Ex.No: 11	**Calculator Application using java AWT packages**
Date:	

Aim:

To create a Java GUI application that mimics the basic and advanced functionalities of a scientific calculator.

Algorithm:

Step 1 Start the Process

Step 2 Display Text View and Number Pad and Option Pads

Step 3 If user presses any number get the existing numbers in Text View add them up and display

Step 4 If user press Operators

- Step 4.1 Get the Text View content as Operant 1 and Set the display to null.
- Step 4.2 If user pressed “+” button set operator as plus
- Step 4.3 If user pressed “-” button set operator as minus
- Step 4.4 If user pressed “x” button set operator as multiply
- Step 4.5 If user pressed “/” button set operator as divide
- Step 4.6 Goto step 3

Step 5 If user pressed “=” button then proceed following steps.

- Step 5.1 Get the Text View content as Operant 2 and Set the display to null.
- Step 5.2 If operator is “plus” then display Text View as Operant 1 + Operant 2
- Step 5.3 If operator is “minus” then display Text View as Operant 1 - Operant 2
- Step 5.4 If operator is “multiply” then display Text View as Operant 1 * Operant 2
- Step 5.5 If operator is “divide” then display Text View as Operant 1 / Operant 2
- Step 5.6 Goto step 4

Step 6 If advanced button pressed

- Step 6.1 Change Operant Types [+,-,x,/ into sin, cos, tan, log] and goto step 2

Step 7 If user pressed any of the operator

- Step 7.1 Get the Text View content as Operant 1 and Set the display to null.
- Step 7.2 If user pressed “sin” button set display the sin value of Operant 1
- Step 7.3 If user pressed “cos” button set display the cos value of Operant 1
- Step 7.4 If user pressed “tan” button set display the tan value of Operant 1
- Step 7.5 If user pressed “log” button set display the log value of Operant 1
- Step 7.6 Goto step 7

Step 8 If advanced pressed again then revert the button changes and return back to normal

Step 9 Repeat the process until user presses the close button then Stop the Process.

Coding:

```
import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Font;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.Panel;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

class Numpan extends Panel implements ActionListener{

	Button n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,point,equal;
	Button plus,minus,multiply, divide;
	Button m_plus,m_minus,clear,advanced;
	TextField display;
	String op1,op2,result;
	String op_flag;
	String data;
	double dop1,dop2,dresult;
	boolean flag_advanced=true;
	public Numpan(TextField display) {
		this.display = display;
		setLayout(new GridLayout(0,4));

		n0 = new Button("0");
		n0.setActionCommand("zero");
		n0.addActionListener(this);

		n1 = new Button("1");
		n1.setActionCommand("one");
		n1.addActionListener(this);

		n2 = new Button("2");
		n2.setActionCommand("two");
		n2.addActionListener(this);

		n3 = new Button("3");
		n3.setActionCommand("three");
		n3.addActionListener(this);

		n4 = new Button("4");
		n4.setActionCommand("four");
		n4.addActionListener(this);

		n5 = new Button("5");
		n5.setActionCommand("five");
```

```
n5.addActionListener(this);

n6 = new Button("6");
n6.setActionCommand("six");
n6.addActionListener(this);

n7 = new Button("7");
n7.setActionCommand("seven");
n7.addActionListener(this);

n8 = new Button("8");
n8.setActionCommand("eight");
n8.addActionListener(this);

n9 = new Button("9");
n9.setActionCommand("nine");
n9.addActionListener(this);

point = new Button(".");
point.setActionCommand("point");
point.addActionListener(this);

equal = new Button("=");
equal.setActionCommand("equal");
equal.addActionListener(this);

plus = new Button("+");
plus.setActionCommand("plus");
plus.addActionListener(this);

minus = new Button("-");
minus.setActionCommand("minus");
minus.addActionListener(this);

multiply = new Button("x");
multiply.setActionCommand("multiply");
multiply.addActionListener(this);

divide = new Button("/");
divide.setActionCommand("divide");
divide.addActionListener(this);

m_plus = new Button("M+");
m_plus.setActionCommand("m_plus");
m_plus.addActionListener(this);

m_minus = new Button("M-");
m_minus.setActionCommand("m_minus");
m_minus.addActionListener(this);

clear = new Button("C");
clear.setActionCommand("clear");
```

```
		clear.addActionListener(this);

		advanced = new Button("ADV");
		advanced.setActionCommand("advanced");
		advanced.addActionListener(this);

		add(m_plus);
		add(m_minus);
		add(clear);
		add(advanced);
		add(n1);
		add(n2);
		add(n3);
		add(plus);
		add(n4);
		add(n5);
		add(n6);
		add(minus);
		add(n7);
		add(n8);
		add(n9);
		add(multiply);
		add(point);
		add(n0);
		add(equal);
		add(divide);
}
public String getDisplayText(){
		return display.getText().toString();
}
public void setDisplay(String text){
		display.setText(text);
}
public void clearDisplay(){
		System.out.println("Clear Called");
		setDisplay("");
		data = "";
}

public void changeAdvanced(boolean toAdvanced){
		if(toAdvanced){
			plus.setLabel("sin");
			plus.setActionCommand("sin");
			//System.out.println("cos in");
			minus.setLabel("cos");
			minus.setActionCommand("cos");
			//System.out.println("cos out");
			multiply.setLabel("tan");
			multiply.setActionCommand("tan");
			divide.setLabel("log");
			divide.setActionCommand("log");
```

```
		}
		else{
			plus.setLabel("+");
			plus.setActionCommand("plus");
			minus.setLabel("-");
			minus.setActionCommand("minus");
			multiply.setLabel("x");
			multiply.setActionCommand("multiply");
			divide.setLabel("/");
			divide.setActionCommand("divide");
		}
}
@Override
public void actionPerformed(ActionEvent e) {

		data = getDisplayText();

		switch(e.getActionCommand()){
		case "zero":
			setDisplay(data+"0");
			break;
		case "one":
			setDisplay(data+"1");
			break;
		case "two":
			setDisplay(data+"2");
			break;
		case "three":
			setDisplay(data+"3");
			break;
		case "four":
			setDisplay(data+"4");
			break;
		case "five":
			setDisplay(data+"5");
			break;
		case "six":
			setDisplay(data+"6");
			break;
		case "seven":
			setDisplay(data+"7");
			break;
		case "eight":
			setDisplay(data+"8");
			break;
		case "nine":
			setDisplay(data+"9");
			break;

		case "plus":
			op1 = data;
			op_flag = "plus";
```

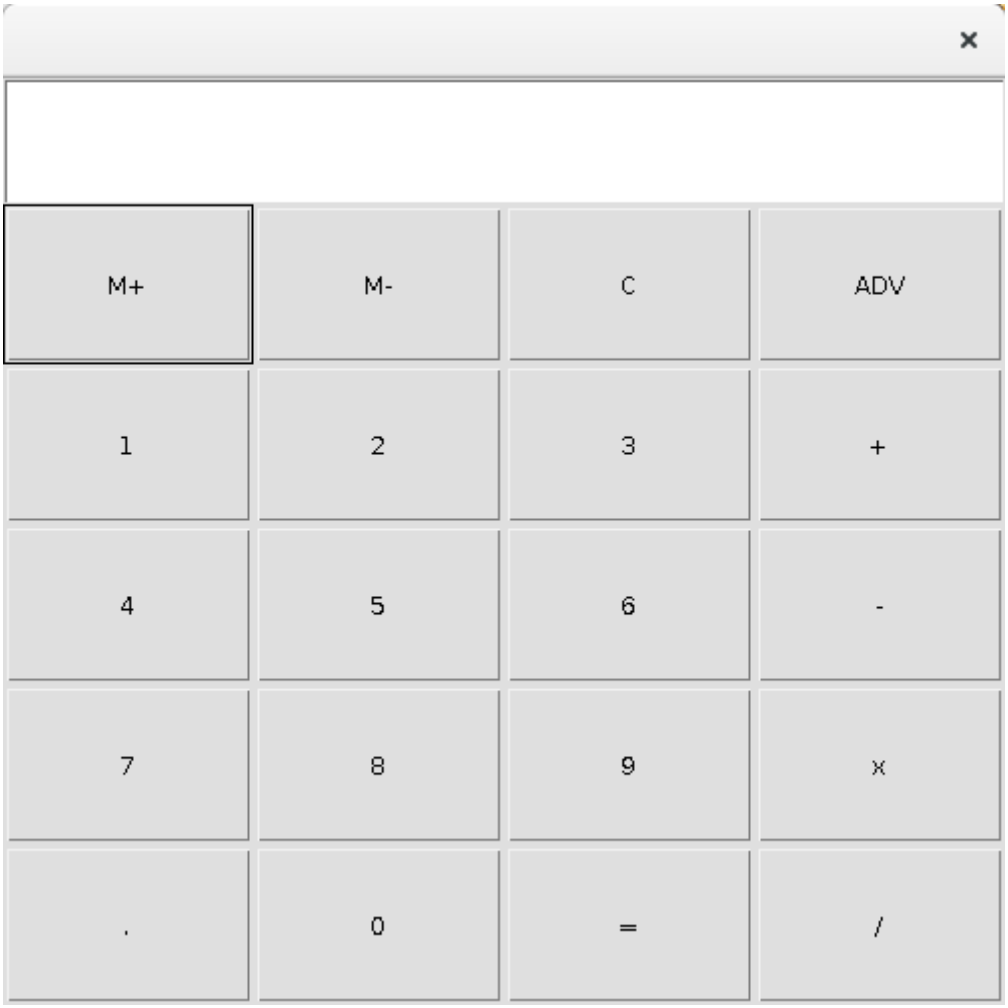
```
            clearDisplay();
            break;
    case "minus":
            op1 = data;
            op_flag = "minus";
            clearDisplay();
            break;
    case "multiply":
            op1 = data;
            op_flag = "multiply";
            clearDisplay();
            break;
    case "divide":
            op1 = data;
            op_flag = "divide";
            clearDisplay();
            break;
    case "clear":
            clearDisplay();
            break;
    case "advanced":
            if(flag_advanced){
                    changeAdvanced(true);
                    flag_advanced = false;
            }
            else{
                    changeAdvanced(false);
                    flag_advanced = true;
            }
            break;

    case "sin":
            op1 = data;
            setDisplay(String.valueOf(Math.sin(Double.valueOf(op1))));
            break;
    case "cos":
            op1 = data;
            setDisplay(String.valueOf(Math.cos(Double.valueOf(op1))));
            break;
    case "tan":
            op1 = data;
            setDisplay(String.valueOf(Math.tan(Double.valueOf(op1))));
            break;
    case "log":
            op1 = data;
            setDisplay(String.valueOf(Math.log(Double.valueOf(op1))));
            break;
    case "equal":
            switch(op_flag){
            case "plus":
                    op2 = data;
                    clearDisplay();
```

```
                    dop1 = Double.parseDouble(op1);
                    dop2 = Double.parseDouble(op2);
                    dresult = dop1 + dop2;
                    result = String.valueOf(dresult);
                    setDisplay(result);
                    op_flag = "";
                    break;
                case "minus":
                    op2 = data;
                    clearDisplay();
                    dop1 = Double.parseDouble(op1);
                    dop2 = Double.parseDouble(op2);
                    dresult = dop1 - dop2;
                    result = String.valueOf(dresult);
                    setDisplay(result);
                    op_flag = "";
                    break;

                case "multiply":
                    op2 = data;
                    clearDisplay();
                    dop1 = Double.parseDouble(op1);
                    dop2 = Double.parseDouble(op2);
                    dresult = dop1 * dop2;
                    result = String.valueOf(dresult);
                    setDisplay(result);
                    op_flag = "";
                    break;

                case "divide":
                    op2 = data;
                    clearDisplay();
                    dop1 = Double.parseDouble(op1);
                    dop2 = Double.parseDouble(op2);
                    dresult = dop1 / dop2;
                    result = String.valueOf(dresult);
                    setDisplay(result);
                    op_flag = "";
                    break;


                }
            }
        }
}

class Calculator extends Frame {
        TextField display;
        public Calculator() {
            display = new TextField();
            display.setFont(new Font("Times New Roman", Font.BOLD, 50));
            setLayout(new BorderLayout());
```

```
                add(new Numpan(display),BorderLayout.CENTER);
                add(display,BorderLayout.NORTH);
                setVisible(true);
                setSize(500,500);
                addWindowListener(new WindowAdapter() {
                        @Override
                        public void windowClosing(WindowEvent e) {
                                dispose();
                        }
                });
        }

}
public class Main {

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

}
```

Output:

Calculator:



Basic Operations

Addition [12 + 12]

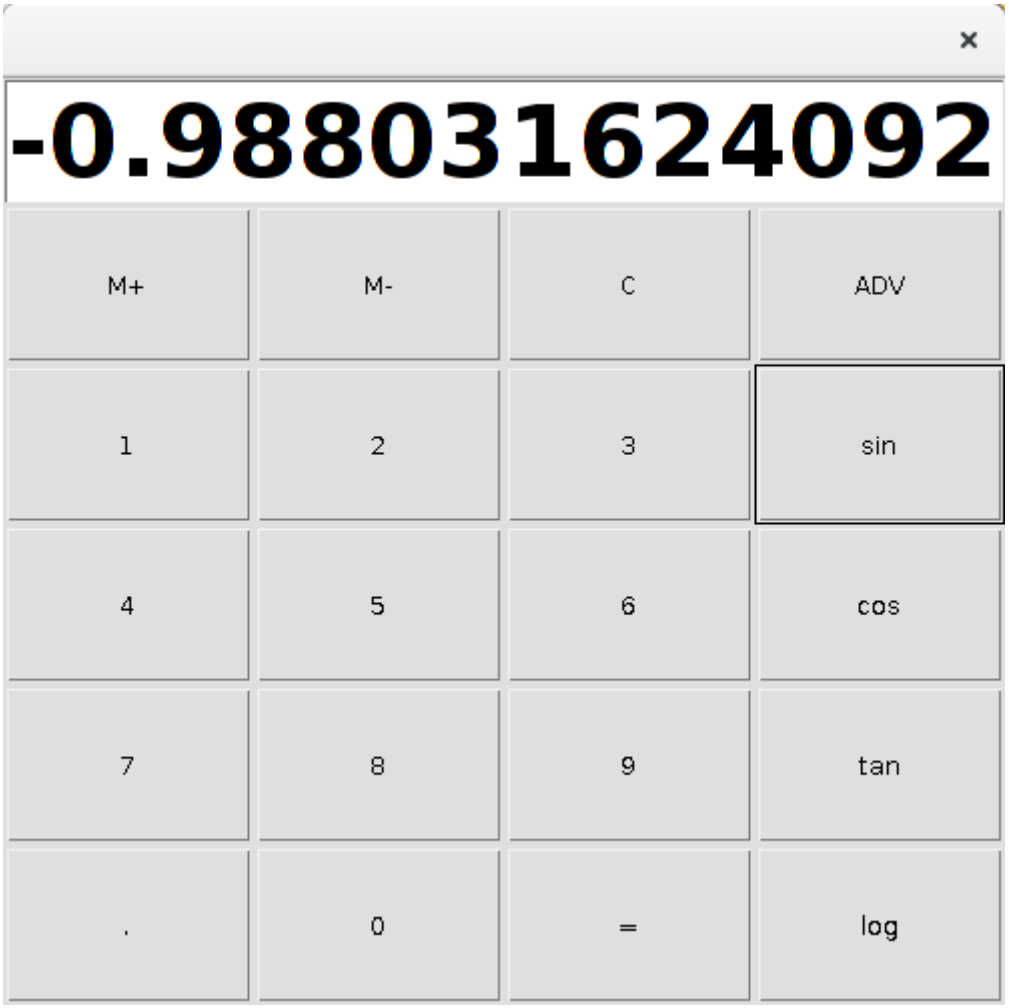
Subtraction [90 – 32]:

Multiplication [36 X 2]

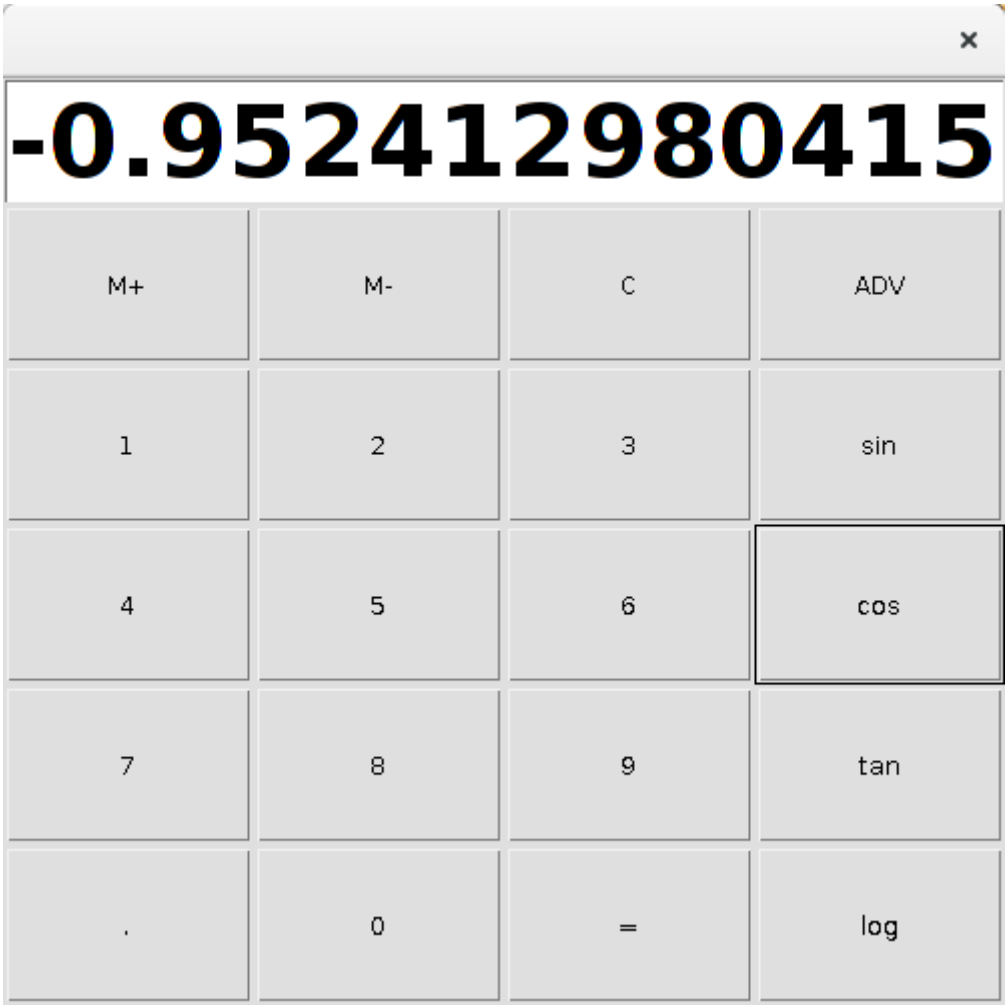
Division [45 / 2]

Scientific Operations:

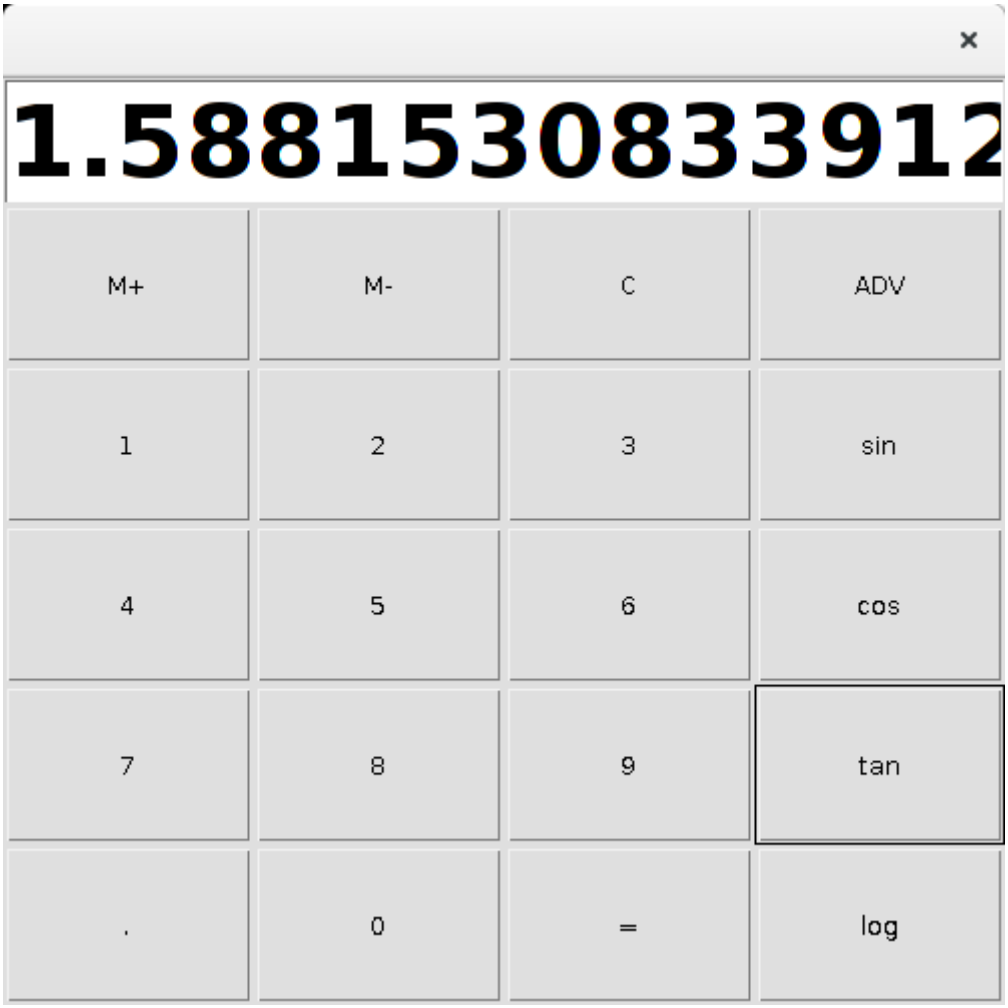
Sin 30



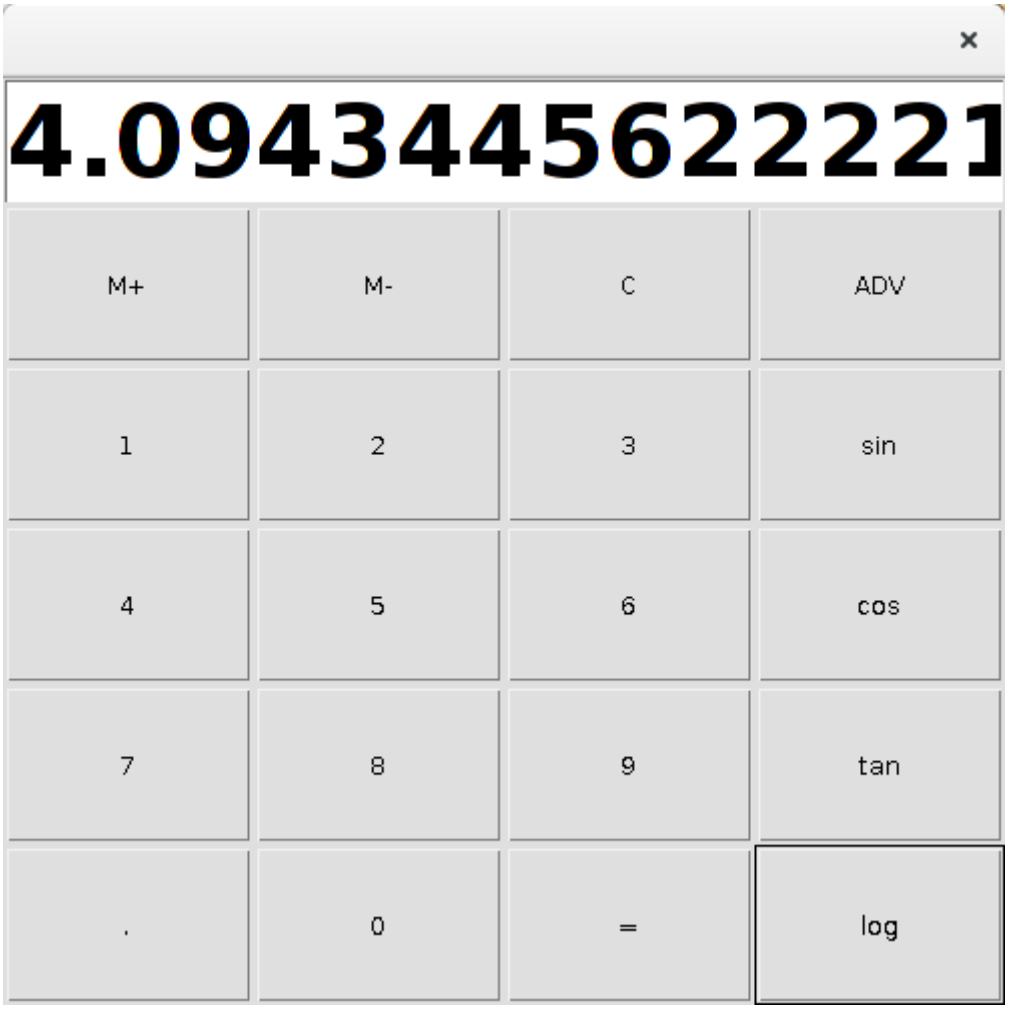
Cos 40:



Tan 23



Log 60



Result:

The java GUI application for calculator with basic and advanced functionalities was developed and tested successfully.