Exp.no:10	**Design of scientific calculator**
Date:12-09-19	

Aim: To design a calculator using event-driven programming paradigm of Java with Decimal and Scientific manipulations.

Algorithm:

Step 1: Declare a package calc.

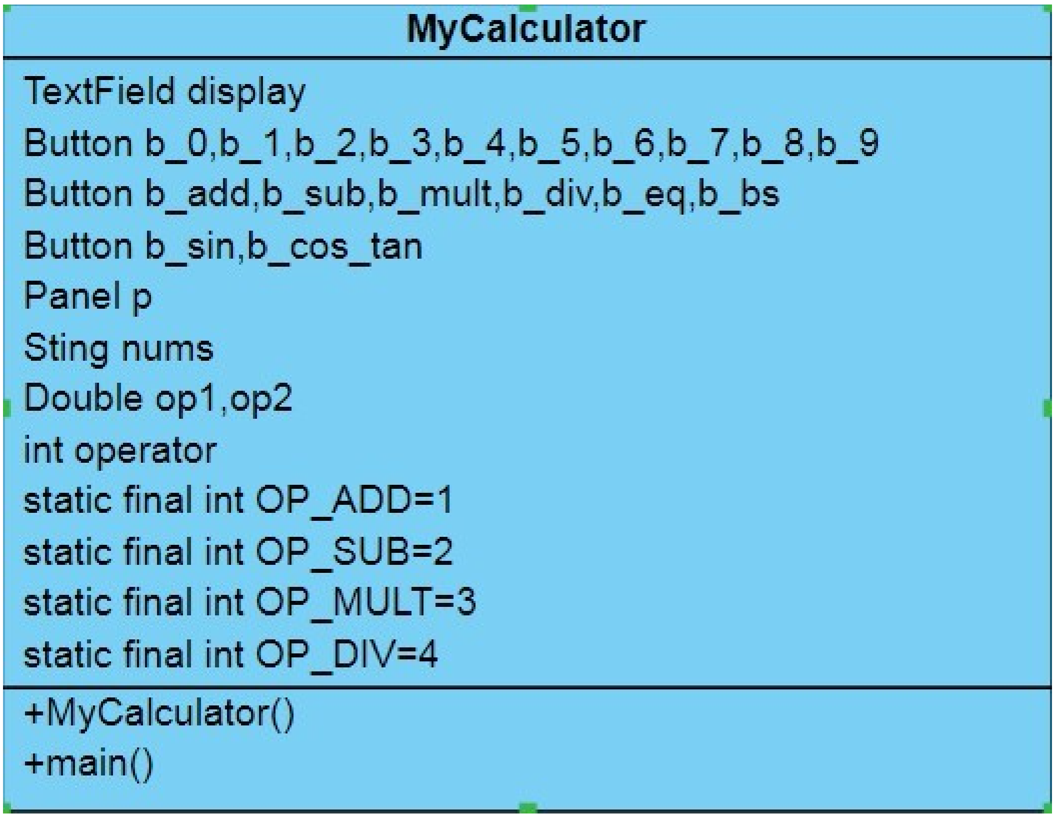
Step 2: Declare the class MyCalculator that extends Frame and implements WindowListener and Action Listener.

Step 3: Add textfield, required buttons, panel, and static members.

Step 4: In the constructor, link WindowListener, ActionListener and Panel to the class and add buttons to the panel.

Step 5: Use ActionListener to perform the required actions.

Class Diagram:



MyCalculator

TextField display
Button b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8,b_9
Button b_add,b_sub,b_mult,b_div,b_eq,b_bs
Button b_sin,b_cos_tan
Panel p
Sting nums
Double op1,op2
int operator
static final int OP_ADD=1
static final int OP_SUB=2
static final int OP_MULT=3
static final int OP_DIV=4

+MyCalculator()
+main()

Program:

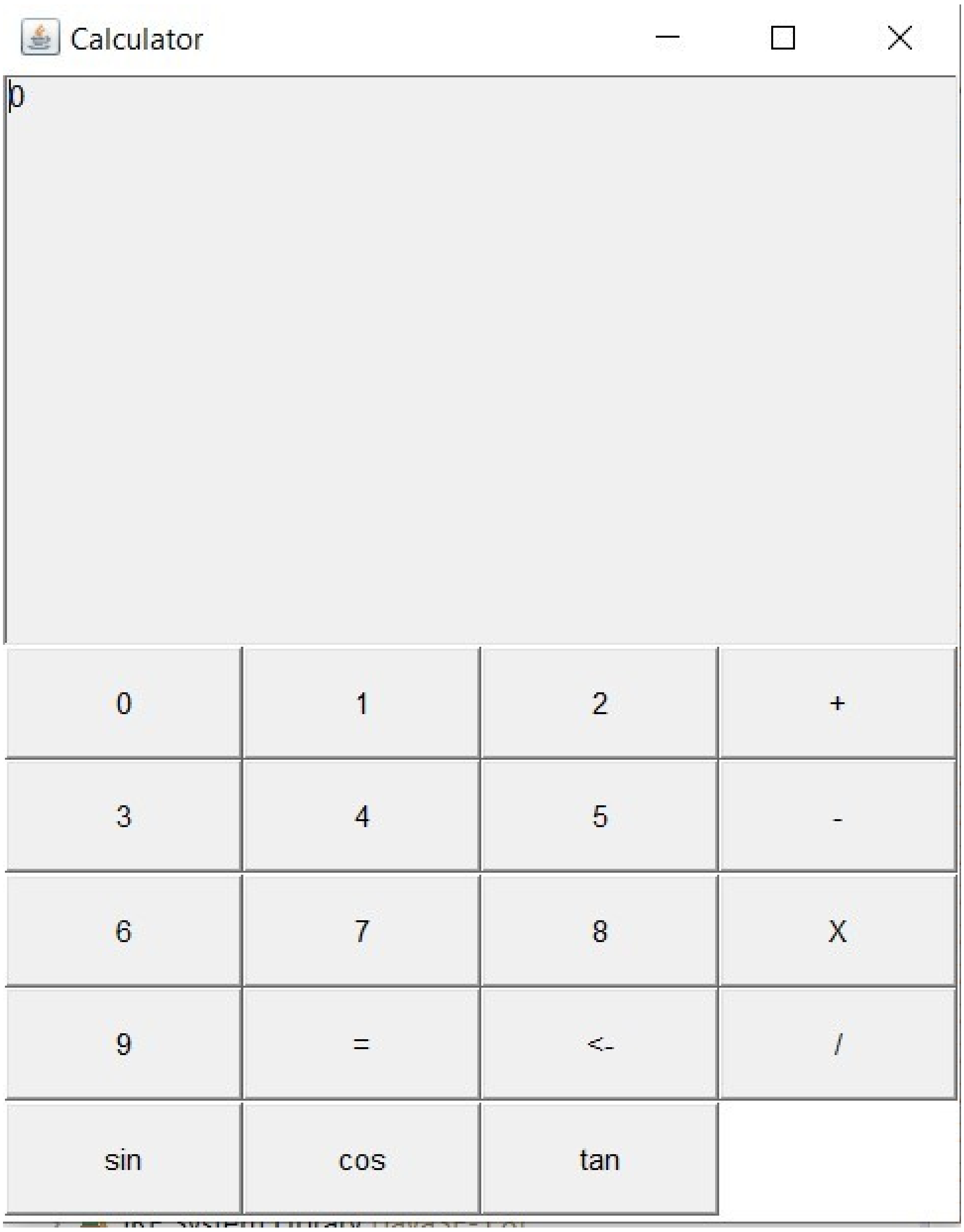
```
/***
* created by
  Nehareddy
* 212217105049
* Eee.b
*  */


package calculator;

import java.awt.*; import
java.awt.event.ActionEvent;
import
java.awt.event.ActionListener;
import
java.awt.event.WindowEvent;
import
java.awt.event.WindowListener
;

public class MyCalculator extends Frame implements
WindowListener,ActionListener{
        TextField display;
        Button b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8,b_9;
        Button b_add,b_sub,b_mul,b_div,b_eq,b_sin,b_cos,b_tan;
        Panel p;
        String nums; Double
        op1,op2; int operator;
        static final int
        OP_ADD=1; static
        final int OP_SUB=2;
        public MyCalculator()
        { this.addWindowListener(this);
                this.setLayout(new
                GridLayout(2,1)); nums="0";

                display=new TextField("0");
                display.setEditable(false);
                this.add(display);

                p=new Panel();
                p.setLayout(new
                GridLayout(3,2)); this.add(p);
```

```
b_0=new Button("0");
b_0.addActionListener(this);
p.add(b_0);

b_1=new Button("1");
b_1.addActionListener(this);
p.add(b_1);

b_2=new Button("2");
b_2.addActionListener(this);
p.add(b_2);

b_3=new Button("3");
b_3.addActionListener(this);
p.add(b_3);

b_4=new Button("4");
b_4.addActionListener(this);
p.add(b_4);

b_5=new Button("5");
b_5.addActionListener(this);
p.add(b_5);

b_6=new Button("6");
b_6.addActionListener(this);
p.add(b_6);

b_7=new Button("7");
b_7.addActionListener(this);
p.add(b_7);

b_8=new Button("8");
b_8.addActionListener(this);
p.add(b_8);

b_9=new Button("9");
b_9.addActionListener(this);
p.add(b_9);


b_add=new Button("+");
b_add.addActionListener(this);
p.add(b_add);
```

```
        b_sub=new Button("-");
        b_sub.addActionListener(this);
        p.add(b_sub);

        b_eq=new Button("=");
        b_eq.addActionListener(this);
        p.add(b_eq);

        b_div=new Button("/");
        b_div.addActionListener(this);
        p.add(b_div);

        b_mul=new Button("*");
        b_mul.addActionListener(this);
        p.add(b_mul);

        b_sin=new Button("sin");
        b_sin.addActionListener(this);
        p.add(b_sin);

        b_cos=new Button("cos");
        b_cos.addActionListener(this);
        p.add(b_cos);

        b_tan=new
Button("tan");
b_tan.addActionListener(this)
; p.add(b_tan); }


public static void main(String[] args)
        { MyCalculator mc;

        mc=new MyCalculator();
        mc.setSize(300,250);
        mc.setTitle("calculator");
        mc.setVisible(true);



}

@Override
public void
        windowOpened(WindowEvent
        e) { // TODO Auto-generated
        method stub

}

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

}

@Override
public void
        windowClosed(WindowEvent e)
        { // TODO Auto-generated
        method stub

}

@Override
public void
        windowIconified(WindowEvent e)
        { // TODO Auto-generated method
        stub

}

@Override
public void
        windowDeiconified(WindowEvent e)
        { // TODO Auto-generated method
        stub

}

@Override
public void
        windowActivated(WindowEvent e)
        { // TODO Auto-generated method
        stub

}

@Override
```

```
                    public void
 windowDeactivated(WindowEvent e) { //
   TODO Auto-generated method stubb

}

@Override
public void
        actionPerformed(ActionEvent e)
        { // TODO Auto-generated
        method stub
        if(e.getSource()==b_0)
        { nums=nums+"0
                ";
                display.setText(nums);

        }else if(e.getSource()==b_1)

        { nums=nums+"1
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_2)
        { nums=nums+"2
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_3)
        { nums=nums+"3
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_4)
        { nums=nums+"4
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_5)
        { nums=nums+"5
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_6)
        { nums=nums+"6
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_7)
        { nums=nums+"7
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_8)
        { nums=nums+"8
                ";
                display.setText(nums);
        }else
        if(e.getSource()==b_9)
        { nums=nums+"9
                ";
                display.setText(nums);
        }else if(e.getSource()==b_add)
        { op1=Double.parseDouble(nu
                ms); nums="0";
                display.setText(nums);
                operator=OP_ADD;

        }else if(e.getSource()==b_eq)
        { switch(operator
                )

                { case
                OP_ADD:
                    op2=Double.parseDouble(
                    nums); nums=""+
                    (op1+op2);
                    display.setText(nums);
                    break;

                }

                }else
                if(e.getSource()==b_sub)
                { op1=Double.parseDouble(nu
                    ms); nums="0";
                    display.setText(nums);
                    operator=OP_SUB;
                }else
                if(e.getSource()==b_eq)
                { switch(operator) { case OP_SUB:
                    op2=Double.parseDouble(num
                    s); nums=""+(op1-op2);
                    display.setText(nums); break;
                    }
                }else
                if(e.getSource()==b_sin)
                {   op1=Double.parseDouble(nums);
                    nums=""+Math.sin(op1*Math.
                    PI/180); display.setText(nums);

                }else
                if(e.getSource()==b_cos)
                {   op1=Double.parseDouble(nums);
                    nums=""+Math.cos(op1*Math.
                    PI/180); display.setText(nums);

                }else
                if(e.getSource()==b_tan)
                {   op1=Double.parseDouble(nums);
                    nums=""+Math.tan(op1*Math.
                    PI/180); display.setText(nums);

                }

                }


            }
```

Output:



Result:

Thus a java console application that uses event-driven programming paradigm of Java to design a calculator with decimal and scientific manipulations is verified.