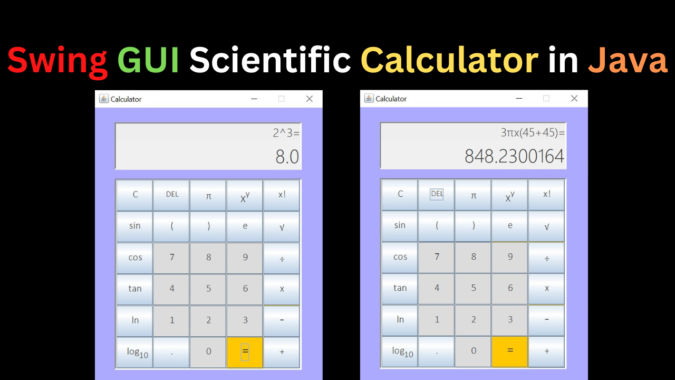
**[Calculator in Java](https://copyassignment.com/scientific-calculator-in-java/)**

THAMMISHETTI NIKHITHA  FEBRUARY 29 2024

[](https://copyassignment.com/scientific-calculator-in-java/)

This article will help you create your own Swing GUI Scientific Calculator in Java. I will give you the source code and explain the major functions of the program. You can use this as the starting point for your own programs or just use it as is. It has all the basic functionalities of a scientific calculator. You can also evaluate expressions and view the result of an expression with parentheses().

**Project Overview: Swing GUI Scientific Calculator in Java**

|  |  |
| --- | --- |
| **Project Name:** | **Swing GUI Scientific Calculator in Java** |
| **Abstract**: | It’s a GUI-based project used with the Swing module to organize all the elements that work under the Scientific Calculator in Java. |
| **Language/s Used:** | **Java** |
| **IDE**: | **IntelliJ Idea Professional(Recommended)** |
| **Java version (Recommended):** | **Java SE 18.0.** **2.1** |
| **Database**: | No need |
| **Type:** | Desktop Application |
| **Recommended for:** | Intermediate to Advance in Java |
| **Time needed for project**: | 2 – 2.5 hours |

**What will you learn?**

* [Math class](https://www.javatpoint.com/java-math#:~:text=Java%20Math%20class%20provides%20several,%2Dfor%2Dbit%20same%20results.) in Java
* Handling Classes and Objects creations
* Functions, Loops, Conditionals, and variables
* Java Swing and Java AWT for creating a user-friendly GUI.

**Features:**

* Addition, Subtraction, Multiplication, and Division
* Finding Sin, Cos, Tan, Log, Factorial, Pi, Square, and Square root of a number

Now, we will look at the code for Scientific Calculator in Java. Comments are provided for better understanding.

**Complete Code for Scientific Calculator in java:-**

import javax.swing.\*;

import javax.swing.border.BevelBorder;

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

class Calculator {

JFrame frmCalculator;

String result="",expression="";

ArrayList<String> token=new ArrayList<String>();

boolean num=false;

boolean dot=false;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Calculator window = new Calculator();

window.frmCalculator.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

Calculator() {

initialize();

}

int precedence(String x)

{

int p=10;

switch(x) {

case "+":

p=1;

break;

case "-":

p=2;

break;

case "x":

p=3;

break;

case "/":

p=4;

break;

case "^":

p=6;

break;

case "!":

p=7;

break;

}

return p;

}

//operator checking

private boolean isoperator(String x)

{

if(x.equals("+") || x.equals("-") || x.equals("x") || x.equals("/") || x.equals("sqrt") || x.equals("^") || x.equals("!") || x.equals("sin") || x.equals("cos") || x.equals("tan") || x.equals("ln") || x.equals("log"))

return true;

else

return false;

}

private String infixTopostfix()

{

Stack<String> s=new Stack<String>();

String y;

int flag;

String p="";

token.add(")");

s.push("(");

for(String i: token) {

if(i.equals("(")){

s.push(i);

}else if(i.equals(")")){

y=s.pop();

while(!y.equals("("))

{

p=p+y+",";

y=s.pop();

}

}else if(isoperator(i)){

y=s.pop();

flag=0;

if(isoperator(y) && precedence(y)>precedence(i)){

p=p+y+",";

flag=1;

}

if(flag==0)

s.push(y);

s.push(i);

}else{

p=p+i+",";

}

}

while(!s.empty()) {

y=s.pop();

if(!y.equals("(") && !y.equals(")")) {

p+=y+",";

}

}

return p;

}

//factorial method

private double factorial(double y) {

double fact=1;

if(y==0 || y==1) {

fact=1;

}else {

for(int i=2; i<=y; i++) {

fact\*=i;

}

}

return fact;

}

//for actual calculation with binary operators

private double calculate(double x,double y,String c)

{

double res=0;

switch(c)

{

case "-":

res= x-y;

break;

case "+":

res= x+y;

break;

case "x":

res= x\*y;

break;

case "/":

res= x/y;

break;

case "^":

res= Math.pow(x,y);

break;

default :

res= 0;

}

return res;

}

//calculation with unary operators

private double calculate(double y,String c) {

double res=0;

switch(c) {

case "log":

res = Math.log10(y);

break;

case "sin":

res= Math.sin(y);

break;

case "cos":

res = Math.cos(y);

break;

case "tan":

res =Math.tan(y);

break;

case "ln":

res= Math.log(y);

break;

case "sqrt":

res= Math.sqrt(y);

break;

case "!":

res=factorial(y);

break;

}

return res;

}

private double Eval(String p)

{

String tokens[] = p.split(",");

ArrayList<String> token2=new ArrayList<String>();

for(int i=0; i<tokens.length; i++) {

if(! tokens[i].equals("") && ! tokens[i].equals(" ") && ! tokens[i].equals("\n") && ! tokens[i].equals(" ")) {

token2.add(tokens[i]); // tokens from post fix form p actual tokens for calculation

}

}

Stack<Double> s=new Stack<Double>();

double x,y;

for(String i:token2) {

if(isoperator(i)){

//if it is unary operator or function

if(i.equals("sin") ||i.equals("cos") ||i.equals("tan") ||i.equals("log") || i.equals("ln") || i.equals("sqrt") || i.equals("!")) {

y=s.pop();

s.push(calculate(y,i));

}else {

//for binary operators

y=s.pop();

x=s.pop();

s.push(calculate(x,y,i));

}

}else{

if(i.equals("pi"))

s.push(Math.PI);

else if(i.equals("e"))

s.push(Math.E);

else

s.push(Double.valueOf(i));

}

}

double res=1;

while(!s.empty()) {

res\*=s.pop();

}

return res; //final result

}

//actual combined method for calculation

private void calculateMain() {

String tokens[]=expression.split(",");

for(int i=0; i<tokens.length; i++) {

if(! tokens[i].equals("") && ! tokens[i].equals(" ") && ! tokens[i].equals("\n") && ! tokens[i].equals(" ")) {

token.add(tokens[i]); //adding token to token array list from expression

}

}

try {

double res = Eval(infixTopostfix());

result= Double.toString(res);

}catch(Exception e) {}

}

//design of the frame with their action listner

private void initialize() {

frmCalculator = new JFrame();

frmCalculator.setResizable(false);

frmCalculator.setTitle("Calculator");

frmCalculator.getContentPane().setBackground(new Color(172, 170, 255));

frmCalculator.getContentPane().setFont(new Font("Calibri", Font.PLAIN, 15));

frmCalculator.getContentPane().setForeground(SystemColor.windowBorder);

frmCalculator.getContentPane().setLayout(null);

JPanel textPanel = new JPanel();

textPanel.setBorder(new BevelBorder(BevelBorder.LOWERED, null, null, null, null));

textPanel.setBounds(34, 25, 316, 80);

frmCalculator.getContentPane().add(textPanel);

textPanel.setLayout(null);

JLabel exprlabel = new JLabel("");

exprlabel.setBackground(SystemColor.control);

exprlabel.setFont(new Font("Yu Gothic UI Light", Font.PLAIN, 20));

exprlabel.setHorizontalAlignment(SwingConstants.RIGHT);

exprlabel.setForeground(UIManager.getColor("Button.disabledForeground"));

exprlabel.setBounds(2, 2, 312, 27);

textPanel.add(exprlabel);

JTextField textField = new JTextField();

exprlabel.setLabelFor(textField);

textField.setHorizontalAlignment(SwingConstants.RIGHT);

textField.setBackground(SystemColor.control);

textField.setEditable(false);

textField.setText("0");

textField.setBorder(null);

textField.setFont(new Font("Yu Gothic UI Light", textField.getFont().getStyle(), 32));

textField.setBounds(2, 30, 312, 49);

textPanel.add(textField);

textField.setColumns(10);

JPanel butttonPanel = new JPanel();

butttonPanel.setBorder(new BevelBorder(BevelBorder.LOWERED, null, null, null, null));

butttonPanel.setBackground(SystemColor.inactiveCaptionBorder);

butttonPanel.setBounds(34, 120, 316, 322);

frmCalculator.getContentPane().add(butttonPanel);

butttonPanel.setLayout(new GridLayout(0, 5, 0, 0));

//clear button

JButton button1 = new JButton("C");

button1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

textField.setText("0");

exprlabel.setText("");

expression ="";

token.clear();

result="";

num=false;

dot=false;

}

});

button1.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button1);

//delete button

JButton button2 = new JButton("DEL");

button2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String s=textField.getText();

if(s != "0" && s.length() > 1) {

String newString = s.substring(0,s.length()-1);

textField.setText(newString);

if(expression.charAt(expression.length()-1)=='.') {

dot=false;

}

if(expression.charAt(expression.length()-1) == ',') {

expression = expression.substring(0,expression.length()-2);

}else {

expression = expression.substring(0,expression.length()-1);

}

}else {

textField.setText("0");

expression="";

}

}

});

button2.setFont(new Font("Calibri Light", Font.PLAIN, 14));

butttonPanel.add(button2);

//button for constant pi

JButton button3 = new JButton("<html><body><span>π</span></body></html>");

button3.setFont(new Font("Calibri Light", Font.PLAIN, 17));

button3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+Character.toString((char)960));

}else {

textField.setText(Character.toString((char)960));

}

expression += ",pi";

num=false;

dot=false;

}

});

butttonPanel.add(button3);

//button for x^Y

JButton button4 = new JButton("<html><body><span>X<sup>y</sup></span></body></html>");

button4.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"^");

expression+=",^";

}else {

textField.setText("0^");

expression += ",0,^";

}

num=false;

dot=false;

}

});

button4.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button4);

//factorial button

JButton buttton5 = new JButton("x!");

buttton5.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"!");

expression+=",!";

}else {

textField.setText("0!");

expression+=",0,!";

}

num=false;

dot=false;

}

});

buttton5.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(buttton5);

//button for sin

JButton button6 = new JButton("sin");

button6.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"sin(");

}else {

textField.setText("sin(");

}

expression+=",sin,(";

num=false;

dot=false;

}

});

button6.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button6);

JButton button7 = new JButton("(");

button7.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"(");

}else {

textField.setText("(");

}

expression+=",(";

num=false;

dot=false;

}

});

button7.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button7);

JButton button8 = new JButton(")");

button8.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+")");

}else {

textField.setText(")");

}

expression+=",)";

num=false;

dot=false;

}

});

button8.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button8);

JButton button9 = new JButton("e");

button9.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"e");

}else {

textField.setText("e");

}

expression+=",e";

num=false;

dot=false;

}

});

button9.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button9);

//button for squre root

JButton button10 = new JButton("<html><body><span>√</span></body></html>");

button10.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+Character.toString((char)8730));

}else {

textField.setText(Character.toString((char)8730));

}

expression+=",sqrt";

num=false;

dot=false;

}

});

button10.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button10);

JButton button11 = new JButton("cos");

button11.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"cos(");

}else {

textField.setText("cos(");

}

expression+=",cos,(";

num=false;

dot=false;

}

});

button11.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button11);

JButton button12 = new JButton("7");

button12.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"7");

}else {

textField.setText("7");

}

if(num) {

expression+="7";

}else {

expression+=",7";

}

num=true;

}

});

button12.setBackground(new Color(220, 220, 220));

button12.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button12);

JButton button13 = new JButton("8");

button13.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"8");

}else {

textField.setText("8");

}

if(num) {

expression+="8";

}else {

expression+=",8";

}

num=true;

}

});

button13.setBackground(new Color(220, 220, 220));

button13.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button13);

JButton button14 = new JButton("9");

button14.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"9");

}else {

textField.setText("9");

}

if(num) {

expression+="9";

}else {

expression+=",9";

}

num=true;

}

});

button14.setBackground(new Color(220, 220, 220));

button14.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button14);

//button for division operator

JButton button15 = new JButton("<html><body><span>÷</span></body></html>");

button15.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String s=textField.getText();

if(s.equals("0")) {

expression+="0";

}

if(s.charAt(s.length()-1)== '-' || s.charAt(s.length()-1)== 'x' || s.charAt(s.length()-1) == '+') {

String newString = s.substring(0,s.length()-1);

textField.setText(newString+Character.toString((char)247));

expression = expression.substring(0,expression.length()-1);

expression += "/";

}else if(s.charAt(s.length()-1)!= (char)247) {

textField.setText(s+Character.toString((char)247));

expression+=",/";

}else {

textField.setText(s);

}

num=false;

dot=false;

}

});

button15.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button15);

JButton button16 = new JButton("tan");

button16.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"tan(");

}else {

textField.setText("tan(");

}

expression+=",tan,(";

num=false;

dot=false;

}

});

button16.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button16);

JButton button17 = new JButton("4");

button17.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"4");

}else {

textField.setText("4");

}

if(num) {

expression+="4";

}else {

expression+=",4";

}

num=true;

}

});

button17.setBackground(new Color(220, 220, 220));

button17.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button17);

JButton button18 = new JButton("5");

button18.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"5");

}else {

textField.setText("5");

}

if(num) {

expression+="5";

}else {

expression+=",5";

}

num=true;

}

});

button18.setBackground(new Color(220, 220, 220));

button18.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button18);

JButton button19 = new JButton("6");

button19.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"6");

}else {

textField.setText("6");

}

if(num) {

expression+="6";

}else {

expression+=",6";

}

num=true;

}

});

button19.setBackground(new Color(220, 220, 220));

button19.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button19);

JButton button20 = new JButton("x");

button20.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String s=textField.getText();

if(s.equals("0")) {

expression+="0";

}

if(s.charAt(s.length()-1)== '-' || s.charAt(s.length()-1)== '+' || s.charAt(s.length()-1) == (char)(247)) {

String newString = s.substring(0,s.length()-1);

newString += "x";

textField.setText(newString);

expression = expression.substring(0,expression.length()-1);

expression += "x";

}else if(s.charAt(s.length()-1)!= 'x') {

s += "x";

textField.setText(s);

expression+=",x";

}else {

textField.setText(s);

}

num=false;

dot=false;

}

});

button20.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button20);

JButton button21 = new JButton("ln");

button21.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"ln(");

}else {

textField.setText("ln(");

}

expression+=",ln,(";

num=false;

dot=false;

}

});

button21.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button21);

JButton button22 = new JButton("1");

button22.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"1");

}else {

textField.setText("1");

}

if(num) {

expression+="1";

}else {

expression+=",1";

}

num=true;

}

});

button22.setBackground(new Color(220, 220, 220));

button22.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button22);

JButton button23 = new JButton("2");

button23.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"2");

}else {

textField.setText("2");

}

if(num) {

expression+="2";

}else {

expression+=",2";

}

num=true;

}

});

button23.setBackground(new Color(220, 220, 220));

button23.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button23);

JButton button24 = new JButton("3");

button24.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"3");

}else {

textField.setText("3");

}

if(num) {

expression+="3";

}else {

expression+=",3";

}

num=true;

}

});

button24.setBackground(new Color(220, 220, 220));

button24.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button24);

JButton button25 = new JButton("-");

button25.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String s=textField.getText();

if(s.equals("0")) {

expression+="0";

}

if(s.charAt(s.length()-1)== '+') {

String newString = s.substring(0,s.length()-1);

newString += "-";

expression = expression.substring(0,expression.length()-1);

expression += "-";

textField.setText(newString);

}else if(s.charAt(s.length()-1)!= '-') {

s += "-";

textField.setText(s);

expression += ",-";

}else {

textField.setText(s);

}

num=false;

dot=false;

}

});

button25.setFont(new Font("Calibri Light", Font.BOLD, 23));

butttonPanel.add(button25);

JButton button26 = new JButton("<html><body><span>log<sub>10</sub></span></body></html>");

button26.setFont(new Font("Calibri Light", Font.PLAIN, 17));

button26.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(! "0".equals(textField.getText())) {

textField.setText(textField.getText()+"log(");

}else {

textField.setText("log(");

}

expression+=",log,(";

num=false;

dot=false;

}

});

butttonPanel.add(button26);

JButton button27 = new JButton(".");

button27.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String s=textField.getText();

if(s.charAt(s.length()-1)!= '.') {

if(num && dot==false) {

expression+=".";

s += ".";

}else if(num==false && dot ==false){

expression+=",.";

s += ".";

}

}

num=true;

dot=true;

textField.setText(s);

}

});

button27.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button27);

JButton button28 = new JButton("0");

button28.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if("0".equals(textField.getText())) {

textField.setText("0");

}else {

textField.setText(textField.getText()+"0");

if(num) {

expression+="0";

}

else {

expression+=",0";

}

}

num=true;

}

});

button28.setBackground(new Color(220, 220, 220));

button28.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button28);

//actual functioning on clicking = button

JButton button29 = new JButton("=");

button29.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

calculateMain();

String s="";

token.remove(token.size()-1);

for(String i: token) {

if(i.equals("/")) {

s+=Character.toString((char)247);

}else if(i.equals("sqrt")) {

s+=Character.toString((char)8730);

}else if(i.equals("pi")) {

s+=Character.toString((char)960);

}else {

s+=i;

}

}

exprlabel.setText(s+"=");

textField.setText(result);

expression = result;

dot=true;

num=true;

token.clear();

}

});

button29.setBackground(Color.ORANGE);

button29.setFont(new Font("Calibri Light", Font.PLAIN, 22));

butttonPanel.add(button29);

JButton button30 = new JButton("+");

button30.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String s=textField.getText();

if(s.equals("0")) {

expression+="0";

}

if(s.charAt(s.length()-1)== '-' || s.charAt(s.length()-1)== 'x' || s.charAt(s.length()-1) == (char)(247)) {

String newString = s.substring(0,s.length()-1);

newString += "+";

textField.setText(newString);

expression = expression.substring(0,expression.length()-1);

expression += "+";

}else if(s.charAt(s.length()-1)!= '+') {

s += "+";

textField.setText(s);

expression+=",+";

}else {

textField.setText(s);

}

num=false;

dot=false;

}

});

button30.setFont(new Font("Calibri Light", Font.PLAIN, 17));

butttonPanel.add(button30);

frmCalculator.setBounds(200, 100, 400, 500);

frmCalculator.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

**Output:**



**Conclusion:**

This article has shown you how you can create a simple GUI Scientific Calculator in Java. There are many other things that can be done with this calculator. You can add the ability to calculate trigonometric equations, change the keys, or use your own images. The sky is the limit !!