Ex No: 10	DESIGN OF SCIENTIFIC CALCULATOR
Date: 23/09/2019	

Aim:

*To design a decimal and scientific calculator using event-driven programming paradigm of Java.

Requirements:

*Design a calculator using event-driven programming paradigm of Java with the following options.

a. Decimal manipulations,
b. Scientific manipulations.

Algorithm:

1. Create a package calc.
2. Create a class MyCalculator in the above package.
3. Declare the buttons, panel, operand and operations as attributes.
4. Declare the constructors to pass the initial attributes.
5. Import the required packages for the design of calculator.
6. Provide the necessary methods for the decimal and scientific calculations.
7. Stop.

Class Diagram:

Program:

```
/**
*

*Developed by manikanta

*Gmail:-amanikanta69@gmail.com
*/ package calc;

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_multiply,b_divide,b_equalsto,b_clear,b_delete;
  Button b_sin,b_cos,b_tan;
  Panel p;
  String nums;
  Double op1, op2;
  int op;
 static final int OP_ADD=1;
```

```
static final int OP_SUB=2;
static final int OP_MULTIPLY=3;
static final int OP_DIVIDE=4;

   public MyCalculator()
   {
     this.addWindowListener(this);
     this.setLayout(new GridLayout(2,1));
    display=new TextField("0");
     this.add(display);
     p=new Panel();
     p.setLayout(new GridLayout(4,4));
     this.add(p);
     nums="0";

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

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

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

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

   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);

b_delete= new Button("DEL");
b_delete.addActionListener(this);
 p.add(b_delete);
   }

   public static void main(String[] args) {
      // TODO Auto-generated method stub

  MyCalculator mc;
   mc=new MyCalculator();
   mc.setTitle("Calculator");
   mc.setSize(450, 400);
    mc.setBackground(Color.yellow);
    mc.setVisible(true);
   }

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

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

      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 stub

   }
    @Override
   public void actionPerformed(ActionEvent e)
{
       // TODO Auto-generated method stub

       if(e.getSource()==b_0)
       {
  nums=nums+"0";
```

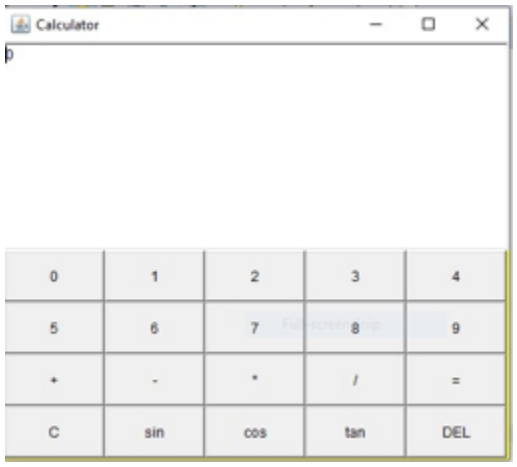
```
display.setText(nums);
                }
                 else if(e.getSource()==b_1)
    {
  nums=nums+"1";
  display.setText(nums);
    }
          else if(e.getSource()==b_2)
    {
    nums=nums+"2";
    display.setText(nums);
    }
            else if(e.getSource()==b_3)
    {
        nums=nums+"3";
        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(nums);
    op=OP_ADD;
     nums="0";
    display.setText(nums);
}
          else if(e.getSource()==b_sub)
{
    op1=Double.parseDouble(nums);
    op=OP_SUB;
     nums="0";
    display.setText(nums);

}
else if(e.getSource()==b_multiply)
{
 Op1=Double.parseDouble(nums);
  op=OP_MULTIPLY;
  nums="0";
  display.setText(nums);
}
else if(e.getSource()==b_divide)
{
    op1=Double.parseDouble(nums);
    op=OP_DIVIDE;
     nums="0";
     display.setText(nums);
}
else if(e.getSource()==b_equalsto)
{
    op2=Double.parseDouble(nums);
```

```
            switch(op)
           {
        case OP_ADD :
              nums=""+(op1+op2);
             display.setText(nums);
              break;

  case OP_SUB:
 nums=""+(op1-op2);
 display.setText(nums);
 break;

   case OP_MULTIPLY:
    nums=""+(op1*op2);
     display.setText(nums);
     break;

case OP_DIVIDE:
  nums=""+(op1/op2);
  display.setText(nums);
  break;
  }
           display.setText(nums);
        }
        else if(e.getSource()==b_clear)
        {
         nums="0";
         display.setText(nums);
        }


        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);
        }
        else if(e.getSource()==b_delete)
        {
         nums=nums;
        nums=nums.substring(0, nums.length()-1);
        display.setText(nums);
            }
        }
    }
```

Output:



Result:

*Thus, the decimal and scientific calculator is designed using event-driven programming paradigm of Java.