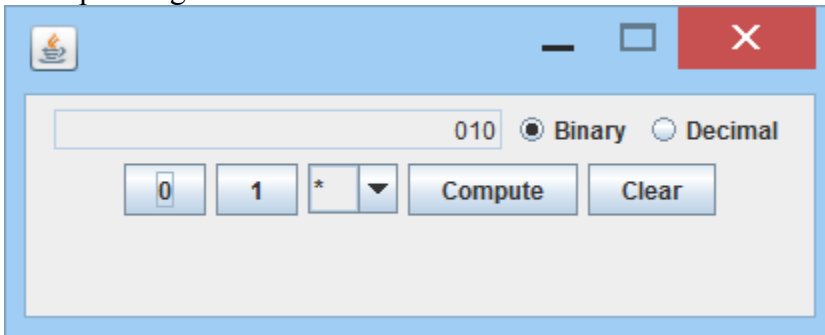
The screenshot shows the calculator application window. The display now shows '1000'. The 'Binary' radio button is selected. The operator dropdown menu is now closed, showing 'OP'. The buttons '0', '1', 'OP', 'Compute', and 'Clear' are visible.

After selecting “*”

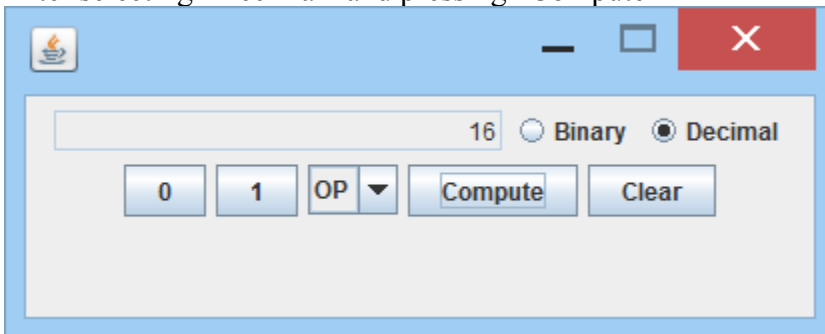


The screenshot shows the calculator application window. The display now shows '0'. The 'Binary' radio button is selected. The operator dropdown menu is open, showing a '*' sign. The buttons '0', '1', '*', 'Compute', and 'Clear' are visible.

After pressing “1” “0”

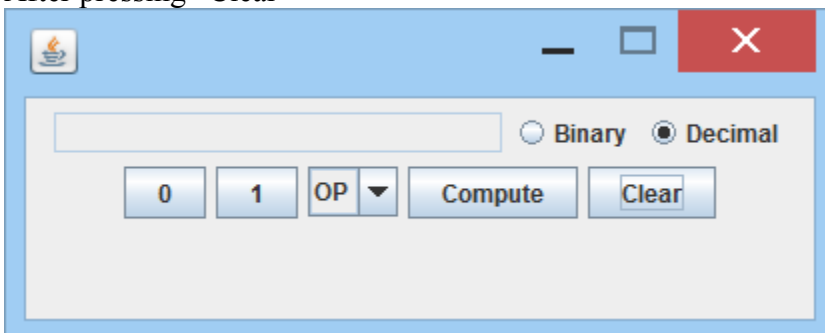


After selecting “Decimal” and pressing “Compute”



Notice that the first operand in the operation was the result of the previous operation. If the user wants to start a new operation they can press “Clear” to start from scratch.

After pressing “Clear”



If the user clicks the Compute button but has not selected an operator – display an error message:

