EXERCISE – 11 (AWT)

11 a) Write a JAVA program to paint like paint brush in applet.

Aim: To implement JAVA Program to paint like paint brush in applet

Program

MouseDrag.java

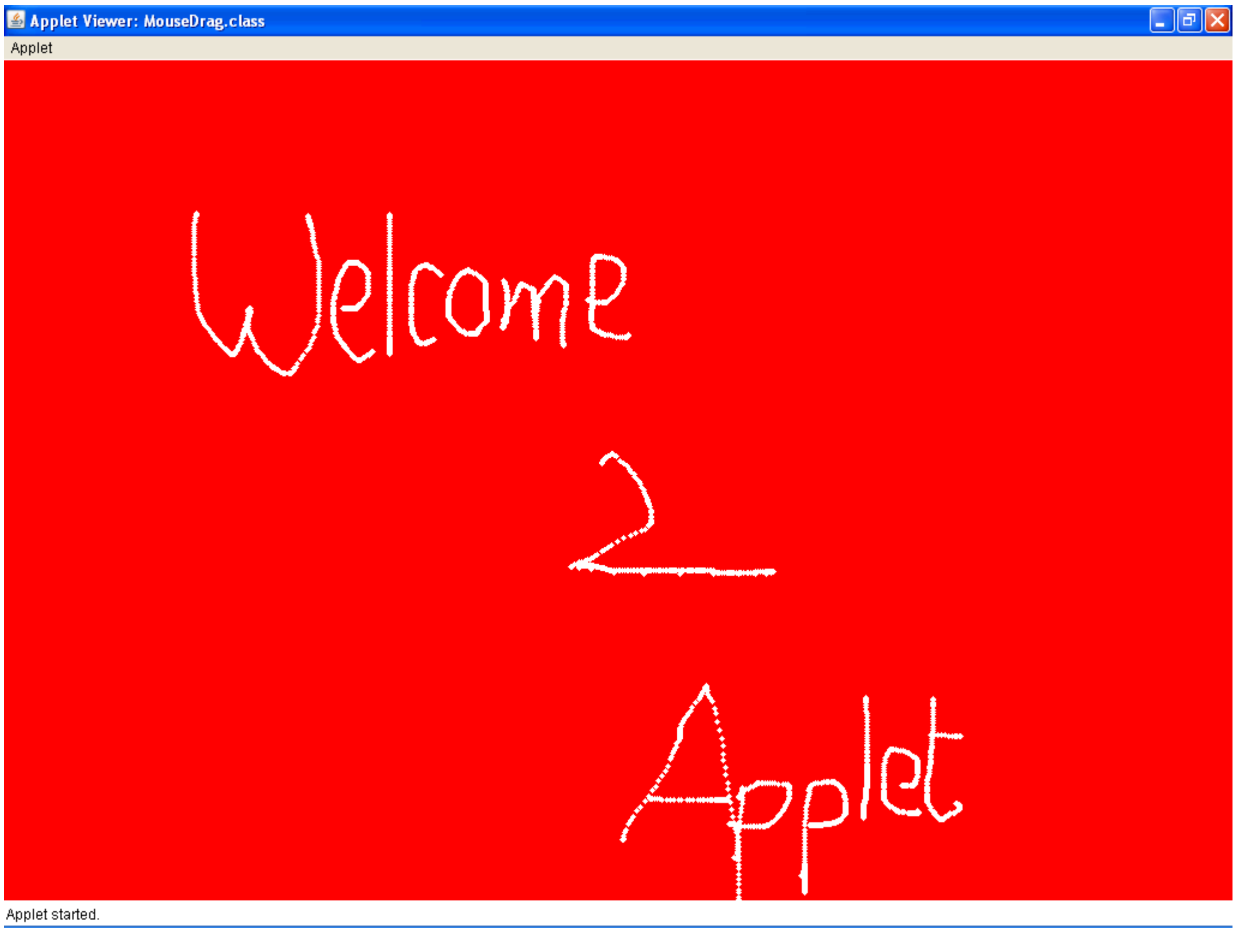
```
import java.awt.*; import
java.awt.event.*; import
java.applet.*;
public class MouseDrag extends Applet implements MouseMotionListener{

public void init(){
addMouseMotionListener(this);
setBackground(Color.red);
}

public void mouseDragged(MouseEvent me){
Graphics g=getGraphics();
g.setColor(Color.white);
g.fillOval(me.getX(),me.getY(),5,5);
}
public void mouseMoved(MouseEvent me){}

}
```

MyAppletMouseDrag.html

```
<html>
<body>
<applet code="MouseDrag.class" width="300" height="300">
</applet>
</body>
</html>
```

Output:

```
>javac MouseDrag.java

> appletviewer myapplet.html
```



11. b) Write a JAVA program that display the x and y position of the cursor movement using Mouse.

Aim: To implement JAVA program that display the x and y position of the cursor movement using Mouse

Program

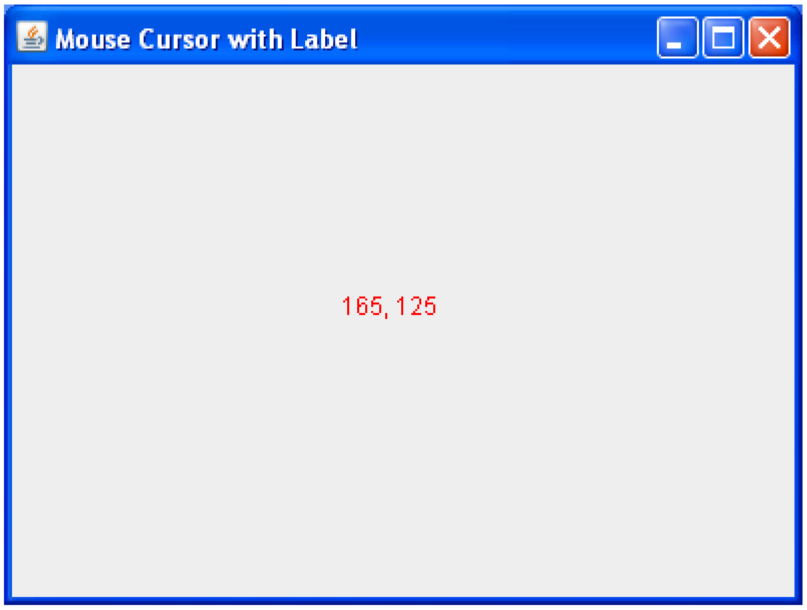
MouseCursorXYLabel.java

```
import java.awt.*; import
java.awt.event.*;
import java.awt.image.BufferedImage;
import javax.swing.*;

public class MouseCursorXYLabel extends JFrame
{

  public static void main(String[] args)
  {
   SwingUtilities.invokeLater(new Runnable()
   {
    public void run()
    {
     displayJFrame();
    }
   });
  }

  static void displayJFrame()
  {
   // create a jframe as usual JFrame
   jFrame = new JFrame();
   jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   jFrame.setTitle("Mouse Cursor with Label");

   // set the jframe size and center it
   jFrame.setPreferredSize(new Dimension(400, 300));
```

```
    jFrame.pack();
    jFrame.setLocationRelativeTo(null);

    // create an instance of my custom mouse cursor component
    final    AlsXYMouseLabelComponent    alsXYMouseLabel    =    new
AlsXYMouseLabelComponent();

    // add my component to the DRAG_LAYER of the layered pane (JLayeredPane)
    JLayeredPane layeredPane = jFrame.getRootPane().getLayeredPane();
    layeredPane.add(alsXYMouseLabel, JLayeredPane.DRAG_LAYER);
    alsXYMouseLabel.setBounds(0, 0, jFrame.getWidth(), jFrame.getHeight());

    // add a mouse motion listener, and update my custom mouse cursor with the x/y
    // coordinates as the user moves the mouse
    jFrame.addMouseMotionListener(new MouseMotionAdapter() { public
    void mouseMoved(MouseEvent me)
      {
        alsXYMouseLabel.x  =  me.getX();
        alsXYMouseLabel.y  =  me.getY();
        alsXYMouseLabel.repaint();
      }
    });

    // make the cursor a crosshair shape
    jFrame.setCursor(new Cursor(Cursor.CROSSHAIR_CURSOR));

    // display the jframe
    jFrame.setVisible(true);
  }


}

/**
 * This is the class that draws the x/y coordinates
 * near the mouse cursor/pointer.
 */
class AlsXYMouseLabelComponent extends JComponent
{
  public int x;
  public int y;
  public AlsXYMouseLabelComponent() {
    this.setBackground(Color.blue);
  }

  // use the xy coordinates to update the mouse cursor text/label
  protected void paintComponent(Graphics g)
  {
    super.paintComponent(g);
    String s = x + ", " + y;
    g.setColor(Color.red);
    g.drawString(s, x, y);
  }
}
```

Output:

```
>javac MouseCursorXYLabel.java

>java MouseCursorXYLabel
```



```
165, 125
```

11 c) Write a JAVA program to build a Calculator in AWT

Aim: To implement JAVA program to build a Calculator in AWT

Program

```
import java.awt.*;
import java.awt.event.*;
/*********************************************/

public class MyCalculator extends Frame
{

public boolean setClear=true;
double number, memValue;
char op;

String digitButtonText[] = {"7", "8", "9", "4", "5", "6", "1", "2", "3", "0", "+/-", "." };
String operatorButtonText[] = {"/", "sqrt", "*", "%", "-", "1/X", "+", "=" };
String memoryButtonText[] = {"MC", "MR", "MS", "M+" };
String specialButtonText[] = {"Backspc", "C", "CE" };

MyDigitButton digitButton[]=new MyDigitButton[digitButtonText.length];
MyOperatorButton operatorButton[]=new MyOperatorButton[operatorButtonText.length];
MyMemoryButton memoryButton[]=new MyMemoryButton[memoryButtonText.length];
MySpecialButton specialButton[]=new MySpecialButton[specialButtonText.length];

Label displayLabel=new Label("0",Label.RIGHT);
Label memLabel=new Label(" ",Label.RIGHT);

final int FRAME_WIDTH=325,FRAME_HEIGHT=325;
final int HEIGHT=30, WIDTH=30, H_SPACE=10,V_SPACE=10;
final int TOPX=30, TOPY=50;
///////////////////////////
MyCalculator(String frameText)//constructor
{
super(frameText);

int tempX=TOPX, y=TOPY;
displayLabel.setBounds(tempX,y,240,HEIGHT);
displayLabel.setBackground(Color.BLUE);
displayLabel.setForeground(Color.WHITE);
add(displayLabel);

memLabel.setBounds(TOPX,  TOPY+HEIGHT+ V_SPACE,WIDTH, HEIGHT);
add(memLabel);
```

```
// set Co-ordinates for Memory Buttons
tempX=TOPX;
y=TOPY+2*(HEIGHT+V_SPACE);
for(int i=0; i<memoryButton.length; i++)
{
memoryButton[i]=new MyMemoryButton(tempX,y,WIDTH,HEIGHT,memoryButtonText[i],
this);
memoryButton[i].setForeground(Color.RED);
y+=HEIGHT+V_SPACE;
}

//set Co-ordinates for Special Buttons
tempX=TOPX+1*(WIDTH+H_SPACE); y=TOPY+1*(HEIGHT+V_SPACE);
for(int i=0;i<specialButton.length;i++)
{
specialButton[i]=new MySpecialButton(tempX,y,WIDTH*2,HEIGHT,specialButtonText[i], th
is);
specialButton[i].setForeground(Color.RED);
tempX=tempX+2*WIDTH+H_SPACE;
}

//set Co-ordinates for Digit Buttons
int digitX=TOPX+WIDTH+H_SPACE;
int digitY=TOPY+2*(HEIGHT+V_SPACE);
tempX=digitX;  y=digitY;
for(int i=0;i<digitButton.length;i++)
{
digitButton[i]=new MyDigitButton(tempX,y,WIDTH,HEIGHT,digitButtonText[i], this);
digitButton[i].setForeground(Color.BLUE);
tempX+=WIDTH+H_SPACE;
if((i+1)%3==0){tempX=digitX; y+=HEIGHT+V_SPACE;}
}

//set Co-ordinates for Operator Buttons
int opsX=digitX+2*(WIDTH+H_SPACE)+H_SPACE;
int opsY=digitY;
tempX=opsX;  y=opsY;
for(int i=0;i<operatorButton.length;i++)
{
tempX+=WIDTH+H_SPACE;
operatorButton[i]=new MyOperatorButton(tempX,y,WIDTH,HEIGHT,operatorButtonText[i],
this);
operatorButton[i].setForeground(Color.RED);
if((i+1)%2==0){tempX=opsX; y+=HEIGHT+V_SPACE;}
}

addWindowListener(new WindowAdapter()
{
```

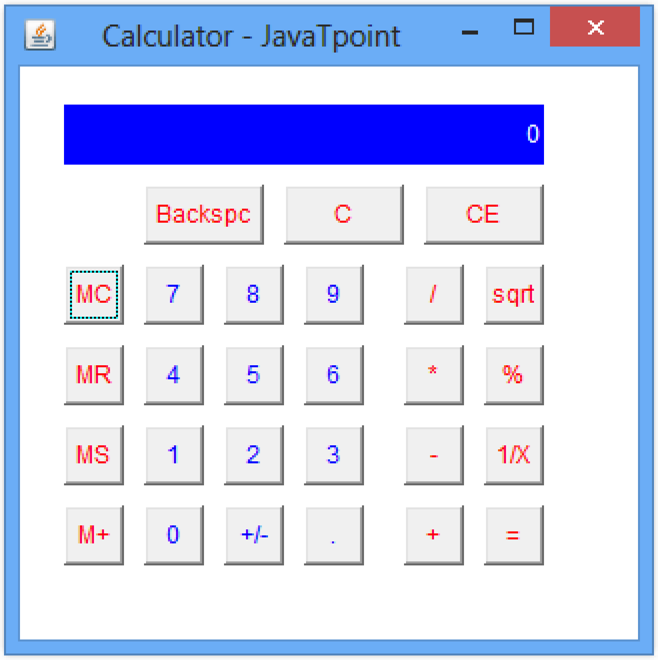
```
public void windowClosing(WindowEvent ev)
{System.exit(0);}
});

setLayout(null);
setSize(FRAME_WIDTH,FRAME_HEIGHT);
setVisible(true);
}
//////////////////////////////////
static String getFormattedText(double temp)
{
String resText=""+temp;
if(resText.lastIndexOf(".0")>0)
   resText=resText.substring(0,resText.length()-2);
return resText;
}
////////////////////////////////////////
public static void main(String []args)
{
new MyCalculator("Calculator - JavaTpoint");
}
}

/*******************************************/

class MyDigitButton extends Button implements ActionListener
{
MyCalculator cl;

//////////////////////////////////////////
MyDigitButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}  ////////////////////////////////////////////////
static boolean isInString(String s, char ch)
{
for(int i=0; i<s.length();i++) if(s.charAt(i)==ch) return true;
return false;
}
/////////////////////////////////////////////////
public void actionPerformed(ActionEvent ev)
{
String tempText=((MyDigitButton)ev.getSource()).getLabel();
```

```
if(tempText.equals("."))
{
 if(cl.setClear)
    {cl.displayLabel.setText("0.");cl.setClear=false;}
 else if(!isInString(cl.displayLabel.getText(),'.'))
    cl.displayLabel.setText(cl.displayLabel.getText()+".");
 return;
}

int index=0;
try{
        index=Integer.parseInt(tempText);
    }catch(NumberFormatException e){return;}

if (index==0 && cl.displayLabel.getText().equals("0")) return;

if(cl.setClear)
            {cl.displayLabel.setText(""+index);cl.setClear=false;}
else
    cl.displayLabel.setText(cl.displayLabel.getText()+index);
}//actionPerformed
}//class defination

/********************************************/

class MyOperatorButton extends Button implements ActionListener
{
MyCalculator cl;

MyOperatorButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
///////////////////////
public void actionPerformed(ActionEvent ev)
{
String opText=((MyOperatorButton)ev.getSource()).getLabel();

cl.setClear=true;
double temp=Double.parseDouble(cl.displayLabel.getText());

if(opText.equals("1/x"))
    {
    try
```

```
        {double tempd=1/(double)temp;
        cl.displayLabel.setText(MyCalculator.getFormattedText(tempd));}
    catch(ArithmeticException excp)
                    {cl.displayLabel.setText("Divide by 0.");}
    return;
    }
if(opText.equals("sqrt"))
    {
    try
        {double tempd=Math.sqrt(temp);
        cl.displayLabel.setText(MyCalculator.getFormattedText(tempd));}
            catch(ArithmeticException excp)
                    {cl.displayLabel.setText("Divide by 0.");}
    return;
    }
if(!opText.equals("="))
    {
    cl.number=temp;
    cl.op=opText.charAt(0);
    return;
    }
// process = button pressed
switch(cl.op)
{
case '+':
    temp+=cl.number;break;
case '-':
    temp=cl.number-temp;break;
case '*':
    temp*=cl.number;break;
case '%':
    try{temp=cl.number%temp;}
    catch(ArithmeticException excp)
        {cl.displayLabel.setText("Divide by 0."); return;}
    break;
case '/':
    try{temp=cl.number/temp;}
        catch(ArithmeticException excp)
                {cl.displayLabel.setText("Divide by 0."); return;}
    break;
}//switch

cl.displayLabel.setText(MyCalculator.getFormattedText(temp));
//cl.number=temp;
}//actionPerformed
}//class

/****************************************/
```

```
class MyMemoryButton extends Button implements ActionListener
{
MyCalculator cl;

////////////////////////////////
MyMemoryButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
////////////////////////////////////////////////
public void actionPerformed(ActionEvent ev)
{
char memop=((MyMemoryButton)ev.getSource()).getLabel().charAt(1);

cl.setClear=true;
double temp=Double.parseDouble(cl.displayLabel.getText());

switch(memop)
{
case 'C':
   cl.memLabel.setText(" ");cl.memValue=0.0;break;
case 'R':
   cl.displayLabel.setText(MyCalculator.getFormattedText(cl.memValue));break;
case 'S':
   cl.memValue=0.0;
case '+':
   cl.memValue+=Double.parseDouble(cl.displayLabel.getText());
   if(cl.displayLabel.getText().equals("0") || cl.displayLabel.getText().equals("0.0")  )
      cl.memLabel.setText(" ");
   else
      cl.memLabel.setText("M");
   break;
}//switch
}//actionPerformed
}//class

/****************************************/

class MySpecialButton extends Button implements ActionListener
{
MyCalculator cl;

MySpecialButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
////////////////////
static String backSpace(String s)
{
String Res="";
for(int i=0; i<s.length()-1; i++) Res+=s.charAt(i);
return Res;
}

//////////////////////////////////////////////////////////
public void actionPerformed(ActionEvent ev)
{
String opText=((MySpecialButton)ev.getSource()).getLabel();
//check for backspace button
if(opText.equals("Backspc"))
{
String tempText=backSpace(cl.displayLabel.getText());
if(tempText.equals(""))
   cl.displayLabel.setText("0");
else
   cl.displayLabel.setText(tempText);
return;
}
//check for "C" button i.e. Reset
if(opText.equals("C"))
{
cl.number=0.0; cl.op=' '; cl.memValue=0.0;
cl.memLabel.setText(" ");
}

//it must be CE button pressed
cl.displayLabel.setText("0");cl.setClear=true;
}//actionPerformed
}//class
```

Output:


Calculator - JavaTpoint
0
Backspc
C
CE
MC
7
8
9
/
sqrt
MR
4
5
6
*
%
MS
1
2
3
-
1/X
M+
0
+/-
.
+
=