

## **Scientific Calculator**

**(PCCCS – 503)**

**Samya Bose; C5-187**

### **Problem Statement:**

Design a scientific calculator using the Java Swing framework. Your design should contain following operations,

1. Basic operators: +, -, \*, /, ^ (for exponential), % (mod)
2. Basic math function: sin, cos, tan, log, sin-1, cos-1, tan-1 (inverse)
3. Basic errors: parenthesis checking, operand checking (e.g. 1..2, 1.2.3, 12A etc. are invalid operands),
4. Basic conversions from one unit to another unit.
5. Basic mathematical conversion like decimal to binary and vice versa.

Input pattern:

- I. Your input expression may contain operands with multiple digits. (For example,  $23+8*9$  is valid).
- II. It can evaluate any combination like  $2+\cos(x)+\log 20$
- III. It can also work perfectly for those operators' associativity working from right to left.

Basic Design characteristics:

You can add multiple panels in a single layout. Each frame consists of multiple buttons of similar type of operations. For example all the buttons of mathematical functions like sin, cos etc must be added into a single panel.

You have to design in a concise manner. For example, you can add multiple layouts to cover up different functionalities. Suppose in a single layout you are covering only the point 4.

## Implementation:

<https://drive.google.com/drive/folders/1JvfA5etfS5q4HtT34ELkqe8-my-lTAuJ?usp=sharing>

## Code:

### ScientificCalculator.java

```
package scientific.calculator.system;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Stack;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Line2D;

public class ScientificCalculator extends JFrame implements ActionListener{

    public static final String[] TEXT = {"C", "Del", "(", ")", "e", "Deg", "sin-1", "sin", "x^y", "\u221A", "!",
        "\u00F7", "cos-1", "cos", "7", "8", "9", "x", "tan-1", "tan", "4", "5", "6", "+", "log", "ln", "1", "2", "3", "-", "%",
        "\u03C0", ".", "0", "00", "=", "Calculate"};

    public JTextField t1, t2, t3;
    public JLabel l1;
    public JComboBox choice;
    public JPanel contentPane;
    public String result="";
    public String expression="";
    public String error="";
    public ArrayList<String> token=new ArrayList<String>();
    public boolean num=false;
    public boolean dot=false;

    String reverse(String input)
```

```

{
    char[] temparray = input.toCharArray();
    int left, right = 0;
    right = temparray.length - 1;

    for (left = 0; left < right; left++, right--)
    {
        char temp = temparray[left];
        temparray[left] = temparray[right];
        temparray[right] = temp;
    }
    return String.valueOf(temparray);
}

```

```

double hexToDecimal(String str) {

    int flag = 0;
    for(int i=0; i<str.length(); i++) {
        if(str.charAt(i)=='.')
            flag = 1;
    }
    if(flag==0)
        str+="0";

    String temp1 = str.split("\\.")[0];
    String temp2 = str.split("\\.")[1];

    double sum = 0.0;
    int k=-1;
    temp1 = reverse(temp1);
    for(int i=0; i<temp1.length();i++) {
        String x = String.valueOf(temp1.charAt(i));
        if(x.equals("A"))
            x="10";
        else if(x.equals("B"))
            x="11";
        else if(x.equals("C"))

```

```

        x="12";
    else if(x.equals("D"))
        x="13";
    else if(x.equals("E"))
        x="14";
    else if(x.equals("F"))
        x="15";

    sum+=Double.parseDouble(x)*Math.pow(16,i);
}

for(int j=0; j<temp2.length(); j++) {
    String x = String.valueOf(temp2.charAt(j));

    if(x.equals("A"))
        x="10";
    else if(x.equals("B"))
        x="11";
    else if(x.equals("C"))
        x="12";
    else if(x.equals("D"))
        x="13";
    else if(x.equals("E"))
        x="14";
    else if(x.equals("F"))
        x="15";

    sum+=Double.parseDouble(x)*Math.pow(16,k);

    k--;
}

return sum;
}

```

```
String toHexFraction(double x, int digits) {
```

```
    // Get fractional part.
```

```
    if (x < 0.0)
```

```
        x = 0.0 - x;
```

```
    x = x % 1.0;
```

```

// Shift left by n digits
long multiplier = (1L << (digits * 4));
long fraction = (long)(x * multiplier);

// Convert integer to hex string.
// String should have at least n digits; prefix with zeros if not.
String hex = Long.toHexString(fraction);
String padding = "0000000000000000";
hex = padding.substring(0, digits - hex.length()) + hex;

return hex;
}

String toHexInteger(double x) {
    long whole = (long) x;
    String prefix;
    if (whole < 0) {
        // Long.toHexString treats the number as an unsigned integer.
        whole = 0 - whole;
        prefix = "-";
    } else {
        prefix = "";
    }
    return Long.toHexString(whole);
}

String toHex (double x, int digits) {
    return toHexInteger(x) + "." + toHexFraction(x, digits);
}

String decimalToBinary(double num, int k_prec)
{
    String binary = "";
    int Integral = (int) num;
    double fractional = num - Integral;

```

```

while (Integral > 0)
{
    int rem = Integral % 2;
    binary += ((char)(rem + '0'));
    Integral /= 2;
}
binary = reverse(binary);
binary += ('.');
while (k_prec-- > 0)
{
    fractional *= 2;
    int fract_bit = (int) fractional;
    if (fract_bit == 1)
    {
        fractional -= fract_bit;
        binary += (char)(1 + '0');
    }
    else
    {
        binary += (char)(0 + '0');
    }
}
return binary;
}

```

```

double binaryToDecimal(String binary,int len) {

```

```

    boolean flag = false;
    for(int i=0; i<binary.length(); i++) {
        if(binary.charAt(i)=='1' || binary.charAt(i)=='0' || binary.charAt(i)=='.')
            flag = true;
        else
            flag = false;
    }
    if(flag){

```

```

int point = binary.indexOf('.');
if (point == -1)
    point = len;

double intDecimal = 0, fracDecimal = 0, twos = 1;
for(int i = point - 1; i >= 0; i--)
{
    intDecimal += (binary.charAt(i) - '0') * twos;
    twos *= 2;
}
twos = 2;
for(int i = point + 1; i < len; i++)
{
    fracDecimal += (binary.charAt(i) - '0') / twos;
    twos *= 2.0;
}
return intDecimal + fracDecimal;
}
else
    return -100.100;
}

```

```

public void paint(Graphics g) {
    super.paint(g);
    Graphics2D graphics = (Graphics2D) g;
    graphics.setColor(new Color(52, 54, 56));
    Line2D line = new Line2D.Float(0,500,420,500);
    graphics.draw(line);
}

```

```

public static void main(String[] args) {
    new ScientificCalculator().setVisible(true);
}

```

```

public ScientificCalculator() {

```

```
setBounds(400, 150, 420, 800);

contentPane = new JPanel();
setContentPane(contentPane);
contentPane.setLayout(null);
contentPane.setBackground(new Color(27, 28, 29));
setLayout(null);


t1 = new JTextField("", 4);
t1.setBounds(50, 30, 300, 60);
t1.setBackground(new Color(52, 54, 56));
t1.setForeground(new Color(255, 255, 255));
Font font = t1.getFont();
float size = font.getSize() + 2.0f;
t1.setFont(font.deriveFont(size));
t1.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
add(t1);


l1 = new JLabel("Unit Converter");
l1.setBounds(160, 480, 300, 30);
l1.setBackground(new Color(52, 54, 56));
l1.setForeground(new Color(255, 255, 255));
add(l1);


t2 = new JTextField("", 4);
t2.setBounds(50, 520, 300, 30);
t2.setBackground(new Color(52, 54, 56));
t2.setForeground(new Color(255, 255, 255));
t2.setFont(font.deriveFont(size));
t2.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
add(t2);


t3 = new JTextField("", 4);
t3.setBounds(50, 570, 300, 30);
t3.setBackground(new Color(52, 54, 56));
```



```
t3.setForeground(new Color(255, 255, 255));

t3.setFont(font.deriveFont(size));

t3.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));

add(t3);
```

```
String[] choices = {"decimal to binary", "binary to decimal", "decimal to hexadecimal", "hexadecimal
to decimal", "kg to lbs", "lbs to kg", "km to miles", "miles to km", "celsius to fahrenheit", "fahrenheit to
celsius", "cm to inch", "inch to cm"};
```

```
choice = new JComboBox<String>(choices);

choice.setBounds(50, 620, 300, 30);

choice.setForeground(new Color(255, 255, 255));

choice.setBackground(new Color(52, 54, 56));

add(choice);
```

```
//      JButton b1 = new JButton(""+ "Calculate");
//      b1.setBounds(150,670,100,50);
//      add(b1);
```

```
int x = 50;

int y = 115;

int k = 1;

JButton b[] = new JButton[37];

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

    b[i] = new JButton(""+ TEXT[i]);

    b[i].setBounds(x * k, y, 50, 50);

    if(TEXT[i]=="=")

        b[i].setBackground(new Color(0,128,128));

    if(TEXT[i]=="Calculate") {

        b[i].setBounds(150,670,100,50);

        b[i].setBackground(new Color(39, 41, 43));

        b[i].setForeground(new Color(255, 255, 255));

        b[i].setBorder(BorderFactory.createLineBorder(new Color(27, 28, 29)));

    }

    else

        b[i].setBackground(new Color(39, 41, 43));
```

```

        b[i].setForeground(new Color(255, 255, 255));

        b[i].setBorder(BorderFactory.createLineBorder(new Color(27, 28, 29)));

        if (k % 6 == 0) {

            x = 50;

            y += 50;

            k = 0;

        }

        k++;

        add(b[i]);

        b[i].addActionListener(this);

    }

}

```

```

public 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;
    }
}

```

```

        return p;
    }

    public boolean isoperator(String x)
    {
        if(x.equals("+") || x.equals("-") || x.equals("x") || x.equals("/") || x.equals("sqrt") ||
x.equals("%") || x.equals("x^y") || x.equals("!") ||

        x.equals("sin") || x.equals("cos") || x.equals("tan") ||x.equals("sin-1") || x.equals("cos-1") ||
x.equals("tan-1") || 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;
        case "%":
            res= x%y;
        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;

    case "sin-1":
        res= Math.asin(y);
    case "cos-1":
        res= Math.acos(y);
    case "tan-1":
        res= Math.atan(y);
    }
    return res;
}

```

//to evaluate postfix expressions using stack

```

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("sin-1")
||i.equals("cos-1") ||i.equals("tan-1") ||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) {
        error = "ERROR";
    }
}

public void actionPerformed(ActionEvent e) {
    String s = e.getActionCommand();
    if (s.equalsIgnoreCase("=")) {
        calculateMain();
        if(!error.equals("")) {
            t1.setText(error);
            expression="";
            dot=false;
            num=false;
        }else {
            String str="";
            token.remove(token.size()-1);
            for(String i: token) {
                if(i.equals("/")) {
                    str+=Character.toString((char)247);
                }else if(i.equals("sqrt")) {
                    str+=Character.toString((char)8730);
                }
            }
        }
    }
}

```



```

        }else if(i.equals("pi")) {
            str+=Character.toString((char)960);
        }else {
            str+=i;
        }
    }
//    exprlabel.setText(s+"=");
    t1.setText(result);

    expression = result;
    dot=true;
    num=true;
}
error="";
token.clear();
} else if (s.equalsIgnoreCase("C")) {
    t1.setText("0");
//    exprlabel.setText("");
    expression = "";
    token.clear();
    result="";
    num=false;
    dot=false;
    error="";
} else if (s.equalsIgnoreCase("Del")) {
    String str=t1.getText();
    if(s != "0" && s.length() > 1) {
        String newString = str.substring(0,str.length()-1);
        t1.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 {
    t1.setText("0");
    expression="";
}
}
else if (s.equalsIgnoreCase("\u03C0")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+Character.toString((char)960));
    }
    else {
        t1.setText(Character.toString((char)960));
    }
    expression += ",pi";
    num=false;
    dot=false;
}
else if (s.equalsIgnoreCase("x^y")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"^");
        expression+=",";
    }
    else {
        t1.setText("0^");
        expression += ",0,";
    }
    num=false;
    dot=false;
}
else if (s.equalsIgnoreCase("!")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"!");
        expression+=",";
    }
    else {
        t1.setText("0!");
        expression+=",";
    }
    num=false;
    dot=false;
}

```

```

}else if (s.equalsIgnoreCase("sin")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"sin(");
    }else {
        t1.setText("sin(");
    }
    expression+="sin,(";
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase("sin-1")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"sin-1(");
    }else {
        t1.setText("sin-1(");
    }
    expression+="sin,(";
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase("(")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"(");
    }else {
        t1.setText("(");
    }
    expression+="(";
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase(")")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+")");
    }else {
        t1.setText(")");
    }
    expression+=",)";
    num=false;

```

```

        dot=false;
    }else if (s.equalsIgnoreCase("e")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"e");
        }else {
            t1.setText("e");
        }
        expression+="e";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("\u221A")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+Character.toString((char)8730));
        }else {
            t1.setText(Character.toString((char)8730));
        }
        expression+="sqrt";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("cos")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"cos(");
        }else {
            t1.setText("cos(");
        }
        expression+="cos,(";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("cos-1")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"cos-1(");
        }else {
            t1.setText("cos-1(");
        }
        expression+="cos-1,(";
    }

```

```

num=false;

dot=false;

}else if (s.equalsIgnoreCase("tan")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"tan(");

    }else {

        t1.setText("tan(");

    }

    expression+="tan,(";

    num=false;

    dot=false;

}else if (s.equalsIgnoreCase("tan-1")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"tan-1(");

    }else {

        t1.setText("tan-1(");

    }

    expression+="tan-1,(";

    num=false;

    dot=false;

}else if (s.equalsIgnoreCase("7")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"7");

    }else {

        t1.setText("7");

    }

    if(num) {

        expression+="7";

    }else {

        expression+="7";

    }

    num=true;

}else if (s.equalsIgnoreCase("8")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"8");

```

```

    }else {
        t1.setText("8");
    }
    if(num) {
        expression+="8";
    }else {
        expression+=",8";
    }
    num=true;
}
else if (s.equalsIgnoreCase("9")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"9");
    }else {
        t1.setText("9");
    }
    if(num) {
        expression+="9";
    }else {
        expression+=",9";
    }
    num=true;
}
else if (s.equalsIgnoreCase("/")) {
    String str=t1.getText();
    if(str.equals("0")) {
        expression+="0";
    }
    if(str.charAt(str.length()-1)=='-' || str.charAt(str.length()-1)=='x' || str.charAt(str.length()-1)=='+')
{
        String newString = str.substring(0,str.length()-1);
        t1.setText(newString+Character.toString((char)247));
        expression = expression.substring(0,expression.length()-1);
        expression += "/";
    }
    else if(str.charAt(str.length()-1)!= (char)247) {
        t1.setText(str+Character.toString((char)247));
        expression+=",/";
    }
}

```

```
}else {  
    t1.setText(str);  
}  
num=false;  
dot=false;  
}else if (s.equalsIgnoreCase("4")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"4");  
    }else {  
        t1.setText("4");  
    }  
    if(num) {  
        expression+="4";  
    }else {  
        expression+=",4";  
    }  
    num=true;  
}else if (s.equalsIgnoreCase("5")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"5");  
    }else {  
        t1.setText("5");  
    }  
    if(num) {  
        expression+="5";  
    }else {  
        expression+=",5";  
    }  
    num=true;  
}else if (s.equalsIgnoreCase("6")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"6");  
    }else {  
        t1.setText("6");  
    }  
}
```

```

        if(num) {
            expression+="6";
        }else {
            expression+=",6";
        }
        num=true;
    }else if (s.equalsIgnoreCase("x")) {
        String str=t1.getText();
        if(str.equals("0")) {
            expression+="0";
        }
        if(str.charAt(str.length()-1)== '-' || str.charAt(str.length()-1)== '+' || str.charAt(str.length()-1) ==
(char)(247)) {
            String newString = str.substring(0,str.length()-1);
            t1.setText(newString+Character.toString('x'));
            expression = expression.substring(0,expression.length()-1);
            expression += "x";
        }else if(str.charAt(str.length()-1)!= (char)247) {
            t1.setText(str+Character.toString('x'));
            expression+=",x";
        }else {
            t1.setText(str);
        }
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("ln")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"ln(");
        }else {
            t1.setText("ln(");
        }
        expression+="ln,(";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("1")) {

```



```
if(! "0".equals(t1.getText())) {  
    t1.setText(t1.getText()+"1");  
}else {  
    t1.setText("1");  
}  
if(num) {  
    expression+="1";  
}else {  
    expression+=",1";  
}  
num=true;  
}else if (s.equalsIgnoreCase("2")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"2");  
    }else {  
        t1.setText("2");  
    }  
    if(num) {  
        expression+="2";  
    }else {  
        expression+=",2";  
    }  
    num=true;  
}else if (s.equalsIgnoreCase("3")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"3");  
    }else {  
        t1.setText("3");  
    }  
    if(num) {  
        expression+="3";  
    }else {  
        expression+=",3";  
    }  
    num=true;
```

```

}else if (s.equalsIgnoreCase("Deg")) {

    new ScientificCalculator1().setVisible(true);

    this.setVisible(false);
}else if (s.equalsIgnoreCase("-")) {

    String str=t1.getText();

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

        expression+="0";

    }

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

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

        newString += "-";

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

        expression += "-";

        t1.setText(newString);
    }else if(str.charAt(str.length()-1)!='-') {

        str += "-";

        t1.setText(str);

        expression += "-";
    }else {

        t1.setText(str);

    }

    num=false;

    dot=false;
}else if (s.equalsIgnoreCase("log")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"log(");

    }else {

        t1.setText("log(");

    }

    expression+="log(";

    num=false;

    dot=false;
}else if (s.equalsIgnoreCase(".")) {

    String str=t1.getText();

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

```

```

        if(num && dot==false) {
            expression+=".";
            str += ".";
        }else if(num==false && dot ==false){
            expression+=",";
            str += ",";
        }
    }
}

num=true;
dot=true;
t1.setText(str);
}else if (s.equalsIgnoreCase("0")) {
    if("0".equals(t1.getText())) {
        t1.setText("0");
    }else {
        t1.setText(t1.getText()+"0");
        if(num) {
            expression+="0";
        }
        else {
            expression+=",0";
        }
    }
}

num=true;
}else if (s.equalsIgnoreCase("00")) {
    if("0".equals(t1.getText())) {
        t1.setText("00");
    }else {
        t1.setText(t1.getText()+"00");
        if(num) {
            expression+="00";
        }
        else {
            expression+=",00";
        }
    }
}

```

```

    }

    num=true;
}
else if (s.equalsIgnoreCase("+")) {

    String str=t1.getText();

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

        expression+="0";

    }

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

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

        newString += "+";

        t1.setText(newString);

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

        expression += "+";

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

        str += "+";

        t1.setText(str);

        expression+=",";

    }
    else {

        t1.setText(str);

    }

    num=false;

    dot=false;

}

else if (s.equalsIgnoreCase("Calculate")) {

    String str = t2.getText();

    String temp = String.valueOf(choice.getSelectedItem());

    try{

        if(temp=="kg to lbs")

            str = String.valueOf(Double.parseDouble(str)*2.205);

        else if(temp=="lbs to kg")

            str = String.valueOf(Double.parseDouble(str)*0.454);

        else if(temp=="km to miles")

            str = String.valueOf(Double.parseDouble(str)*0.621);

        else if(temp=="miles to km")

```

```

        str = String.valueOf(Double.parseDouble(str)*1.609);
    else if(temp=="celsius to fahrenheit")
        str = String.valueOf(Double.parseDouble(str)*9/5+32);
    else if(temp=="fahrenheit to celsius")
        str = String.valueOf((Double.parseDouble(str)-32)*5/9);
    else if(temp=="cm to inch")
        str = String.valueOf(Double.parseDouble(str)*0.394);
    else if(temp=="inch to cm")
        str = String.valueOf(Double.parseDouble(str)*2.54);
    else if(temp=="decimal to binary") {
        Double res = Double.parseDouble(str);
        str = decimalToBinary(res, 10);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    else if(temp=="binary to decimal") {
        Double res = binaryToDecimal(str, str.length());
        if(res== -100.100)
            str="ERROR";
        else
            str = String.valueOf(res);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    else if(temp=="decimal to hexadecimal") {
        str = toHex(Double.parseDouble(str), 10);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    else if(temp=="hexadecimal to decimal") {
        Double result = hexToDecimal(str);
        str = String.valueOf(result);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    t3.setText(str);
} catch (Exception err) {
    t3.setText("ERROR");
}

```

```

    }
}
}

```

ScientificCalculator1.java

```

package scientific.calculator.system;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Stack;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Line2D;

public class ScientificCalculator1 extends JFrame implements ActionListener{

    public static final String[] TEXT = {"C", "Del", "(", ")", "e", "Rad", "sin-1", "sin", "x^y", "\u221A", "!",
        "\u00F7", "cos-1", "cos", "7", "8", "9", "x", "tan-1", "tan", "4", "5", "6", "+", "log", "ln", "1", "2", "3", "-", "%",
        "\u03C0", ".", "0", "00", "=", "Calculate"};

    public JTextField t1, t2, t3;
    public JLabel l1;
    public JComboBox choice;
    public JPanel contentPane;
    public String result="";
    public String expression="";
    public String error="";
    public ArrayList<String> token=new ArrayList<String>();
    public boolean num=false;
    public boolean dot=false;

    String reverse(String input)
    {

```

```

        char[] temparray = input.toCharArray();

        int left, right = 0;

        right = temparray.length - 1;

        for (left = 0; left < right; left++, right--)
        {
            char temp = temparray[left];
            temparray[left] = temparray[right];
            temparray[right] = temp;
        }

        return String.valueOf(temparray);
    }

```

```

double hexToDecimal(String str) {

    int flag = 0;

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

        if(str.charAt(i)=='.')

            flag = 1;

    }

    if(flag==0)

        str+="0";

    String temp1 = str.split("\\.")[0];
    String temp2 = str.split("\\.")[1];

    double sum = 0.0;

    int k=-1;

    temp1 = reverse(temp1);

    for(int i=0; i<temp1.length();i++) {

        String x = String.valueOf(temp1.charAt(i));

        if(x.equals("A"))

            x="10";

        else if(x.equals("B"))

            x="11";

        else if(x.equals("C"))

            x="12";

```

```

else if(x.equals("D"))
    x="13";
else if(x.equals("E"))
    x="14";
else if(x.equals("F"))
    x="15";
    sum+=Double.parseDouble(x)*Math.pow(16,i);
}
for(int j=0; j<temp2.length(); j++) {
    String x = String.valueOf(temp2.charAt(j));
    if(x.equals("A"))
        x="10";
    else if(x.equals("B"))
        x="11";
    else if(x.equals("C"))
        x="12";
    else if(x.equals("D"))
        x="13";
    else if(x.equals("E"))
        x="14";
    else if(x.equals("F"))
        x="15";
    sum+=Double.parseDouble(x)*Math.pow(16,k);
    k--;
}
return sum;
}

```

```

String toHexFraction(double x, int digits) {

```

```

    // Get fractional part.

```

```

    if (x < 0.0)

```

```

        x = 0.0 - x;

```

```

    x = x % 1.0;

```

```

    // Shift left by n digits

```



```

    long multiplier = (1L << (digits * 4));
    long fraction = (long)(x * multiplier);

    // Convert integer to hex string.
    // String should have at least n digits; prefix with zeros if not.
    String hex = Long.toHexString(fraction);
    String padding = "0000000000000000";
    hex = padding.substring(0, digits - hex.length()) + hex;

    return hex;
}

String toHexInteger(double x) {
    long whole = (long) x;
    String prefix;
    if (whole < 0) {
        // Long.toHexString treats the number as an unsigned integer.
        whole = 0 - whole;
        prefix = "-";
    } else {
        prefix = "";
    }
    return Long.toHexString(whole);
}

String toHex (double x, int digits) {
    return toHexInteger(x) + "." + toHexFraction(x, digits);
}

String decimalToBinary(double num, int k_prec)
{
    String binary = "";
    int Integral = (int) num;
    double fractional = num - Integral;
    while (Integral > 0)

```

```

    {
        int rem = Integral % 2;
        binary += ((char)(rem + '0'));
        Integral /= 2;
    }
    binary = reverse(binary);
    binary += ('.');
    while (k_prec-- > 0)
    {
        fractional *= 2;
        int fract_bit = (int) fractional;
        if (fract_bit == 1)
        {
            fractional -= fract_bit;
            binary += (char)(1 + '0');
        }
        else
        {
            binary += (char)(0 + '0');
        }
    }
    return binary;
}

```

```
double binaryToDecimal(String binary,int len) {
```

```

    boolean flag = false;
    for(int i=0; i<binary.length(); i++) {
        if(binary.charAt(i)=='1' || binary.charAt(i)=='0' || binary.charAt(i)=='.')
            flag = true;
        else
            flag = false;
    }
    if(flag){
        int point = binary.indexOf('.');

```

```

        if (point == -1)
            point = len;

        double intDecimal = 0, fracDecimal = 0, twos = 1;
        for(int i = point - 1; i >= 0; i--)
        {
            intDecimal += (binary.charAt(i) - '0') * twos;
            twos *= 2;
        }
        twos = 2;
        for(int i = point + 1; i < len; i++)
        {
            fracDecimal += (binary.charAt(i) - '0') / twos;
            twos *= 2.0;
        }
        return intDecimal + fracDecimal;
    }
    else
        return -100.100;
}

```

```

public void paint(Graphics g) {
    super.paint(g);
    Graphics2D graphics = (Graphics2D) g;
    graphics.setColor(new Color(52, 54, 56));
    Line2D line = new Line2D.Float(0,500,420,500);
    graphics.draw(line);
}

```

```

    public static void main(String[] args) {
        new ScientificCalculator1().setVisible(true);
    }

```

```

public ScientificCalculator1() {

```

```
setBounds(400, 150, 420, 800);

contentPane = new JPanel();
setContentPane(contentPane);
contentPane.setLayout(null);

contentPane.setBackground(new Color(27, 28, 29));
setLayout(null);


t1 = new JTextField("", 4);
t1.setBounds(50, 30, 300, 60);
t1.setBackground(new Color(52, 54, 56));
t1.setForeground(new Color(255, 255, 255));

Font font = t1.getFont();
float size = font.getSize() + 2.0f;
t1.setFont(font.deriveFont(size));
t1.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
add(t1);


l1 = new JLabel("Unit Converter");
l1.setBounds(160, 480, 300, 30);
l1.setBackground(new Color(52, 54, 56));
l1.setForeground(new Color(255, 255, 255));
add(l1);


t2 = new JTextField("", 4);
t2.setBounds(50, 520, 300, 30);
t2.setBackground(new Color(52, 54, 56));
t2.setForeground(new Color(255, 255, 255));
t2.setFont(font.deriveFont(size));
t2.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
add(t2);


t3 = new JTextField("", 4);
t3.setBounds(50, 570, 300, 30);
t3.setBackground(new Color(52, 54, 56));
t3.setForeground(new Color(255, 255, 255));
```

```

t3.setFont(font.deriveFont(size));

t3.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));

add(t3);

String[] choices = {"decimal to binary", "binary to decimal", "decimal to hexadecimal", "hexadecimal
to decimal", "kg to lbs", "lbs to kg", "km to miles", "miles to km", "celsius to fahrenheit", "fahrenheit to
celsius", "cm to inch", "inch to cm"};

choice = new JComboBox<String>(choices);

choice.setBounds(50, 620, 300, 30);

choice.setForeground(new Color(255, 255, 255));

choice.setBackground(new Color(52, 54, 56));

add(choice);

//      JButton b1 = new JButton(""+ "Calculate");
//      b1.setBounds(150,670,100,50);
//      add(b1);

int x = 50;

int y = 115;

int k = 1;

JButton b[] = new JButton[37];

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

    b[i] = new JButton(""+ TEXT[i]);

    b[i].setBounds(x * k, y, 50, 50);

    if(TEXT[i]=="=")

        b[i].setBackground(new Color(0,128,128));

    if(TEXT[i]=="Calculate") {

        b[i].setBounds(150,670,100,50);

        b[i].setBackground(new Color(39, 41, 43));

        b[i].setForeground(new Color(255, 255, 255));

        b[i].setBorder(BorderFactory.createLineBorder(new Color(27, 28, 29)));

    }

    else

        b[i].setBackground(new Color(39, 41, 43));

        b[i].setForeground(new Color(255, 255, 255));

```

```

b[i].setBorder(BorderFactory.createLineBorder(new Color(27, 28, 29)));
if (k % 6 == 0) {
    x = 50;
    y += 50;
    k = 0;
}
k++;
add(b[i]);
b[i].addActionListener(this);

}
}

```

```

public 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;
    }

```

```

    return p;

```

```

    }

    public boolean isoperator(String x)
    {
        if(x.equals("+") || x.equals("-") || x.equals("x") || x.equals("/") || x.equals("sqrt") ||
x.equals("%") || x.equals("x^y") || x.equals("!") ||

        x.equals("sin") || x.equals("cos") || x.equals("tan") ||x.equals("sin-1") || x.equals("cos-1") ||
x.equals("tan-1") || 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;  
        case "%":  
            res= x%y;  
        default :  
            res= 0;  
    }  
    return res;  
}
```

```
//calculation with unary operators
```

```
private double calculate(double y,String c) {
```

```
    double res=0;  
    y = y/57.2958;  
    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;

    case "sin-1":
        res= Math.asin(y);
    case "cos-1":
        res= Math.acos(y);
    case "tan-1":
        res= Math.atan(y);
    }
    return res;
}

```

//to evaluate postfix expressions using stack

```

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("sin-1")
||i.equals("cos-1") ||i.equals("tan-1") ||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) {
        error = "ERROR";
    }
}

public void actionPerformed(ActionEvent e) {
    String s = e.getActionCommand();
    if (s.equalsIgnoreCase("=")) {
        calculateMain();
        if(!error.equals("")) {
            t1.setText(error);
            expression="";
            dot=false;
            num=false;
        }else {
            String str="";
            token.remove(token.size()-1);
            for(String i: token) {
                if(i.equals("/")) {
                    str+=Character.toString((char)247);
                }else if(i.equals("sqrt")) {
                    str+=Character.toString((char)8730);
                }
            }
        }
    }
}

```

```

        }else if(i.equals("pi")) {
            str+=Character.toString((char)960);
        }else {
            str+=i;
        }
    }
//    exprlabel.setText(s+"=");
    t1.setText(result);

    expression = result;
    dot=true;
    num=true;
}
error="";
token.clear();
} else if (s.equalsIgnoreCase("C")) {
    t1.setText("0");
//    exprlabel.setText("");
    expression = "";
    token.clear();
    result="";
    num=false;
    dot=false;
    error="";
} else if (s.equalsIgnoreCase("Del")) {
    String str=t1.getText();
    if(s != "0" && s.length() > 1) {
        String newString = str.substring(0,str.length()-1);
        t1.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 {
    t1.setText("0");
    expression="";
}
}
else if (s.equalsIgnoreCase("\u03C0")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+Character.toString((char)960));
    }
    else {
        t1.setText(Character.toString((char)960));
    }
    expression += ",pi";
    num=false;
    dot=false;
}
else if (s.equalsIgnoreCase("x^y")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"^");
        expression+=",";
    }
    else {
        t1.setText("0^");
        expression += ",0,";
    }
    num=false;
    dot=false;
}
else if (s.equalsIgnoreCase("!")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"!");
        expression+=",";
    }
    else {
        t1.setText("0!");
        expression+=",";
    }
    num=false;
    dot=false;
}

```

```

}else if (s.equalsIgnoreCase("sin")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"sin(");
    }else {
        t1.setText("sin(");
    }
    expression+="sin,(";
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase("sin-1")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"sin-1(");
    }else {
        t1.setText("sin-1(");
    }
    expression+="sin,(";
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase("(")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"(");
    }else {
        t1.setText("(");
    }
    expression+="(";
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase(")")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+")");
    }else {
        t1.setText(")");
    }
    expression+=")";
    num=false;

```

```

        dot=false;
    }else if (s.equalsIgnoreCase("e")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"e");
        }else {
            t1.setText("e");
        }
        expression+="e";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("\u221A")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+Character.toString((char)8730));
        }else {
            t1.setText(Character.toString((char)8730));
        }
        expression+="sqrt";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("cos")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"cos(");
        }else {
            t1.setText("cos(");
        }
        expression+="cos,(";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("cos-1")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"cos-1(");
        }else {
            t1.setText("cos-1(");
        }
        expression+="cos-1,(";
    }

```



```

num=false;

dot=false;

}else if (s.equalsIgnoreCase("tan")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"tan(");

    }else {

        t1.setText("tan(");

    }

    expression+="tan,(";

    num=false;

    dot=false;

}else if (s.equalsIgnoreCase("tan-1")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"tan-1(");

    }else {

        t1.setText("tan-1(");

    }

    expression+="tan-1,(";

    num=false;

    dot=false;

}else if (s.equalsIgnoreCase("7")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"7");

    }else {

        t1.setText("7");

    }

    if(num) {

        expression+="7";

    }else {

        expression+="7";

    }

    num=true;

}else if (s.equalsIgnoreCase("8")) {

    if(! "0".equals(t1.getText())) {

        t1.setText(t1.getText()+"8");

```

```

    }else {
        t1.setText("8");
    }
    if(num) {
        expression+="8";
    }else {
        expression+=",8";
    }
    num=true;
}
else if (s.equalsIgnoreCase("9")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"9");
    }else {
        t1.setText("9");
    }
    if(num) {
        expression+="9";
    }else {
        expression+=",9";
    }
    num=true;
}
else if (s.equalsIgnoreCase("/")) {
    String str=t1.getText();
    if(str.equals("0")) {
        expression+="0";
    }
    if(str.charAt(str.length()-1)=='-' || str.charAt(str.length()-1)=='x' || str.charAt(str.length()-1)=='+')
{
        String newString = str.substring(0,str.length()-1);
        t1.setText(newString+Character.toString((char)247));
        expression = expression.substring(0,expression.length()-1);
        expression += "/";
    }
    else if(str.charAt(str.length()-1)!= (char)247) {
        t1.setText(str+Character.toString((char)247));
        expression+=",/";
    }
}

```

```
}else {  
    t1.setText(str);  
}  
num=false;  
dot=false;  
}else if (s.equalsIgnoreCase("4")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"4");  
    }else {  
        t1.setText("4");  
    }  
    if(num) {  
        expression+="4";  
    }else {  
        expression+=",4";  
    }  
    num=true;  
}else if (s.equalsIgnoreCase("5")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"5");  
    }else {  
        t1.setText("5");  
    }  
    if(num) {  
        expression+="5";  
    }else {  
        expression+=",5";  
    }  
    num=true;  
}else if (s.equalsIgnoreCase("6")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"6");  
    }else {  
        t1.setText("6");  
    }  
}
```

```

        if(num) {
            expression+="6";
        }else {
            expression+=",6";
        }
        num=true;
    }else if (s.equalsIgnoreCase("x")) {
        String str=t1.getText();
        if(str.equals("0")) {
            expression+="0";
        }
        if(str.charAt(str.length()-1)== '-' || str.charAt(str.length()-1)== '+' || str.charAt(str.length()-1) ==
(char)(247)) {
            String newString = str.substring(0,str.length()-1);
            t1.setText(newString+Character.toString('x'));
            expression = expression.substring(0,expression.length()-1);
            expression += "x";
        }else if(str.charAt(str.length()-1)!= (char)247) {
            t1.setText(str+Character.toString('x'));
            expression+=",x";
        }else {
            t1.setText(str);
        }
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("ln")) {
        if(! "0".equals(t1.getText())) {
            t1.setText(t1.getText()+"ln(");
        }else {
            t1.setText("ln(");
        }
        expression+="ln,(";
        num=false;
        dot=false;
    }else if (s.equalsIgnoreCase("1")) {

```

```
if(! "0".equals(t1.getText())) {  
    t1.setText(t1.getText()+"1");  
}else {  
    t1.setText("1");  
}  
  
if(num) {  
    expression+="1";  
}else {  
    expression+=",1";  
}  
  
num=true;  
}else if (s.equalsIgnoreCase("2")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"2");  
    }else {  
        t1.setText("2");  
    }  
  
    if(num) {  
        expression+="2";  
    }else {  
        expression+=",2";  
    }  
  
    num=true;  
}else if (s.equalsIgnoreCase("3")) {  
    if(! "0".equals(t1.getText())) {  
        t1.setText(t1.getText()+"3");  
    }else {  
        t1.setText("3");  
    }  
  
    if(num) {  
        expression+="3";  
    }else {  
        expression+=",3";  
    }  
  
    num=true;
```

```

}else if (s.equalsIgnoreCase("Rad")) {
    this.setVisible(false);
    new ScientificCalculator().setVisible(true);
}else if (s.equalsIgnoreCase("-")) {
    String str=t1.getText();
    if(str.equals("0")) {
        expression+="0";
    }
    if(str.charAt(str.length()-1)=='+') {
        String newString = str.substring(0,str.length()-1);
        newString += "-";
        expression = expression.substring(0,expression.length()-1);
        expression += "-";
        t1.setText(newString);
    }else if(str.charAt(str.length()-1)!='-') {
        str += "-";
        t1.setText(str);
        expression += "-";
    }else {
        t1.setText(str);
    }
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase("log")) {
    if(! "0".equals(t1.getText())) {
        t1.setText(t1.getText()+"log(");
    }else {
        t1.setText("log(");
    }
    expression+="log(";
    num=false;
    dot=false;
}else if (s.equalsIgnoreCase(".")) {
    String str=t1.getText();
    if(str.charAt(str.length()-1)!='.') {

```

```

        if(num && dot==false) {
            expression+=".";
            str += ".";
        }else if(num==false && dot ==false){
            expression+=",";
            str += ",";
        }
    }
}

num=true;
dot=true;
t1.setText(str);
}else if (s.equalsIgnoreCase("0")) {
    if("0".equals(t1.getText())) {
        t1.setText("0");
    }else {
        t1.setText(t1.getText()+"0");
        if(num) {
            expression+="0";
        }
        else {
            expression+=",0";
        }
    }
}

num=true;
}else if (s.equalsIgnoreCase("00")) {
    if("0".equals(t1.getText())) {
        t1.setText("00");
    }else {
        t1.setText(t1.getText()+"00");
        if(num) {
            expression+="00";
        }
        else {
            expression+=",00";
        }
    }
}

```

```

    }

    num=true;
} else if (s.equalsIgnoreCase("+")) {

    String str=t1.getText();

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

        expression+="0";

    }

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

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

        newString += "+";

        t1.setText(newString);

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

        expression += "+";

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

        str += "+";

        t1.setText(str);

        expression+=",";

    } else {

        t1.setText(str);

    }

    num=false;

    dot=false;

}

else if (s.equalsIgnoreCase("Calculate")) {

    String str = t2.getText();

    String temp = String.valueOf(choice.getSelectedItem());

    try{

        if(temp=="kg to lbs")

            str = String.valueOf(Double.parseDouble(str)*2.205);

        else if(temp=="lbs to kg")

            str = String.valueOf(Double.parseDouble(str)*0.454);

        else if(temp=="km to miles")

            str = String.valueOf(Double.parseDouble(str)*0.621);

        else if(temp=="miles to km")

```



```

        str = String.valueOf(Double.parseDouble(str)*1.609);
    else if(temp=="celsius to fahrenheit")
        str = String.valueOf(Double.parseDouble(str)*9/5+32);
    else if(temp=="fahrenheit to celsius")
        str = String.valueOf((Double.parseDouble(str)-32)*5/9);
    else if(temp=="cm to inch")
        str = String.valueOf(Double.parseDouble(str)*0.394);
    else if(temp=="inch to cm")
        str = String.valueOf(Double.parseDouble(str)*2.54);
    else if(temp=="decimal to binary") {
        Double res = Double.parseDouble(str);
        str = decimalToBinary(res, 10);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    else if(temp=="binary to decimal") {
        Double res = binaryToDecimal(str, str.length());
        if(res== -100.100)
            str="ERROR";
        else
            str = String.valueOf(res);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    else if(temp=="decimal to hexadecimal") {
        str = toHex(Double.parseDouble(str), 10);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    else if(temp=="hexadecimal to decimal") {
        Double result = hexToDecimal(str);
        str = String.valueOf(result);
        str = str.indexOf(".") < 0 ? str : str.replaceAll("0*$", "").replaceAll("\\.$", "");
    }
    t3.setText(str);
} catch (Exception err) {
    t3.setText("ERROR");
}

```

```

    }
}
}

```

Output:

