Java Institute for Advanced Technology

UNIT NAME : SOFTWARE APPLICATION DEVELOPMENT

UNIT ID : H7E1 04

ASSIGNMENT ID : H7E1 04/AS/01

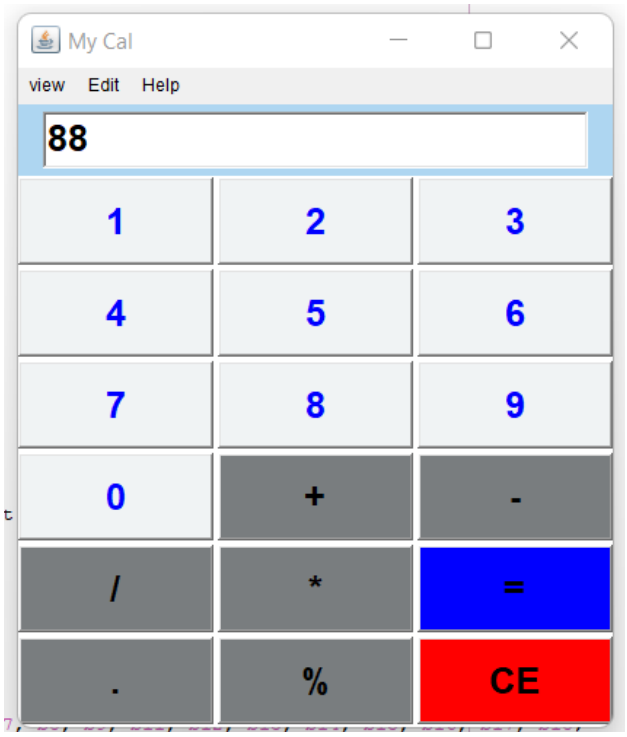
Name : Jayaweerage Namindu Wathsala Fernando

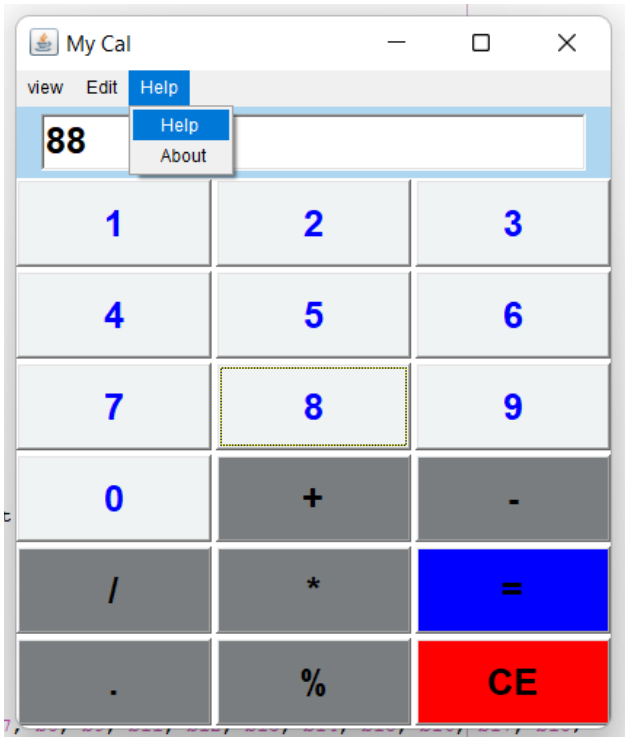
SQA ID : 207981559

NIC : 200133000408

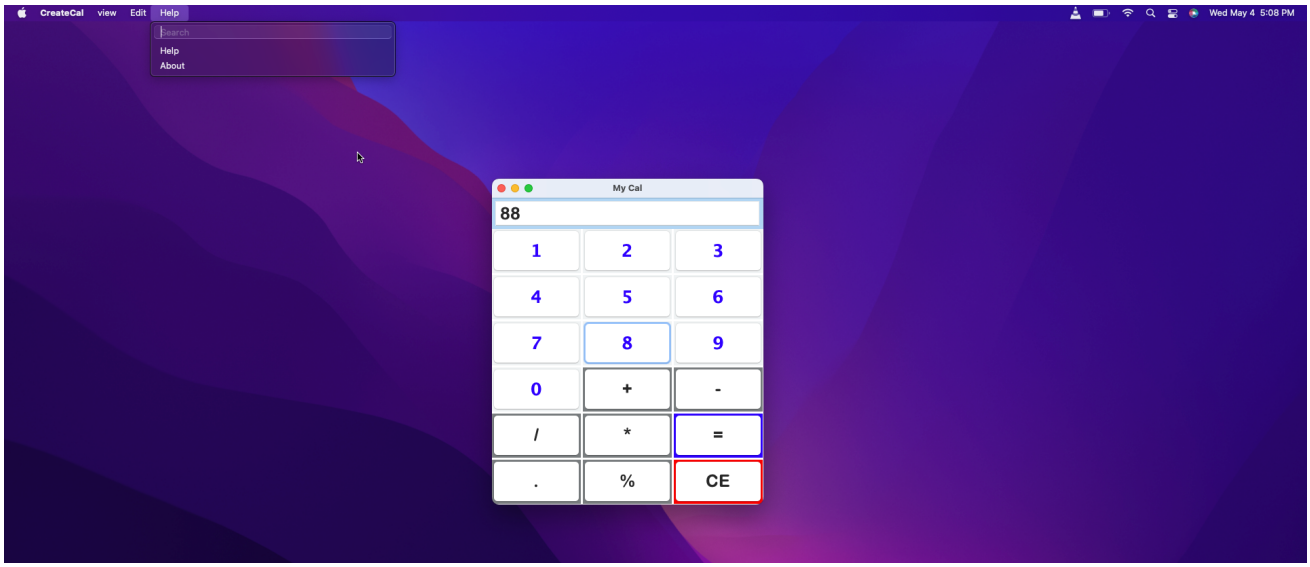
BRANCH : Colombo

On Windows





On Mac OS



```
package cal;

import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Color;
import java.awt.Font;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.Menu;
import java.awt.MenuBar;
import java.awt.MenuItem;
import java.awt.Panel;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JLabel;

class callisner extends WindowAdapter {

   @Override
   public void windowClosing(WindowEvent e) {
      System.exit(0);
   }

}

public class cal implements ActionListener {
```

```
    Button b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, b11, b12, b13, b14, b15, b16,
b17, b18;
    TextField tf;
    String fv, sv, op;
    double fdv, sdv, tot;

    cal() {

        Frame f1 = new Frame();
        Color lb = new Color(174, 214, 241);

        f1.setBackground(lb);
        f1.addWindowListener(new callisner());
        f1.setBounds(500, 250, 400, 480);
        f1.setTitle("My Cal");

        f1.setVisible(true);

        MenuBar mbar = new MenuBar();
        MenuItem ma1 = new MenuItem("Standard");
        MenuItem ma2 = new MenuItem("Scientific");
        MenuItem ma3 = new MenuItem("Copy");
        MenuItem ma4 = new MenuItem("Help");
        MenuItem ma5 = new MenuItem("About");
        Menu m1 = new Menu("view");
        m1.add(ma1);
        m1.add(ma2);
        Menu m2 = new Menu("Edit");
```

```
m2.add(ma3);
Menu m3 = new Menu("Help");
m3.add(ma4);
m3.add(ma5);

 mbar.add(m1);
 mbar.add(m2);
 mbar.add(m3);

f1.setMenuBar(mbar);

Panel p1 = new Panel();
Panel p2 = new Panel();
GridLayout g1 = new GridLayout(6, 5, 3, 3);

b0 = new Button("0");
b1 = new Button("1");
b2 = new Button("2");
b3 = new Button("3");
b4 = new Button("4");
b5 = new Button("5");
b6 = new Button("6");
b7 = new Button("7");
b8 = new Button("8");
b9 = new Button("9");

b11 = new Button("=");
b12 = new Button("+");
b13 = new Button("-");
```

```
b14 = new Button("/");
b15 = new Button("CE");
b16 = new Button("*");
b17 = new Button(".");
b18 = new Button("%");

Color lg = new Color(121, 125, 127);
Color lw = new Color(240, 243, 244);
b0.setBackground(lw);
b1.setBackground(lw);
b2.setBackground(lw);
b3.setBackground(lw);
b4.setBackground(lw);
b5.setBackground(lw);
b6.setBackground(lw);
b7.setBackground(lw);
b8.setBackground(lw);
b9.setBackground(lw);
b11.setBackground(Color.blue);
b13.setBackground(lg);
b12.setBackground(lg);
b14.setBackground(lg);
b15.setBackground(Color.red);
b16.setBackground(lg);
b17.setBackground(lg);
b18.setBackground(lg);

Font fon1 = new Font("Calibri", Font.BOLD, 25);
Font fon2 = new Font("arial rounded MT BOLD", Font.BOLD, 25);
```

```
b0.setFont(fon1);
b1.setFont(fon1);
b2.setFont(fon1);
b3.setFont(fon1);
b4.setFont(fon1);
b5.setFont(fon1);
b6.setFont(fon1);
b7.setFont(fon1);
b8.setFont(fon1);
b9.setFont(fon1);

b11.setFont(fon2);
b12.setFont(fon2);
b13.setFont(fon2);
b14.setFont(fon2);
b15.setFont(fon2);
b16.setFont(fon2);
b17.setFont(fon2);
b18.setFont(fon2);

b0.setForeground(Color.BLUE);
b1.setForeground(Color.BLUE);
b2.setForeground(Color.BLUE);
b3.setForeground(Color.BLUE);
b4.setForeground(Color.BLUE);
b5.setForeground(Color.BLUE);
b6.setForeground(Color.BLUE);
b7.setForeground(Color.BLUE);
```

```
b8.setForeground(Color.BLUE);
b9.setForeground(Color.BLUE);

tf = new TextField(25);
tf.setFont(fon2);

p2.add(b1);
p2.add(b2);
p2.add(b3);
p2.add(b4);
p2.add(b5);
p2.add(b6);
p2.add(b7);
p2.add(b8);
p2.add(b9);
p2.add(b0);

p2.add(b12);
p2.add(b13);
p2.add(b14);
p2.add(b16);
p2.add(b11);
p2.add(b17);
p2.add(b18);
p2.add(b15);

p1.add(tf);

f1.add(p1, BorderLayout.NORTH);
```

```
    f1.add(p2, BorderLayout.CENTER);
    p2.setLayout(g1);
    p2.setBackground(Color.WHITE);

    b0.addActionListener(this);
    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);
    b5.addActionListener(this);
    b6.addActionListener(this);
    b7.addActionListener(this);
    b8.addActionListener(this);
    b9.addActionListener(this);
    b11.addActionListener(this);
    b12.addActionListener(this);
    b13.addActionListener(this);
    b14.addActionListener(this);
    b15.addActionListener(this);
    b16.addActionListener(this);
    b17.addActionListener(this);
    b18.addActionListener(this);

}

@Override
public void actionPerformed(ActionEvent e) {

    Object o = e.getSource();
```

```
if (o.equals(b0)) {
    tf.setText(tf.getText() + b0.getLabel());
} else if (o.equals(b1)) {
    tf.setText(tf.getText() + b1.getLabel());
} else if (o.equals(b2)) {
    tf.setText(tf.getText() + b2.getLabel());
} else if (o.equals(b3)) {
    tf.setText(tf.getText() + b3.getLabel());
} else if (o.equals(b4)) {
    tf.setText(tf.getText() + b4.getLabel());
} else if (o.equals(b5)) {
    tf.setText(tf.getText() + b5.getLabel());
} else if (o.equals(b6)) {
    tf.setText(tf.getText() + b6.getLabel());
```

```
} else if (o.equals(b7)) {

    tf.setText(tf.getText() + b7.getLabel());

} else if (o.equals(b8)) {

    tf.setText(tf.getText() + b8.getLabel());

} else if (o.equals(b9)) {

    tf.setText(tf.getText() + b9.getLabel());

} else if (o.equals(b17)) {

    tf.setText(tf.getText() + b17.getLabel());

} else if (o.equals(b12)) {
    fv = tf.getText();
    tf.setText("");
    op = b12.getLabel();

} else if (o.equals(b13)) {
    fv = tf.getText();
    tf.setText("");
    op = b13.getLabel();

} else if (o.equals(b14)) {
    fv = tf.getText();
    tf.setText("");
```

```
    op = b14.getLabel();

} else if (o.equals(b15)) {
    fv = tf.getText();
    tf.setText("");
    op = b15.getLabel();

} else if (o.equals(b16)) {
    fv = tf.getText();
    tf.setText("");
    op = b16.getLabel();

} else if (o.equals(b18)) {
    fdv = Double.parseDouble(tf.getText());
    tf.setText(Double.toString(fdv / 100));

} else if (o.equals(b11)) {
    sv = tf.getText();
    fdv = Double.parseDouble(fv);
    sdv = Double.parseDouble(sv);

    if (op.equals("+")) {
        tot = fdv + sdv;
        tf.setText(tot + "");
    } else if (op.equals("-")) {
        tot = fdv - sdv;
        tf.setText(tot + "");
    } else if (op.equals("*")) {
        tot = fdv * sdv;
```

```
            tf.setText(tot + "");
         } else if (op.equals("/")) {
            tot = fdv / sdv;
            tf.setText(tot + "");
         }

      }

   }

}

class CreateCal {

   public static void main(String[] args) {
      new cal();
   }

}
```