Practical:06:canva

MainActivity.kt:

```
package com.example.h6
import android.annotation.SuppressLint
import android.graphics.Bitmap
import android.graphics.Canvas
import android.graphics.Color
import android.graphics.Paint
import android.os.Build
import android.os.Bundle
import android.view.MotionEvent
import android.view.View
import android.widget.ImageView
import androidx.annotation.RequiresApi
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity(), View.OnTouchListener {
    // Declaring ImageView, Bitmap, Canvas, Paint, Down Coordinates and Up
Coordinates
    private lateinit var mImageView: ImageView
    private lateinit var bitmap: Bitmap
    private lateinit var canvas: Canvas
    private lateinit var paint: Paint
    private var downX = 0f
    private var downY = 0f
    private var upX = 0f
    private var upY = 0f

    @RequiresApi(Build.VERSION_CODES.R)
    @SuppressLint("ClickableViewAccessibility")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Initializing the ImageView
        mImageView = findViewById(R.id.image_view_1)

        // Get the current window dimensions
        val currentDisplay = windowManager.currentWindowMetrics
        val dw = currentDisplay.bounds.width()
        val dh = currentDisplay.bounds.height()

        // Creating a bitmap with fetched dimensions
        bitmap = Bitmap.createBitmap(dw, dh, Bitmap.Config.ARGB_8888)
        // Storing the canvas on the bitmap
        canvas = Canvas(bitmap)

        // Initializing Paint to set stroke attributes like color and size
        paint = Paint().apply {
            color = Color.RED
            strokeWidth = 10f
        }

        // Setting the bitmap on ImageView
        mImageView.setImageBitmap(bitmap)

        // Setting onTouchListener on the ImageView
```

```
        mImageView.setOnTouchListener(this)
    }

    // When touch is detected on the ImageView, initial and final
coordinates are recorded,
    // and a line is drawn between them. The ImageView is updated.
    @SuppressLint("ClickableViewAccessibility")
    override fun onTouch(v: View?, event: MotionEvent?): Boolean {
        when (event!!.action) {
            MotionEvent.ACTION_DOWN -> {
                downX = event.x
                downY = event.y
            }
            MotionEvent.ACTION_MOVE -> {
                upX = event.x
                upY = event.y
                canvas.drawLine(downX, downY, upX, upY, paint)
                downX = upX // Update start point for continuous drawing
                downY = upY
                mImageView.invalidate() // Redraw the ImageView
            }
            MotionEvent.ACTION_UP -> {
                upX = event.x
                upY = event.y
                canvas.drawLine(downX, downY, upX, upY, paint)
                mImageView.invalidate() // Redraw the ImageView
            }
        }
        return true // Ensure event is consumed
    }
}
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/image_view_1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/white"
        tools:ignore="ContentDescription" />
</RelativeLayout>
```