```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Calculator extends JFrame implements ActionListener {
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	JPanel[] row = new JPanel[5];
	String[] buttonString = { "7", "8", "9", "+", "4", "5", "6", "-", "1", "2", "3", "*", ".", "/", "C", "√",
"+/-",
				"=", "0", "sin", "cos", "tan", "sec", "csc", "cot", "log10", "x^2" };
	JButton[] button = new JButton[buttonString.length]; //Changed to make adding buttons easier
	int[] dimW = { 400, 45, 100, 90 }; // Changed for appearance's sake
	int[] dimH = { 35, 40 };
	Dimension displayDimension = new Dimension(dimW[0], dimH[0]);
	Dimension regularDimension = new Dimension(dimW[1], dimH[1]);
	Dimension rColumnDimension = new Dimension(dimW[2], dimH[1]);
	Dimension zeroButDimension = new Dimension(dimW[3], dimH[1]);
	boolean[] function = new boolean[4];
	double[] temporary = { 0, 0 };
	JTextArea display = new JTextArea(1, 20);
	Font font = new Font("Times new Roman", Font.BOLD, 14);

	Calculator() {
		super("Calculator");
		setDesign();
		setSize(500, 250);
		setResizable(false);
		setDefaultCloseOperation(EXIT_ON_CLOSE);
		GridLayout grid = new GridLayout(5, 7);
		setLayout(grid);

		for (int i = 0; i < 4; i++)
			function[i] = false;

		FlowLayout f1 = new FlowLayout(FlowLayout.CENTER);
		FlowLayout f2 = new FlowLayout(FlowLayout.CENTER, 1, 1);
		for (int i = 0; i < 5; i++)
			row[i] = new JPanel();
		row[0].setLayout(f1);
		for (int i = 1; i < 5; i++)
```

```
        row[i].setLayout(f2);

for (int i = 0; i < buttonString.length; i++) { //Changed to make adding buttons easier
        button[i] = new JButton();
        button[i].setText(buttonString[i]);
        button[i].setFont(font);
        button[i].addActionListener(this);
}
font = new Font("Times new Roman", Font.ITALIC, 14);
display.setFont(font);
display.setEditable(false);
display.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
display.setPreferredSize(displayDimension);
for (int i = 0; i < 14; i++)
        button[i].setPreferredSize(regularDimension);
for (int i = 14; i < 18; i++)
        button[i].setPreferredSize(rColumnDimension);
button[18].setPreferredSize(zeroButDimension);
for (int i = 18; i < 27; i++)
        button[i].setPreferredSize(zeroButDimension);
row[0].add(display);
add(row[0]);

for (int i = 0; i < 4; i++)
        row[1].add(button[i]);
row[1].add(button[14]);
row[1].add(button[19]);
row[1].add(button[20]);
add(row[1]);

for (int i = 4; i < 8; i++)
        row[2].add(button[i]);
row[2].add(button[15]);
row[2].add(button[21]);
row[2].add(button[22]);
add(row[2]);

for (int i = 8; i < 12; i++)
        row[3].add(button[i]);
row[3].add(button[16]);
row[3].add(button[23]);
row[3].add(button[24]);
add(row[3]);
```

```
		row[4].add(button[18]);
		for (int i = 12; i < 14; i++)
			row[4].add(button[i]);
		row[4].add(button[17]);
		row[4].add(button[25]);
		row[4].add(button[26]);
		add(row[4]);

		setVisible(true);
	}

	public void clear() {
		try {
			display.setText("");
			for (int i = 0; i < 4; i++)
				function[i] = false;
			for (int i = 0; i < 2; i++)
				temporary[i] = 0;
		} catch (NullPointerException e) {
		}
	}

	public void getSqrt() {
		try {
			double value = Math.sqrt(Double.parseDouble(display.getText()));
			display.setText(Double.toString(value));
		} catch (NumberFormatException e) {
		}
	}

	public void getSquare() {
		try {
			double value = Double.parseDouble(display.getText()) *
Double.parseDouble(display.getText());
			display.setText(Double.toString(value));
		} catch (NumberFormatException e) {
		}
	}


	public void getLog10() {
		try {
```

```
                    double value = Math.log10(Double.parseDouble(display.getText()));
                    display.setText(Double.toString(value));
                } catch (NumberFormatException e) {
                }
        }

        //All of the following trigonometry functions assume that the number given is in degrees.
        public void getSin() {
                try {
                    double value =
Math.sin(Math.toRadians(Double.parseDouble(display.getText())));
                    display.setText(Double.toString(value));
                } catch (NumberFormatException e) {
                }
        }

        public void getCos() {
                try {
                    double value =
Math.cos(Math.toRadians(Double.parseDouble(display.getText())));
                    display.setText(Double.toString(value));
                } catch (NumberFormatException e) {
                }
        }

        public void getTan() {
                try {
                    double value =
Math.tan(Math.toRadians(Double.parseDouble(display.getText())));
                    display.setText(Double.toString(value));
                } catch (NumberFormatException e) {
                }
        }

        public void getSec() {
                try {
                    double value = 1.0 /
Math.cos(Math.toRadians(Double.parseDouble(display.getText())));
                    display.setText(Double.toString(value));
                } catch (NumberFormatException e) {
                }
        }
```

```
	public void getCsc() {
		try {
			double value = 1.0 /
Math.sin(Math.toRadians(Double.parseDouble(display.getText())));
			display.setText(Double.toString(value));
		} catch (NumberFormatException e) {
		}
	}

	public void getCot() {
		try {
			double value = 1.0 /
Math.tan(Math.toRadians(Double.parseDouble(display.getText())));
			display.setText(Double.toString(value));
		} catch (NumberFormatException e) {
		}
	}


	public void getPosNeg() {
		try {
			double value = Double.parseDouble(display.getText());
			if (value != 0) {
				value = value * (-1);
				display.setText(Double.toString(value));
			} else {
			}
		} catch (NumberFormatException e) {
		}
	}

	public void getResult() {
		double result = 0;
		temporary[1] = Double.parseDouble(display.getText());
		String temp0 = Double.toString(temporary[0]);
		String temp1 = Double.toString(temporary[1]);
		try {
			if (temp0.contains("-")) {
				String[] temp00 = temp0.split("-", 2);
				temporary[0] = (Double.parseDouble(temp00[1]) * -1);
			}
			if (temp1.contains("-")) {
				String[] temp11 = temp1.split("-", 2);
```

```
                        temporary[1] = (Double.parseDouble(temp11[1]) * -1);
                    }
                } catch (ArrayIndexOutOfBoundsException e) {
                }
                try {
                    if (function[2] == true)
                        result = temporary[0] * temporary[1];
                    else if (function[3] == true)
                        result = temporary[0] / temporary[1];
                    else if (function[0] == true)
                        result = temporary[0] + temporary[1];
                    else if (function[1] == true)
                        result = temporary[0] - temporary[1];
                    display.setText(Double.toString(result));
                    for (int i = 0; i < 4; i++)
                        function[i] = false;
                } catch (NumberFormatException e) {
                }
        }

        public final void setDesign() {
            try {

UIManager.setLookAndFeel("com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel");
            } catch (Exception e) {
            }
        }

        @Override
        public void actionPerformed(ActionEvent ae) {
            if (ae.getSource() == button[0])
                display.append("7");
            if (ae.getSource() == button[1])
                display.append("8");
            if (ae.getSource() == button[2])
                display.append("9");
            if (ae.getSource() == button[3]) {
                // add function[0]
                temporary[0] = Double.parseDouble(display.getText());
                function[0] = true;
                display.setText("");
            }
            if (ae.getSource() == button[4])
```

```
            display.append("4");
if (ae.getSource() == button[5])
            display.append("5");
if (ae.getSource() == button[6])
            display.append("6");
if (ae.getSource() == button[7]) {
            // subtract function[1]
            temporary[0] = Double.parseDouble(display.getText());
            function[1] = true;
            display.setText("");
}
if (ae.getSource() == button[8])
            display.append("1");
if (ae.getSource() == button[9])
            display.append("2");
if (ae.getSource() == button[10])
            display.append("3");
if (ae.getSource() == button[11]) {
            // multiply function[2]
            temporary[0] = Double.parseDouble(display.getText());
            function[2] = true;
            display.setText("");
}
if (ae.getSource() == button[12])
            display.append(".");
if (ae.getSource() == button[13]) {
            // divide function[3]
            temporary[0] = Double.parseDouble(display.getText());
            function[3] = true;
            display.setText("");
}
if (ae.getSource() == button[14])
            clear();
if (ae.getSource() == button[15])
            getSqrt();
if (ae.getSource() == button[16])
            getPosNeg();
if (ae.getSource() == button[17])
            getResult();
if (ae.getSource() == button[18])
            display.append("0");
if (ae.getSource() == button[19])
            getSin();
```

```
                if (ae.getSource() == button[20])
                        getCos();
                if (ae.getSource() == button[21])
                        getTan();
                if (ae.getSource() == button[22])
                        getSec();
                if (ae.getSource() == button[23])
                        getCsc();
                if (ae.getSource() == button[24])
                        getCot();
                if (ae.getSource() == button[25])
                        getLog10();
                if (ae.getSource() == button[26])
                        getSquare();

        }

        public static void main(String[] arguments) {
                @SuppressWarnings("unused")
                Calculator c = new Calculator();
        }
}
/*
 * http://www.dreamincode.net/forums/topic/321933-creating-a-calculator-using-
 * jframe/
 */
lic class Calculator extends JFrame implements ActionListener {
 /**
  *
  */
 private st
 JPanel[] r
 String[] b
          "=
 JButton[]
 int[] dimW
 int[] dimH
 Dimension
 Dimension
 Dimension
 Dimension
 boolean[]
 double[] t
 JTextArea
 Font font

 Calculator() {
     super("Calculator");
     setDesign();
```