**OBJECT ORIENTED PROGRAMMING**

**LAB# 09 TASKS**

## **Graphical User Interface(GUI) in Java**

|  |  |
| --- | --- |
| 1. Create a Java GUI application that provides the user usage of Ohm’s Law formula. In electrical circuits, Ohm's law states that the current through a conductor between two points is directly proportional to the potential difference or voltage across the two points, and inversely proportional to the resistance between them, provided that the temperature remains constant.   The mathematical equation that describes this relationship is:    where V is the potential difference measured across the resistance in units of volts; I is the current through the resistance in units of amperes and R is the resistance of the conductor in units of ohms. More specifically, Ohm's law states that the R in this relation is constant, independent of the current.  The law was named after the German physicist Georg Ohm, who, in a treatise published in 1827, described measurements of applied voltage and current through simple electrical circuits containing various lengths of wire. He presented a slightly more complex equation than the one above to explain his experimental results. The above equation is the modern  form of Ohm's law. |  |
|  |

Double-click on the Calculate JButton—this should take you in the Source code calculateJButtonActionPerformed method. Delete the comment line and type the following code:

|  |
| --- |
| // Calculate the current given voltage and resistance  DecimalFormat numberFormatter = new DecimalFormat("#,##0.0##");  float volts, resistance, current;  volts = Float.parseFloat(voltsJTextField.getText());  resistance = Float.parseFloat(resistanceJTextField.getText());  current = volts/resistance;  currentJTextField.setText(numberFormatter.format(current)); |

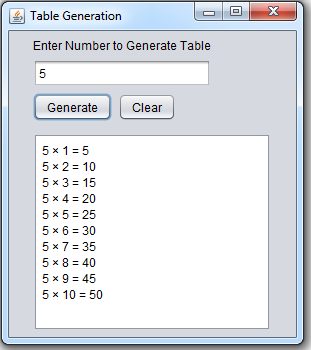
Double-click on the Clear button and add the following code:

|  |
| --- |
| // Clear all text fields to reset the form  voltsJTextField.setText("");  resistanceJTextField.setText("");  currentJTextField.setText("");  voltsJTextField.requestFocus(); |

Double-click on the Quit button and add the following code:

|  |
| --- |
| // Terminate the application  System.exit(0); |

1. Create a GUI application that prints the multiplication table of any given number.



Double-click on the Generate button and add the following code:

|  |
| --- |
| private void btnGenerateActionPerformed (java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  int base, i, n;  base = Integer.parseInt(txtTableValue.getText());  txtAreaTable.setText("");  for (i=1; i<=10; i++) {  n = base\*i;  txtAreaTable.append(base + " × " + i + " = " + n + "\n"); }  } |

Double-click on the Clear button and add the following code:

|  |
| --- |
| private void btnClearActionPerformed(java.awt.event.ActionEvent evt){  // TODO add your handling code here:  txtTableValue.setText("");  txtAreaTable.setText("");  } |