

**GROUP A**  
**ASSIGNMENT NO-5**

## **1 Problem Statement:-**

A Mobile App for Calculator having Trigonometry functionality is to be designed and tested. The data storage uses 1.textiles, 2.XML Use latest open source software modeling, Designing and testing tool/ Scrum-it. Implement the design using HTML-5/Scala/ Python/Java/C++/Rubi on Rails. Perform Positive and Negative testing.

## **2 Theory :**

A software calculator is a calculator that has been implemented as a computer program, rather than as a physical hardware device. They are among the simpler interactive software tools, and, as such, they. Provide operations for the user to select one at a time. Can be used to perform any process that consists of a sequence of steps each of which applies one of these operations. Have no purpose other than these processes, because the operations are the sole, or at least the primary, features of the calculator, rather than being secondary features that support other functionality that is not normally known simply as calculation.

### **JAVA Swing:**

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) an API for providing a graphical user interface (GUI) for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists. Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element. Swing is currently in the process of being replaced by JavaFX.

Swing is a platform-independent, Model-View-Controller GUI framework for Java, which follows a single-threaded programming model. Additionally, this framework provides a layer of abstraction between the code structure and graphic presentation of a Swing-based GUI.

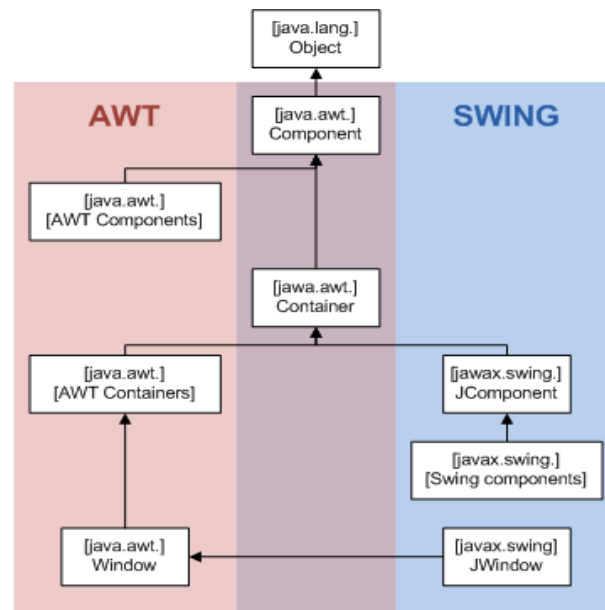


Figure.1

Swing is platform-independent because it is completely written in Java. Complete documentation for all Swing classes can be found in the Java API Guide.

### Extensible:

Swing is a highly modular-based architecture, which allows for the "plugging" of various custom implementations of specified framework interfaces: Users can provide their own custom implementation(s) of these components to override the default implementations using Java's inheritance mechanism.

Swing is a component-based framework, whose components are all ultimately derived from the javax.swing.JComponent class. Swing objects asynchronously fire events, have bound properties, and respond to a documented set of methods specific to the component. Swing components are Java Beans components, compliant with the Java Beans Component Architecture specifications.

### Customizable:

Given the programmatic rendering model of the Swing framework, fine control over the details of rendering of a component is possible. As a general pattern, the visual representation of a Swing component is a composition of a standard set of elements, such as a border, inset, decorations, and other properties. Typically, users will programmatically customize a standard Swing component (such as a JTable) by assigning specific borders, colors, backgrounds, opacities, etc. The core component will then use these properties to render itself. However, it is also completely possible to create unique GUI controls with highly customized visual representation.

**Configurable:**

Swing's heavy reliance on runtime mechanisms and indirect composition patterns allows it to respond at run time to fundamental changes in its settings. For example, a Swing-based application is capable of hot swapping its user-interface during runtime. Furthermore, users can provide their own look and feel implementation, which allows for uniform changes in the look and feel of existing Swing applications without any programmatic change to the application code.

**Relationship to AWT:**

**AWT and Swing class hierarchy** Since early versions of Java, a portion of the Abstract Window Toolkit (AWT) has provided platform-independent APIs for user interface components. In AWT, each component is rendered and controlled by a native peer component specific to the underlying windowing system.

By contrast, Swing components are often described as lightweight because they do not require allocation of native resources in the operating system's windowing toolkit. The AWT components are referred to as heavyweight components.

Much of the Swing API is generally a complementary extension of the AWT rather than a direct replacement. In fact, every Swing lightweight interface ultimately exists within an AWT heavyweight component because all of the top-level components in Swing (JApplet, JDialog, JFrame, and JWindow) extend an AWT top-level container. Prior to Java 6 Update 10, the use of both lightweight and heavyweight components within the same window was generally discouraged due to Z-order incompatibilities. However, later versions of Java have fixed these issues, and both Swing and AWT components can now be used in one GUI without Z-order issues.

The core rendering functionality used by Swing to draw its lightweight components is provided by Java 2D, another part of JFC.

**Relationship to SWT:**

The Standard Widget Toolkit (SWT) is a competing toolkit originally developed by IBM and now maintained by the Eclipse community. SWT's implementation has more in common with the heavyweight components of AWT. This confers benefits such as more accurate fidelity with the underlying native windowing toolkit, at the cost of an increased exposure to the native platform in the programming model.

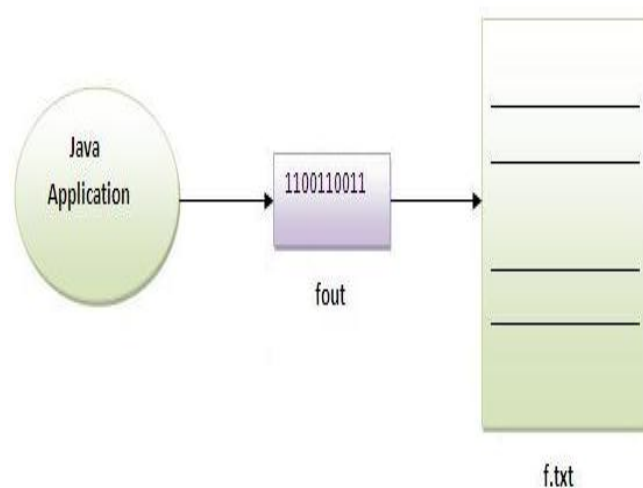
There has been significant debate and speculation about the performance of SWT versus Swing; some hinted that SWT's heavy dependence on JNI would make it slower when the GUI component and Java need to communicate data, but faster at rendering when the data model has been loaded into the GUI, but this has not been

confirmed either way. A fairly thorough set of benchmarks in 2005 concluded that neither Swing nor SWT clearly outperformed the other in the general case.

### **FileInputStream and FileOutputStream (File Handling)**

In Java, `FileInputStream` and `FileOutputStream` classes are used to read and write data in file. In another words, they are used for file handling in java. **Java FileOutputStream class** Java `FileOutputStream` is an output stream for writing data to a file.

If you have to write primitive values then use `FileOutputStream`. Instead, for character-oriented data, prefer `FileWriter`. But you can write byte-oriented as well as character-oriented data.



**File output stream**

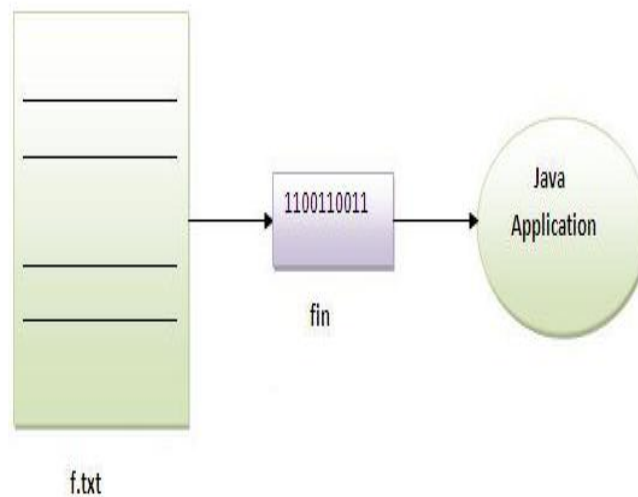
### **Java FileInputStream class:**

Java `FileInputStream` class obtains input bytes from a file. It is used for reading streams of raw bytes such as image data. For reading streams of characters, consider using `FileReader`.

It should be used to read byte-oriented data for example to read image, audio, video etc.

### **Positive and Negative Testing:**

A test case, in software engineering, is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle.



File output stream

In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written - when they are usually collected into test suites

### Positive Testing:

Positive Testing is testing process where the system validated against the valid input data. In this testing tester always check for only valid set of values and check if a application behaves as expected with its expected inputs. The main intention of this testing is to check whether software application not showing error when not supposed to & showing error when supposed to. Such testing is to be carried out keeping positive point of view & only execute the positive scenario. Positive Testing always tries to prove that a given product and project always meets the requirements and specifications. Under Positive testing is test the normal day to day life scenarios and check the expected behavior of application.

### Negative Testing:

Negative Testing is testing process where the system validated against the invalid input data. A negative test checks if a application behaves as expected with its negative inputs. The main intention of this testing is to check whether software application not showing error when supposed to & showing error when not supposed to. Such testing is to be carried out keeping negative point of view & only execute the test cases for only invalid set of input data.

Negative testing is a testing process to identify the inputs where system is not designed

or un-handled inputs by providing different invalid. The main reason behind Negative testing is to check the stability of the software application against the influences of different variety of incorrect validation data set.

### 3 Algorithm :

- 1.)Start the Application.
- 2.)Enter the first number i.e operands.
- 3.)Enter the operator.
- 4.)Enter the second number i.e operands.
- 5.)Result will be generated.
- 6.)If user want to store this operation and result it will get store in history.
- 7.)Close the application.

### 4 Mathematical model:

Let S be a calculator system:

$$S = \{Q, \sum, O, \delta, I, F(me)\}$$

Where ,

- 1.Q is set of states that performs operations on system(S)
2. $\sum$  is input given by user to select state(q)
- 3.O is output generated by states (Q)
4. $\delta$  is transition function which maps  $Q^* \sum \rightarrow Q$
- 5.I is input values x given by user

$$1).\delta:Q[\sum x=0] \rightarrow q_0$$

Start Application

Enter the first Number( $N_1$ )

$$N \in \{n_1, n_2, n_3, n_4, \dots, n_n\}$$

$$2).\delta:Q[\sum x=1] \rightarrow q_1$$

Enter the operator(O)

$$O \in \{o_1, o_2, o_3, o_4, \dots, o_n\}$$

Example-(+, -, \*, /)

$$3).\delta:Q[\sum x=2] \rightarrow q_2$$

Enter the second Number( $N_2$ )

$$N_2 \in \{n_1, n_2, n_3, n_4, \dots, n_n\}$$

4). $\delta:Q[\sum x=3] \rightarrow q_3$

At this step we will perform operation

$$R = N_1 \text{ O } N_2$$

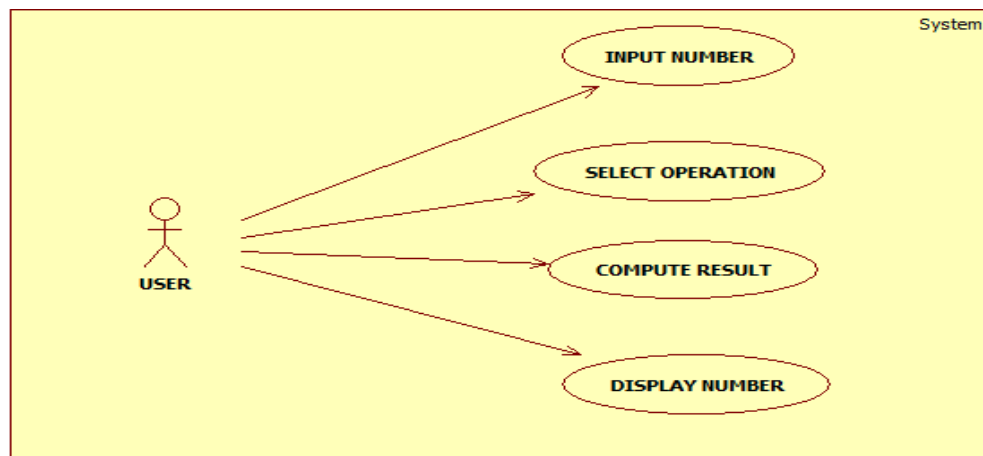
where R is result

5). $\delta:Q[\sum x=4] \rightarrow q_4$

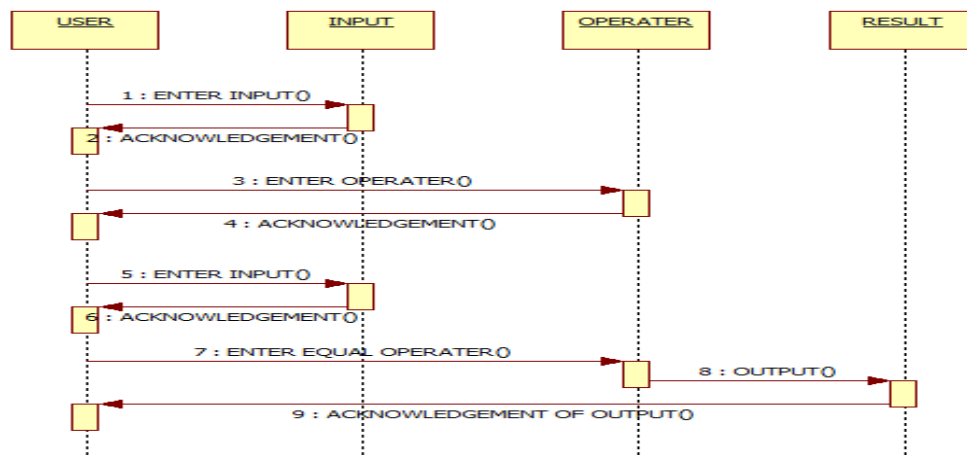
Close application

## 5 UML Diagram

Use Case Diagram



Sequence Diagram



## 6 Program:-

```
/*
 * To change this license header, choose License Headers
 * in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package calculator;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;

/**
 *
 * @author abhi
 */
public class Calc_Ui extends javax.swing.JFrame {

    Double firstNumber, secondNumber;
    char operation;

    public Calc_Ui() {
        initComponents();
        setResizable(false);
        setTitle(" Calculator");
    }

    /**
     * This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of
     * this method is always
     * regenerated by the Form Editor.
     */
}
```



```
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
//GEN-BEGIN: initComponents
private void initComponents() {

    scrn = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
    jButton4 = new javax.swing.JButton();
    jButton5 = new javax.swing.JButton();
    jButton6 = new javax.swing.JButton();
    jButton7 = new javax.swing.JButton();
    jButton8 = new javax.swing.JButton();
    jButton9 = new javax.swing.JButton();
    jButton0 = new javax.swing.JButton();
    jButton_dot = new javax.swing.JButton();
    jButton_sign = new javax.swing.JButton();
    jButton_clear = new javax.swing.JButton();
    jButton_add = new javax.swing.JButton();
    jButton_sub = new javax.swing.JButton();
    jButton_mul = new javax.swing.JButton();
    jButton_div = new javax.swing.JButton();
    jButton_sin = new javax.swing.JButton();
    jButton_cos = new javax.swing.JButton();
    jButton_sqrt = new javax.swing.JButton();
    jButton_sqrt1 = new javax.swing.JButton();
    jButton_equals = new javax.swing.JButton();
    oper = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    scrn.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
    scrn.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            scrnActionPerformed(evt);
        }
    });
}
```

```
jButton1.setText("1");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
```

```
jButton2.setText("2");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
```

```
jButton3.setText("3");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
```

```
jButton4.setText("4");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});
```

```
jButton5.setText("5");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});
```

```
jButton6.setText("6");
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
jButton6ActionPerformed ( evt );  
}  
});
```

```
jButton7.setText ( " 7" );  
jButton7.addActionListener ( new java.awt.event.ActionListener () {  
    public void actionPerformed ( java.awt.event.ActionEvent evt ) {  
        jButton7ActionPerformed ( evt );  
    }  
});
```

```
jButton8.setText ( " 8" );  
jButton8.addActionListener ( new java.awt.event.ActionListener () {  
    public void actionPerformed ( java.awt.event.ActionEvent evt ) {  
        jButton8ActionPerformed ( evt );  
    }  
});
```

```
jButton9.setText ( " 9" );  
jButton9.addActionListener ( new java.awt.event.ActionListener () {  
    public void actionPerformed ( java.awt.event.ActionEvent evt ) {  
        jButton9ActionPerformed ( evt );  
    }  
});
```

```
jButton0.setText ( " 0" );  
jButton0.addActionListener ( new java.awt.event.ActionListener () {  
    public void actionPerformed ( java.awt.event.ActionEvent evt ) {  
        jButton0ActionPerformed ( evt );  
    }  
});
```

```
jButton_dot.setText ( "." );  
jButton_dot.addActionListener ( new java.awt.event.ActionListener () {  
    public void actionPerformed ( java.awt.event.ActionEvent evt ) {  
        jButton_dotActionPerformed ( evt );  
    }  
});
```

```
jButton_sign.setText("+/-");
jButton_sign.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton_signActionPerformed(evt);
    }
});

jButton_clear.setText(" Clear ");
jButton_clear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton_clearActionPerformed(evt);
    }
});

jButton_add.setText("+");
jButton_add.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton_addActionPerformed(evt);
    }
});

jButton_sub.setText("-");
jButton_sub.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton_subActionPerformed(evt);
    }
});

jButton_mul.setText("*");
jButton_mul.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton_mulActionPerformed(evt);
    }
});

jButton_div.setText("/");
jButton_div.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton_divActionPerformed(evt);
    }
});
```

```
}  
});
```

```
jButton_sin.setText(" sin ");  
jButton_sin.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton_sinActionPerformed(evt);  
    }  
});
```

```
jButton_cos.setText(" cos ");  
jButton_cos.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton_cosActionPerformed(evt);  
    }  
});
```

```
jButton_sqrt.setText(" sq-root ");  
jButton_sqrt.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton_sqrtActionPerformed(evt);  
    }  
});
```

```
jButton_sqrt1.setText(" History ");  
jButton_sqrt1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton_sqrt1ActionPerformed(evt);  
    }  
});
```

```
jButton_equals.setText("=");  
jButton_equals.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton_equalsActionPerformed(evt);  
    }  
});
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout
```

```
(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(scrn)
.addGroup(layout.createSequentialGroup()
.addContainerGap()
.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout
.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout
.Alignment.LEADING, false)
.addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout
.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addComponent(jButton4,
javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(10, 10, 10)
.addComponent(jButton5,
javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.RELATED)
.addComponent(jButton6, javax.swing.
GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(layout.createSequentialGroup()
.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addComponent(jButton3 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(layout.createSequentialGroup())
.addComponent(jButton7 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jButton8 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jButton9 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(jButton_sqrt , javax.swing.GroupLayout.PREFERRED_SIZE,
130, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addGap(18, 18, 18)
.addComponent(jButton_sqrt1 , javax.swing.GroupLayout.PREFERRED_SIZE,
154, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(layout.createSequentialGroup())
.addGap(98, 98, 98)
.addComponent(jButton_sub , javax.swing.GroupLayout.PREFERRED_SIZE,
75, javax.swing.GroupLayout.PREFERRED_SIZE))))
.addGroup(layout.createSequentialGroup())
.addComponent(jButton0 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement
.UNRELATED)
.addComponent(jButton_dot , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jButton_sign , javax.swing.GroupLayout.PREFERRED_SIZE,
49, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
```

```
.addComponent(jButton_mul , javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jButton_cos , javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGap(6, 6, 6)
.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jButton_sin , javax.swing.GroupLayout
.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_div , javax.swing.GroupLayout
.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGroup(layout.createSequentialGroup())
.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.TRAILING, false)
.addComponent(jButton_add , javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing
.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jButton_equals , javax.swing.GroupLayout
.Alignment.LEADING, javax.swing.GroupLayout
.DEFAULT_SIZE, 75, Short.MAX_VALUE))
.addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.RELATED, javax.swing
.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jButton_clear , javax.swing.GroupLayout
.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE))))
.addContainerGap(javax.swing.GroupLayout
.DEFAULT_SIZE, Short.MAX_VALUE))
.addComponent(oper , javax.swing.GroupLayout
.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing
.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addComponent(oper , javax.swing.GroupLayout
.DEFAULT_SIZE, 32, Short.MAX_VALUE)
.addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.RELATED)
```



```
.addComponent(scrn , javax.swing.GroupLayout
.PREFERRED_SIZE, 53, javax.swing.GroupLayout
.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax
.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton_sqrt , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_sqrt1 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
.addComponent(jButton7 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton8 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton9 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_cos , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_sin , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
.addComponent(jButton4 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton5 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton6 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_mul , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_div , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
```

```
. Alignment.BASELINE)
.addComponent(jButton1 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton2 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton3 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_sub , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_add , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement
.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
.addComponent(jButton0 , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_dot , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_sign , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_clear , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton_equals , javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(22, 22, 22))
);

pack();
} // </editor-fold> //GEN-END: initComponents

private void scrnActionPerformed
(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_scrnActionPerformed
// TODO add your handling code here:
} //GEN-LAST:event_scrnActionPerformed

private void jButton8ActionPerformed
(java.awt.event.ActionEvent evt)
```

```
//GEN-FIRST:event_jButton8ActionPerformed
// TODO add your handling code here:
String temp=scrn.getText();
String num=jButton8.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton8ActionPerformed

private void jButton9ActionPerformed
(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_jButton9ActionPerformed
    // TODO add your handling code here:
    String temp=scrn.getText();
    String num=jButton9.getText();
    scrn.setText(temp+num);
}//GEN-LAST:event_jButton9ActionPerformed

private void jButton6ActionPerformed
(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_jButton6ActionPerformed
    // TODO add your handling code here:
    String temp=scrn.getText();
    String num=jButton6.getText();
    scrn.setText(temp+num);
}//GEN-LAST:event_jButton6ActionPerformed

private void jButton_clearActionPerformed(java.awt.event
.ActionEvent evt)
{
    //GEN-FIRST:event_jButton_clearActionPerformed
    // TODO add your handling code here:
    scrn.setText("");
    operation='';
    oper.setText("");
}//GEN-LAST:event_jButton_clearActionPerformed

private void jButton_addActionPerformed
(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_jButton_addActionPerformed
    // TODO add your handling code here:
    firstNumber=Double.parseDouble(scrn.getText());
```

```
operation='+';
scrn.setText("");
} //GEN-LAST:event_jButton_addActionPerformed

private void jButton_subActionPerformed
(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton_subActionPerformed
// TODO add your handling code here:
firstNumber=Double.parseDouble(scrn.getText());
operation='-';
scrn.setText("");
} //GEN-LAST:event_jButton_subActionPerformed

private void jButton_mulActionPerformed
(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton_mulActionPerformed
// TODO add your handling code here:
firstNumber=Double.parseDouble(scrn.getText());
operation='*';
scrn.setText("");
} //GEN-LAST:event_jButton_mulActionPerformed

private void jButton_divActionPerformed
(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton_divActionPerformed
// TODO add your handling code here:
firstNumber=Double.parseDouble(scrn.getText());
operation='/';
scrn.setText("");
} //GEN-LAST:event_jButton_divActionPerformed

private void jButton_sinActionPerformed
(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton_sinActionPerformed
// TODO add your handling code here:
firstNumber=Double.parseDouble(scrn.getText());
double radians = Math.toRadians(firstNumber);
oper.setText("Sin("+firstNumber+")");
scrn.setText(Math.sin(radians)+"");
```

```
FileOutputStream fos;
try {
fos = new FileOutputStream("log.txt",true);
fos.write( (" Sin("+firstNumber+")"+"="+Math.cos(radians)+"\n")
.getBytes() );
} catch (FileNotFoundException ex) {
Logger.getLogger( Calc_Ui.class.getName()).log( Level.SEVERE, null , ex);
} catch (IOException ex) {
Logger.getLogger( Calc_Ui.class.getName()).log( Level.SEVERE, null , ex);
}

} //GEN-LAST:event_jButton_sinActionPerformed

private void jButton_cosActionPerformed
(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton_cosActionPerformed
FileOutputStream fos = null;
try {
// TODO add your handling code here:
firstNumber=Double.parseDouble( scrn.getText());
double radians = Math.toRadians( firstNumber );
oper.setText( " Cos("+firstNumber+")");
fos = new FileOutputStream("log.txt",true);
fos.write( (" Cos("+firstNumber+")"+"="+Math.cos(radians)+"\n")
.getBytes() );
scrn.setText(Math.cos(radians)+"");
} catch (FileNotFoundException ex) {
Logger.getLogger( Calc_Ui.class.getName()).log( Level.SEVERE, null , ex);
} catch (IOException ex) {
Logger.getLogger( Calc_Ui.class.getName()).log( Level.SEVERE, null , ex);
} finally {
try {
fos.close();
} catch (IOException ex) {
Logger.getLogger( Calc_Ui.class.getName()).log( Level.SEVERE, null , ex);
}
}
} //GEN-LAST:event_jButton_cosActionPerformed
```

```
private void jButton_sqrtActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton_sqrtActionPerformed
// TODO add your handling code here:
firstNumber=Double.parseDouble(scrn.getText());
oper.setText(" sqrt("+firstNumber+")");
scrn.setText(Math.sqrt(firstNumber)+"");

try {
FileOutputStream fos = new FileOutputStream("log.txt",true);
fos.write( (" sqrt("+firstNumber+")"+"="+Math.sqrt(firstNumber)+"\n")
.getBytes() );

} catch (FileNotFoundException ex) {
Logger.getLogger(Calc_Ui.class.getName()).log(Level.SEVERE, null, ex);
} catch (IOException ex) {
Logger.getLogger(Calc_Ui.class.getName()).log(Level.SEVERE, null, ex);
}

}//GEN-LAST:event_jButton_sqrtActionPerformed

private void jButton_sqrt1ActionPerformed
(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton_sqrt1ActionPerformed
// TODO add your handling code here:
History his =new History();
his.setVisible(true);

his.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

}//GEN-LAST:event_jButton_sqrt1ActionPerformed

private void jButton0ActionPerformed
(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton0ActionPerformed
String temp=scrn.getText();
String num=jButton0.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton0ActionPerformed
```

```
private void jButton1ActionPerformed
(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
// TODO add your handling code here:
String temp=scrn.getText();
String num=jButton1.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton1ActionPerformed
```

```
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton2ActionPerformed
// TODO add your handling code here:
String temp=scrn.getText();
String num=jButton2.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton2ActionPerformed
```

```
private void jButton3ActionPerformed
(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton3ActionPerformed
// TODO add your handling code here:
String temp=scrn.getText();
String num=jButton3.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton3ActionPerformed
```

```
private void jButton4ActionPerformed
(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton4ActionPerformed
// TODO add your handling code here:
String temp=scrn.getText();
String num=jButton4.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton4ActionPerformed
```

```
private void jButton5ActionPerformed
(java.awt.event.ActionEvent evt)
```

```
//GEN-FIRST:event_jButton5ActionPerformed
// TODO add your handling code here:
String temp=scrn.getText();
String num=jButton5.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton5ActionPerformed

private void jButton7ActionPerformed
(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton7ActionPerformed
// TODO add your handling code here:
String temp=scrn.getText();
String num=jButton7.getText();
scrn.setText(temp+num);
}//GEN-LAST:event_jButton7ActionPerformed

private void jButton_dotActionPerformed
(java.awt.event.ActionEvent evt)
\{
//GEN-FIRST:event_jButton_dotActionPerformed
String temp=scrn.getText();
if (!temp.contains(".")){
scrn.setText(temp+".");
}
}
}//GEN-LAST:event_jButton_dotActionPerformed

private void jButton_signActionPerformed
(java.awt.event.ActionEvent evt)
{
//GEN-FIRST:event_jButton_signActionPerformed
String temp=scrn.getText();
if(temp.startsWith("-")){
scrn.setText(temp.split("-")[1]);
}
else{
scrn.setText("-"+temp);
}
}

}//GEN-LAST:event_jButton_signActionPerformed

private void jButton_equalsActionPerformed
```



```
(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButton1ActionPerformed
    FileOutputStream fos=null;
    try {
        // TODO add your handling code here:
        secondNumber=Double.parseDouble(scrn.getText());
        fos = new FileOutputStream("log.txt",true);

        if(operation!=' '){
            oper.setText(firstNumber+" "+operation+" "+secondNumber);}

        switch(operation){

            case '+':Double ans=firstNumber+secondNumber;
            fos.write((firstNumber.toString()+" "+secondNumber
                .toString()+"="+ans.toString()+"\n").getBytes());
            scrn.setText(ans.toString());
            break;

            case '-':ans=firstNumber-secondNumber;
            fos.write((firstNumber.toString()+" "+secondNumber
                .toString()+"="+ans.toString()+"\n").getBytes());
            scrn.setText(ans.toString());
            break;

            case '*':ans=firstNumber*secondNumber;
            fos.write((firstNumber.toString()+" "+secondNumber
                .toString()+"="+ans.toString()+"\n").getBytes());
            scrn.setText(ans.toString());
            break;
            case '/':ans=firstNumber/secondNumber;
            fos.write((firstNumber.toString()+" "+secondNumber
                .toString()+"="+ans.toString()+"\n").getBytes());
            scrn.setText(ans.toString());
            break;
            default:operation=' ';
        }
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Calc_Ui.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

```
}//GEN-LAST:event_jButton_equalsActionPerformed
```

SCOE 26 DEPT(COMP ENGG) 2015-2016

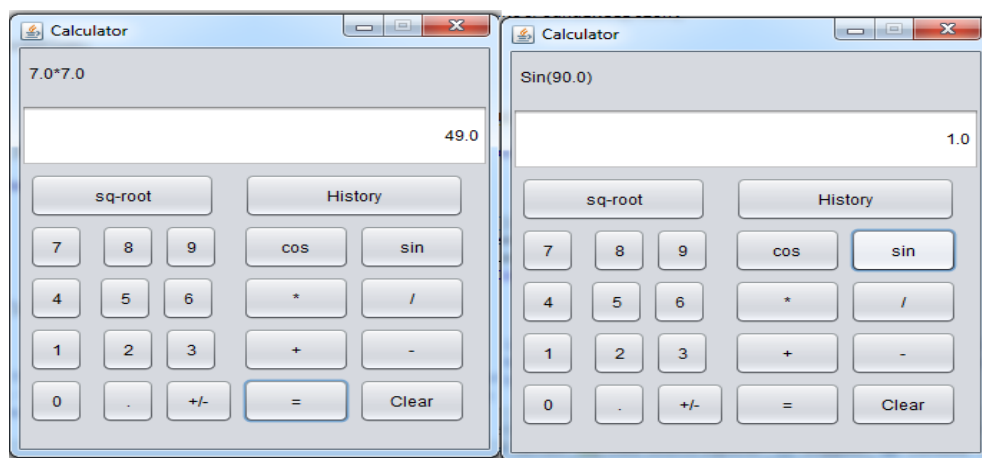
```
java.util.logging.Logger.getLogger(Calc_Ui.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Calc_Ui.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>
```

```
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new Calc_Ui().setVisible(true);
}
});
}
```

```
// Variables declaration – do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton0;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JButton jButton9;
private javax.swing.JButton jButton_add;
private javax.swing.JButton jButton_clear;
private javax.swing.JButton jButton_cos;
private javax.swing.JButton jButton_div;
private javax.swing.JButton jButton_dot;
private javax.swing.JButton jButton_equals;
private javax.swing.JButton jButton_mul;
private javax.swing.JButton jButton_sign;
private javax.swing.JButton jButton_sin;
private javax.swing.JButton jButton_sqrt;
private javax.swing.JButton jButton_sqrt1;
private javax.swing.JButton jButton_sub;
```

```
private javax.swing.JLabel oper;  
private javax.swing.JTextField scrn;  
// End of variables declaration//GEN-END:variables  
}
```

## 7 Output-



## 8 Conclusion:

Thus we studied and implemented a mobile app for calculator having trigonometry functionality is to be designed and tested.

Signature of Subject Teacher:-\_\_\_\_\_