**Q1) List syntax of all windowListener Events.**

**Ans:-**

The following are the commonly used methods in Java WindowListener.

windowActivated(WindowEvent ev): This method will be invoked when the window is set as active.

windowDeiconified(WindowEvent ev): This method will be invoked when there is a change in the size of the window from minimal to normal state.

windowClosing(WindowEvent ev): This method will be invoked when the user tries to get the window closed using the system menu in the window.

windowClosed(WindowEvent ev): This method will be invoked when the window gets closed due to the calling of disposing in the window.

windowDeactivated(WindowEvent ev): This method will be invoked when the window is not active anymore.

windowOpened(WindowEvent e): This method will get invoked when a window is visible for the first time.

windowIconified(WindowEvent e): This method will be invoked when there is a change in the size of the window from normal to a small state.

**Q2) Write a program to display a line, Rectangle and filled Circle.**

**Ans:-**

package com.mkyong;

import java.awt.Dimension;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Rectangle;

import java.awt.Shape;

import java.awt.geom.Ellipse2D;

import java.awt.geom.Line2D;

import java.awt.geom.RoundRectangle2D;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.SwingUtilities;

public class DrawShapes extends JFrame {

private static final long serialVersionUID = 1L;

public DrawShapes() {

setSize(new Dimension(320, 320));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

JPanel p = new JPanel() {

@Override

public void paintComponent(Graphics g) {

Graphics2D g2 = (Graphics2D) g;

Shape line = new Line2D.Double(3, 3, 303, 303);

Shape rect = new Rectangle(3, 3, 303, 303);

Shape circle = new Ellipse2D.Double(100, 100, 100, 100);

Shape roundRect = new RoundRectangle2D.Double(20, 20, 250, 250, 5, 25);

g2.draw(line);

g2.draw(rect);

g2.draw(circle);

g2.draw(roundRect);

}

};

setTitle("My Shapes");

this.getContentPane().add(p);

}

public static void main(String arg[]) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

// TODO Auto-generated method stub

new DrawShapes();

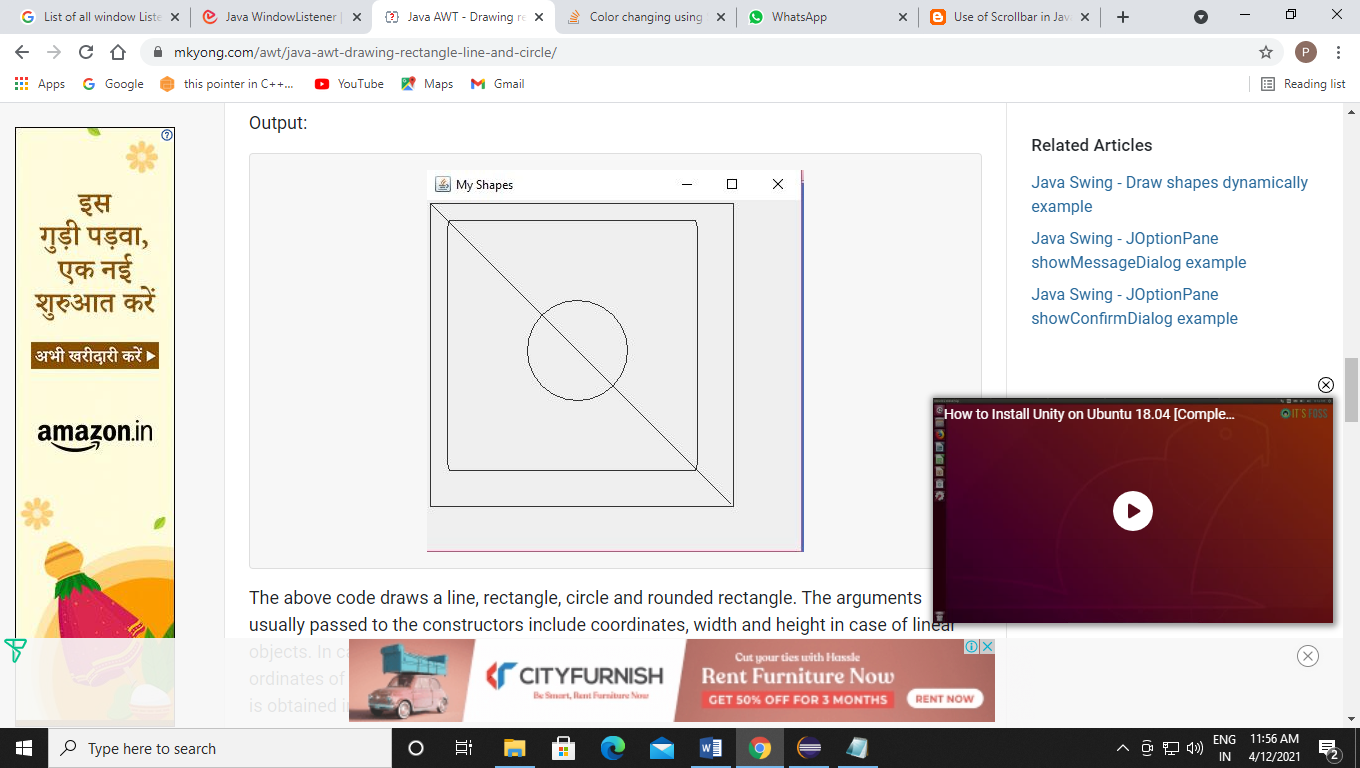
}

});

}

}

Output:-



**Q3) Write a program to change the color of a panel using a scrollbar.**

**Ans:-**

ScrollColor.java

import java.awt.event.AdjustmentEvent;

import java.awt.FlowLayout;

import java.awt.Color;

import javax.swing.JButton;

import javax.swing.JScrollBar;

import javax.swing.JLabel;

import javax.swing.JFrame;

class SetScrollBar extends JFrame implements AdjustmentListener{

private JLabel labelRed, labelGreen, labelBlue;

private JScrollBar sBarRed, sBarGreen, sBarBlue;

public SetScrollBar(){

//setting layout

setLayout(new FlowLayout());

//initializing label for each scrollbar;

labelRed = new JLabel("Adjust Red:");

labelGreen = new JLabel("Adjust Green:");

labelBlue = new JLabel("Adjust Blue:");

//initializing scrollbar

sBarRed = new JScrollBar(JScrollBar.HORIZONTAL,0,5,0,255);

sBarGreen = new JScrollBar(JScrollBar.HORIZONTAL,0,5,0,255);

sBarBlue = new JScrollBar(JScrollBar.HORIZONTAL,0,5,0,255);

//adding eventlistener

sBarRed.addAdjustmentListener(this);

sBarGreen.addAdjustmentListener(this);

sBarBlue.addAdjustmentListener(this);

//adding label and scrollbar to the frame;

add(labelRed);

add(sBarRed);

add(labelGreen);

add(sBarGreen);

add(labelBlue);

add(sBarBlue);

}//end of constructor;

public void adjustmentValueChanged(AdjustmentEvent event){

int cRed, cGreen, cBlue; //store value of scrollbar

cRed = sBarRed.getValue(); //getting current value of scroll bar;

cGreen = sBarGreen.getValue();

cBlue = sBarBlue.getValue();

getContentPane().setBackground(new Color(cRed,cGreen,cBlue)); //change the background color;

}//end of event handler

} //end of SetScrollBar class

public class ScrollColor{

public static void main(String[] args){

SetScrollBar demo = new SetScrollBar();

demo.setSize(200,300);

demo.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

demo.setVisible(true);

}

}

Output:-

