University of Kalyani
Department of Computer Science & Engineering

Suraj Mandal
M.Tech in Computer Science and Engineering
Semester- III, Roll No: 90/CSE/200014
Paper: RS-GIS Lab

INDEX

No.	Program name
1	Write a program to assign three features in RS-GIS image
2	Write a GIS based program to draw a polygon(four different places with 1 cm,7cm,9cm width) into a GIS image
3	Write a program to draw line(2cm,6cm,9cm) in three different places and point into a GIS image
4	Write a program to show various processes to open a GIS image into a desktop (at least three) with resizing it, mark some portion of the image with some color and point it with an attribute.
5	Write a program to cascade multiple designed pages(at least three) with a resizing tool, take some tools to input and output data.
6	Write a program to assign three features in an RS-GIS image. Compute how many segment has features of a type
7	Write a GIS based program to draw a polygon(four different places with 5 cm,7cm,9cm width) into a GIS image and designate the mark with some attribute
8	Write a program to draw lines(4cm,5cm,6cm) in three different places and point into a GIS image so that they can form a triangle.
9	Write a program to show various processes to open GIS images into desktop(at least three) with resizing tools.
10	Write a program to cascade multiple designed pages(at least three) with resizing tool
11	Draw Point, Line, Polygon, text, oval and gradient on the image.
12	Write a program to perform the following operations: click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the corresponding text field...session1.gif image has four features they are.. 1. Vegetation 2. water body 3. moist land 4. bare soil
13	Write a GIS based program to draw a polygon(four different places with 2 cm,6cm,8cm width) into a GIS image
14	Write a GIS application program that will do the following : a. it will open an image in a panel. b. an option will be there for labeling regions in the current image depending on the pixel color. c. after completion of step 2, the user can ask for the percentage of a particular region in the current image. Program should be able to show the percentage value of the region with respect to the whole image. d. the above stated problem in step 3, should be implemented for an image where the same color pixels are scattered.

15	Write a program to show various processes to open a GIS image into a desktop(at least three) with resizing it, mark some portion of the image with some color and point it with an attribute.
16	Write a program to perform the following operations: click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the corresponding text field...session1.gif image has four features they are.. 1. Rock 2. River 3. moist land 4. bare soil
17	Write a program to draw lines(6cm,8cm,10cm) in three different places and point them into a GIS image so that they can form a triangle.
18	Write a program to perform the following operations: click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the corresponding text field...session1.gif image has four features they are.. 1. Vegetation 2. water body 3. moist land 4. bare soil
19	Write a GIS based program to draw a polygon(four different places with 4 cm,8cm,12cm width) into a GIS image
20	Write a program to perform the following operations: click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the corresponding text field...session1.gif image has four features they are. 1. Vegetation 2. water body 3. moist land 4. bare soil

1. Write a program to assign three features in RS-GIS image.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;

class page1 extends JFrame implements MouseListener,MouseMotionListener
,ActionListener
{
        JFrame f;
        JLabel l;
        JPanel p1;
        ImageIcon ii;
        Image img;
        int height;
        int width;
        Container c;
        int pixels[];
        PixelGrabber pg;
        JPanel panel;
        JLabel leb1,leb2,leb3;
        JComboBox cb1a;
        JTextField tf1,tf2;
        JButton fcb,cb;
        Color cc=new Color(10,90,60,200);
        int x,y;
        int rgb;
        int frgb[]=new int[2];
        String fea[]=new String[2];
        int count=0;
        String ss;
        int ccb=0;

        page1()
        {
                f=new JFrame("Data Association Page");
                ii=new ImageIcon("session1.gif");
                img=ii.getImage();
                height=ii.getIconHeight();
                width=ii.getIconWidth();
                pixels=new int[ii.getIconWidth()*ii.getIconHeight()];
```

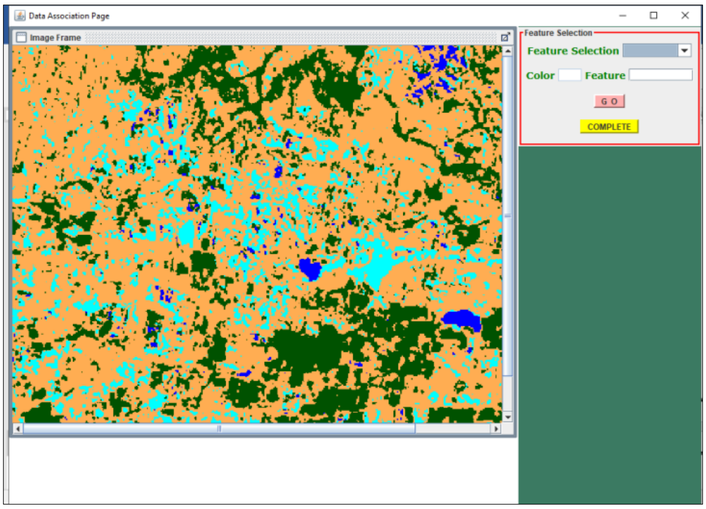
```
		pg=new
PixelGrabber(img,0,0,ii.getIconWidth(),ii.getIconHeight(),pixels,0,ii.getIconWidth());
		try
		{
			pg.grabPixels();
		}
		catch(InterruptedException k)
		{
		}
		l=new JLabel(ii,JLabel.CENTER);
		c=f.getContentPane();
		JDesktopPane desk = new JDesktopPane();

	JInternalFrame p = new JInternalFrame("Image Frame",false, false,  true,
false);
		JScrollPane scroll = new JScrollPane(l);
		p.setContentPane(scroll);
		p.setBounds(0, 0, 740, 600);
		desk.add(p);
		p.setVisible(true);
		l.addMouseListener(this);
		l.addMouseMotionListener(this);
		panel=new JPanel();
		panel.setLayout(new GridLayout(4,1));
		panel.setBackground(cc);

		p1=new JPanel();
	    p1.setBorder(new TitledBorder(new LineBorder(Color.red,2),"Feature
Selection"));
	    p1.setLayout(new GridLayout(4,1));
	    panel.add(p1);
	    /*--------------------*/
	    JPanel p1a=new JPanel();
	    String ht1 =
	"<html><p><font color=\"GREEN\" "+
	"size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
		leb1=new JLabel(ht1);
		p1a.add(leb1);

		cb1a=new JComboBox();
		cb1a.setPreferredSize(new Dimension (100,20) );
	    cb1a.setEditable(false);
	    cb1a.setBackground(Color.WHITE);
	    cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
	    cb1a.addItem("          ");
```

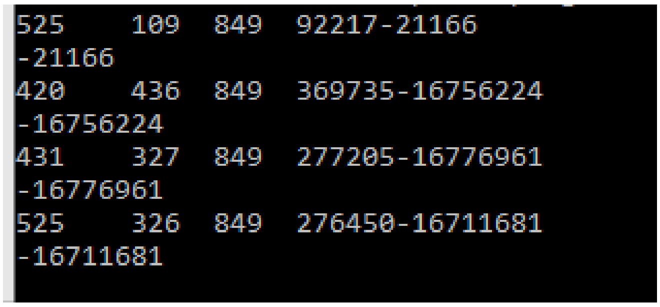
```
    //cb1a.addActionListener(this);
     p1a.add(cb1a);
          p1.add(p1a);

          /*--------------------*/
          JPanel p1b=new JPanel();
          String ht2 =
"<html><p><font color=\"GREEN\" "+
"size=\"4\" face=\"Verdana\">Color</font> </p>";
          leb2=new JLabel(ht2);
          p1b.add(leb2);
          tf1=new JTextField(3);
     tf1.setFont(new Font("TIMES NEW ROMAN",Font.BOLD,12));
     tf1.setEditable(false);
     tf1.setBackground(Color.white);
     p1b.add(tf1);

 /*--------------------*/

          String ht3 =
"<html><p><font color=\"GREEN\" "+
"size=\"4\" face=\"Verdana\">Feature</font> </p>";
          leb3=new JLabel(ht3);
          p1b.add(leb3);
          tf2=new JTextField(9);
     tf2.setFont(new Font("TIMES NEW ROMAN",Font.BOLD,12));
     p1b.add(tf2);

     p1.add(p1b);
     /*--------------------*/

          JPanel p1c=new JPanel();
          fcb=new JButton("   G  O   ");
          fcb.setBackground(Color.pink);
          fcb.setBorder(new BevelBorder (BevelBorder.RAISED));
          fcb.addActionListener(this);
          fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
          p1c.add(fcb);

          p1.add(p1c);
           /*--------------------*/
          JPanel p1d=new JPanel();
          cb=new JButton("   COMPLETE   ");
          cb.setBackground(Color.yellow);
          cb.setBorder(new BevelBorder (BevelBorder.RAISED));
          cb.addActionListener(this);
```

```
                cb.setBorder(new BevelBorder(BevelBorder.RAISED));
                p1d.add(cb);

                p1.add(p1d);
                 /*---------------------*/
                c.add(desk, BorderLayout.CENTER);
                c.add(panel,BorderLayout.EAST);
           f.setSize(1024,738);
                f.setVisible(true);

        }

        public static void main(String args[])
{
                new page1();
}

public void mouseClicked(MouseEvent me)
{
  ColorModel md=pg.getColorModel();
  x=me.getX();
  y=me.getY();
  int a1=(y-1)*ii.getIconWidth()+x;
  System.out.println(x+"    "+y+"  "+ii.getIconWidth()+"  "+a1+pixels[a1]);
 // System.out.println(a1);
  rgb=(md.getRGB(pixels[a1]));
  System.out.println(rgb);
  tf1.setBackground(new Color(rgb));
}

public void mouseEntered(MouseEvent me)
{

}
public void mouseExited(MouseEvent me)
{

}

public void mousePressed(MouseEvent me)
{

}
public void mouseReleased(MouseEvent me)
{
}

public void mouseMoved(MouseEvent me)
{

}
public void mouseDragged(MouseEvent me)
{

}
public void actionPerformed(ActionEvent ae)
{

if(ae.getSource()==fcb)
  {
        int flag=0;

        if(count==2)
        {
                JOptionPane.showMessageDialog((Component)null,"Assignment
compleated before",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                tf1.setBackground(Color.white);
                tf2.setText(null);

        }
        ss=tf2.getText();
        if(ss.length()==0&&flag==0)
          {
                JOptionPane.showMessageDialog((Component)null,"Please Assign
Feature Name",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
          }
          if(rgb==0&&flag==0)
          {
                JOptionPane.showMessageDialog((Component)null,"Please Select
Feature",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
          }

          if(flag==0)
          {
          for(int i=0;i<count;i++)
          {
                if(rgb==frgb[i])
                {
                 JOptionPane.showMessageDialog((Component)null,"Please Select New
Feature",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }

                if(ss.equalsIgnoreCase(fea[i]))
                {
                  JOptionPane.showMessageDialog((Component)null,"Feature Name
Used Before",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                tf2.setText(null);
                }
          }
          }
          if(flag==0)
          {
                frgb[count]=rgb;
                //System.out.println(rgb);
                fea[count]=ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb=0;
          }
          if(count==2&&flag==0)
          {
                JOptionPane.showMessageDialog((Component)null,"Assignment
compleated",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                tf1.setBackground(Color.white);
                tf2.setText(null);
                tf2.setEditable(false);
          }
  }

  if(ae.getSource()==cb)
  {
        if(count==2&& ccb==0)
        {
         for(int i=0;i<2;i++)
         {
```

```
                cb1a.addItem(fea[i]);
                System.out.println(frgb[i]+"   is for the feature "+fea[i]);
            }
            ccb=1;
          }
      }
  }
}
```

OUTPUT:





```
525     109   849   92217-21166
-21166
420     436   849   369735-16756224
-16756224
431     327   849   277205-16776961
-16776961
525     326   849   276450-16711681
-16711681
```

-16756224 is for the feature vegetation
-21166 is for the feature soil
-16776961 is for the feature water body
-16711681 is for the feature moist land

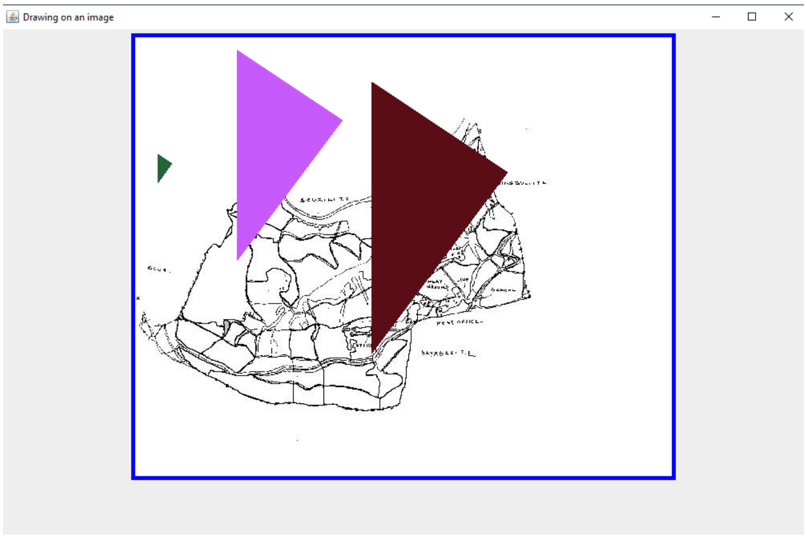
2. Write a GIS based program to draw a polygon (four different places with 1 cm,7cm,9cm width) into a GIS image.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    public static double getDPI(){
        double dpi = Toolkit.getDefaultToolkit().getScreenResolution();
        // System.out.println("Screen DPI: " + dpi);
        return dpi;
    }

    public static double cmToPixel(double cm){
        double dpi = getDPI();
        double pixel = cm * dpi / 2.54;
        // System.out.println("cm: " + cm + " pixel: " + pixel);
        return pixel;
    }

    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
          pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);
    }

    public void mouseClicked(MouseEvent me) {
        gg = (Graphics2D) l.getGraphics();

        // 1cm 7cm 9cm polygon
        Polygon p1 = drawPolygon(cmToPixel(1), 33, 150);
        gg.setColor(new Color(33, 100, 56));
        gg.fillPolygon(p1);

        Polygon p2 = drawPolygon(cmToPixel(7), 132, 20);
        gg.setColor(new Color(200, 90, 250));
        gg.fillPolygon(p2);

        Polygon p3 = drawPolygon(cmToPixel(9), 300, 60);
        gg.setColor(new Color(89, 13, 20));
        gg.fillPolygon(p3);
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }
```

```
    public void mouseReleased(MouseEvent me) {
    }

    public Polygon drawPolygon (double width, double x, double y) {
        Polygon p = new Polygon();
        p.addPoint((int)x, (int)y);
        p.addPoint((int)(x), (int)(y+ width));
        p.addPoint((int)(x + width/2), (int)(y + width/3));
        return p;
    }

    public void mouseMoved(MouseEvent me) {

    }
    public void mouseDragged(MouseEvent me) {
    }

    public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:

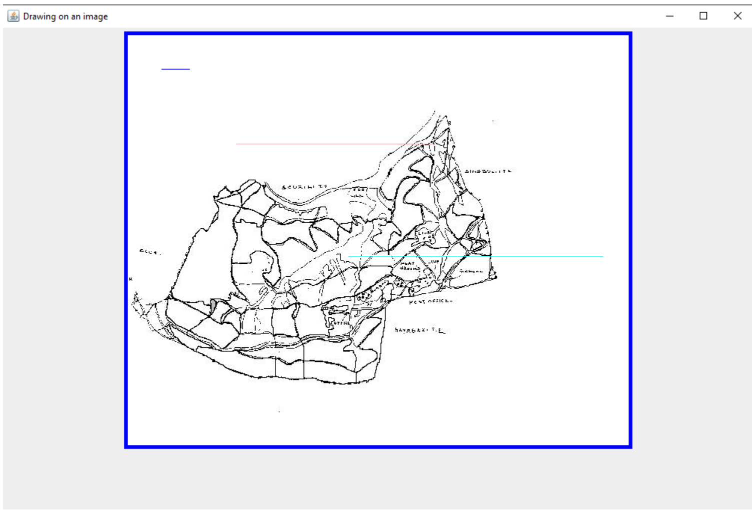


3. Write a program to draw line(2cm,6cm,9cm) in three different places and point into a GIS image.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    public static double getDPI(){
        double dpi = Toolkit.getDefaultToolkit().getScreenResolution();
        // System.out.println("Screen DPI: " + dpi);
        return dpi;
    }

    public static double cmToPixel(double cm){
        double dpi = getDPI();
        double pixel = cm * dpi / 2.54;
        // System.out.println("cm: " + cm + " pixel: " + pixel);
        return pixel;
    }
    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);
    }

    public void mouseClicked(MouseEvent me) {

        ColorModel cm = pg.getColorModel();
        gg = (Graphics2D) l.getGraphics();

        int startX = 50;
        int startY = 50;

        // Drawing line 2cm, 6cm, 9cm long
        gg.setColor(Color.blue);
        gg.drawLine(startX, startY, (int)(startX+cmToPixel(1)), startY);

        startX = 150;
        startY = 150;
        gg.setColor(Color.pink);
        gg.drawLine(startX, startY, (int)(startX+cmToPixel(7)), startY);

        startX = 300;
        startY = 300;
        gg.setColor(Color.cyan);
        gg.drawLine(startX, startY, (int)(startX+cmToPixel(9)), startY);
    }

    public void mouseEntered(MouseEvent me) {
    }
    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }

    public void mouseReleased(MouseEvent me) {
    }

    public void mouseMoved(MouseEvent me) {
    }
    public void mouseDragged(MouseEvent me) {
    }



     public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:



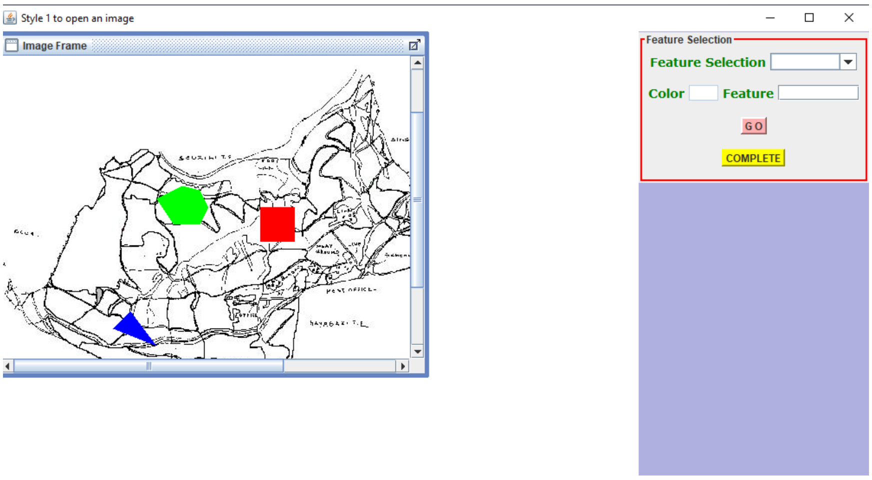
4. Write a program to show various process to open GIS image into desktop (at least three) with resizing it mark some portion of the image with some color and point it with attribute.

PROGRAM:

Option 1:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.JButton;
import javax.swing.JFrame;

class option1 extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
    Graphics2D gg1, gg2, gg3;
    Polygon pp1 = new Polygon();
    Polygon pp2 = new Polygon();
    Polygon pp3 = new Polygon();
    boolean startHovercurrent, startHoverprev = false;

    option1() {
        f = new JFrame("Style 1 to open an image");
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 500, 400);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        //
        pp1.addPoint(300, 300);
        pp1.addPoint(340, 300);
        pp1.addPoint(340, 260);
        pp1.addPoint(300, 260);
        pp2.addPoint(200, 280);
        pp2.addPoint(230, 280);
        pp2.addPoint(240, 260);
        pp2.addPoint(230, 240);
        pp2.addPoint(210, 235);
        pp2.addPoint(180, 250);
        pp3.addPoint(130, 400);
        pp3.addPoint(180, 420);
        pp3.addPoint(150, 380);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem(" ");
        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);
        /*--------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        tf1.setEditable(false);
        tf1.setBackground(Color.white);
        p1b.add(tf1);
        /*--------------------*/
        String ht3 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature</font> </p>";
        leb3 = new JLabel(ht3);
        p1b.add(leb3);
        tf2 = new JTextField(9);
        tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        p1b.add(tf2);
        p1.add(p1b);
        /*--------------------*/
        JPanel p1c = new JPanel();
        fcb = new JButton(" G O ");
        fcb.setBackground(Color.pink);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        fcb.addActionListener(this);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1c.add(fcb);
        p1.add(p1c);
        /*--------------------*/
        JPanel p1d = new JPanel();
        cb = new JButton(" COMPLETE ");
        cb.setBackground(Color.yellow);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        cb.addActionListener(this);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1d.add(cb);
        p1.add(p1d);
        /*--------------------*/
        c.add(desk, BorderLayout.CENTER);
        c.add(panel, BorderLayout.EAST);
        f.setSize(1024, 738);
        f.setVisible(true);
    }

    public static void main(String args[]) {
        new option1();
    }

    public void mouseClicked(MouseEvent me) {
        boolean contain1, contain2, contain3;
        ColorModel md = pg.getColorModel();
        x = me.getX();
        y = me.getY();
        contain1 = pp1.contains(x, y);
        contain2 = pp2.contains(x, y);
        contain3 = pp3.contains(x, y);
        System.out.print("\nPoint inside polygon:" + (contain1 || contain2 || contain3) +
"\n");
        int a1 = (y - 1) * ii.getIconWidth() + x;
        System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + "" + pixels[a1]);
        rgb = (md.getRGB(pixels[a1]));
        System.out.print(rgb);
        if (contain1 == true) {
            Color color = new Color(255, 0, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(255, 0, 0));
        } else if (contain2 == true) {
            Color color = new Color(0, 255, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 255, 0));
        } else if (contain3 == true) {
            Color color = new Color(0, 0, 255);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 0, 255));
        } else
            tf1.setBackground(new Color(rgb));
        for (int i = 0; i < count; i++) {
            if (rgb == frgb[i]) {
                System.out.print(" " + fea[i] + "\n\n");
                break;
            }
        }
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }

    public void mouseReleased(MouseEvent me) {
    }

    public void mouseMoved(MouseEvent me) {
        // fcb.setToolTipText("HI");
        ColorModel cm = pg.getColorModel();
        gg1 = (Graphics2D) l.getGraphics();
        gg1.setColor(new Color(255, 0, 0));
        gg1.fillPolygon(pp1);
        gg2 = (Graphics2D) l.getGraphics();
        gg2.setColor(new Color(0, 255, 0));
        gg2.fillPolygon(pp2);
        gg3 = (Graphics2D) l.getGraphics();
        gg3.setColor(new Color(0, 0, 255));
        gg3.fillPolygon(pp3);
        boolean contain1, contain2, contain3;
        // System.out.println("lol");
        int mx = me.getX();
        int my = me.getY();
        contain1 = pp1.contains(mx, my);
        contain2 = pp2.contains(mx, my);
        contain3 = pp3.contains(mx, my);
        if (contain1 || contain2 || contain3) {
            startHovercurrent = true;
            if (startHovercurrent != startHoverprev) {
                if (contain1 == true) {
                    Color color = new Color(255, 0, 0);
                    rgb = color.getRGB();
                    tf1.setBackground(new Color(255, 0, 0));
                } else if (contain2 == true) {
                    Color color = new Color(0, 255, 0);
                    rgb = color.getRGB();
                    tf1.setBackground(new Color(0, 255, 0));
                } else if (contain3 == true) {
                    Color color = new Color(0, 0, 255);
                    rgb = color.getRGB();
                    tf1.setBackground(new Color(0, 0, 255));
                } else
                    tf1.setBackground(new Color(rgb));
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        System.out.print(" " + fea[i] + "\n\n");
                        break;
                    }
                }
            }
            startHoverprev = startHovercurrent;
        } else {
            startHovercurrent = false;
            // if(startHovercurrent!=startHoverprev)
            // System.out.println("Outside region");
            startHoverprev = startHovercurrent;
        }
    }

    public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }
            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            // allowing only three attribute here
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
        }
    }
}
```

OUTPUT Option1 :



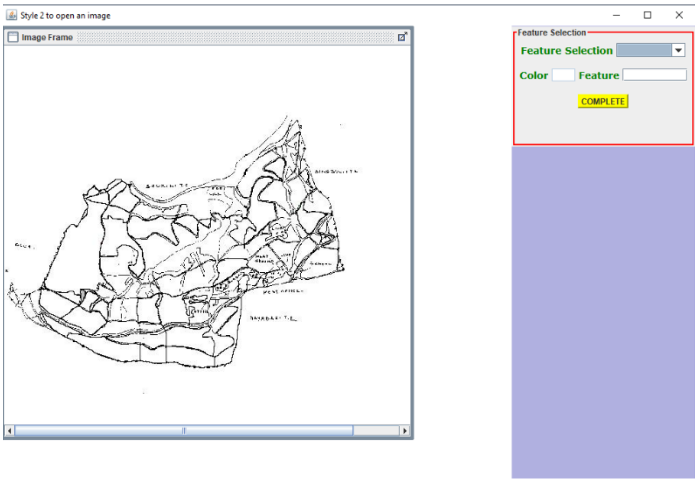
Option 2:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;

class option2 extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
    Graphics2D gg1, gg2, gg3;
    Polygon pp1 = new Polygon();
    Polygon pp2 = new Polygon();
    Polygon pp3 = new Polygon();
    boolean startHovercurrent, startHoverprev = false;

    option2() {
        f = new JFrame("Style 2 to open an image");
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();

        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 600, 600);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        pp1.addPoint(300, 300);
        pp1.addPoint(340, 300);
        pp1.addPoint(340, 260);
        pp1.addPoint(300, 260);
        pp2.addPoint(130, 400);
        pp2.addPoint(180, 420);
        pp2.addPoint(150, 380);
        pp3.addPoint(200, 280);
        pp3.addPoint(230, 280);
        pp3.addPoint(240, 260);
        pp3.addPoint(230, 240);
        pp3.addPoint(210, 235);
        pp3.addPoint(180, 250);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem(" ");
        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);
        /*--------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        tf1.setEditable(false);
        tf1.setBackground(Color.white);
        p1b.add(tf1);
        /*--------------------*/
        String ht3 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature</font> </p>";
        leb3 = new JLabel(ht3);
        p1b.add(leb3);
        tf2 = new JTextField(9);
        tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        p1b.add(tf2);
        p1.add(p1b);
        /*--------------------*
        JPanel p1c=new JPanel();
        fcb=new JButton(" G O ");
        fcb.setBackground(Color.pink);
        fcb.setBorder(new BevelBorder (BevelBorder.RAISED));
        fcb.addActionListener(this);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1c.add(fcb);
        p1.add(p1c);
        /*--------------------*/
        JPanel p1d = new JPanel();
        cb = new JButton(" COMPLETE ");
        cb.setBackground(Color.yellow);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        cb.addActionListener(this);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1d.add(cb);
        p1.add(p1d);
        /*--------------------*/
        c.add(desk, BorderLayout.CENTER);
        c.add(panel, BorderLayout.EAST);
        f.setSize(1024, 738);
        f.setVisible(true);
    }

    public static void main(String args[]) {
        new option2();
    }

    public void mouseClicked(MouseEvent me) {
        boolean contain1, contain2, contain3;
        ColorModel md = pg.getColorModel();
        x = me.getX();
        y = me.getY();
        contain1 = pp1.contains(x, y);
        contain2 = pp2.contains(x, y);
        contain3 = pp3.contains(x, y);
        System.out.print("\nPoint inside polygon:" + (contain1 || contain2 || contain3) +
"\n");
```

```
    int a1 = (y - 1) * ii.getIconWidth() + x;
    System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + "" + pixels[a1]);
    rgb = (md.getRGB(pixels[a1]));
    System.out.print(rgb);
    if (contain1 == true) {
        Color color = new Color(255, 0, 0);
        rgb = color.getRGB();
        tf1.setBackground(new Color(255, 0, 0));
    } else if (contain2 == true) {
        Color color = new Color(0, 255, 0);
        rgb = color.getRGB();
        tf1.setBackground(new Color(0, 255, 0));
    } else if (contain3 == true) {
        Color color = new Color(0, 0, 255);
        rgb = color.getRGB();
        tf1.setBackground(new Color(0, 0, 255));
    } else
        tf1.setBackground(new Color(rgb));
    for (int i = 0; i < count; i++) {
        if (rgb == frgb[i]) {
            System.out.print(" " + fea[i] + "\n\n");
            break;
        }
    }
}

public void mouseEntered(MouseEvent me) {
}

public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}

public void mouseMoved(MouseEvent me) {
    ColorModel cm = pg.getColorModel();
    gg1 = (Graphics2D) l.getGraphics();
    gg1.setColor(new Color(255, 0, 0));
    gg1.fillPolygon(pp1);
    gg2 = (Graphics2D) l.getGraphics();
    gg2.setColor(new Color(0, 255, 0));
    gg2.fillPolygon(pp2);
    gg3 = (Graphics2D) l.getGraphics();
    gg3.setColor(new Color(0, 0, 255));
    gg3.fillPolygon(pp3);
    boolean contain1, contain2, contain3;
    // System.out.println("lol");
    int mx = me.getX();
    int my = me.getY();
    contain1 = pp1.contains(mx, my);
    contain2 = pp2.contains(mx, my);
    contain3 = pp3.contains(mx, my);
    if (contain1 || contain2 || contain3) {
        startHovercurrent = true;
        if (startHovercurrent != startHoverprev) {
            if (contain1 == true) {
                Color color = new Color(255, 0, 0);
                rgb = color.getRGB();
                tf1.setBackground(new Color(255, 0, 0));
            } else if (contain2 == true) {
                Color color = new Color(0, 255, 0);
                rgb = color.getRGB();
                tf1.setBackground(new Color(0, 255, 0));
            } else if (contain3 == true) {
                Color color = new Color(0, 0, 255);
                rgb = color.getRGB();
                tf1.setBackground(new Color(0, 0, 255));
            } else
                tf1.setBackground(new Color(rgb));
            for (int i = 0; i < count; i++) {
                if (rgb == frgb[i]) {
                    System.out.print(" " + fea[i] + "\n\n");
                    break;
                }
            }
        }
        startHoverprev = startHovercurrent;
    } else {
        startHovercurrent = false;
        // if(startHovercurrent!=startHoverprev)
        // System.out.println("Outside region");
        startHoverprev = startHovercurrent;
    }
}

public void mouseDragged(MouseEvent me) {
}
```

```
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }
            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            // allowing only three attribute here
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
        }
    }
}
```

OUTPUT Option2 :



Option 3:

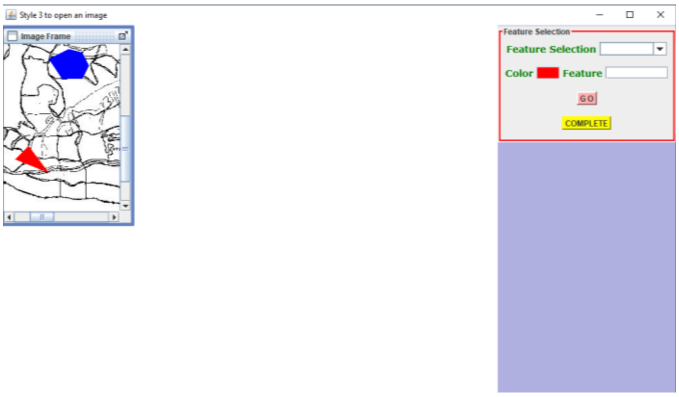
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
```

```
class option3 extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;

    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
    Graphics2D gg1, gg2, gg3;
    Polygon pp1 = new Polygon();
    Polygon pp2 = new Polygon();
    Polygon pp3 = new Polygon();
    boolean startHovercurrent, startHoverprev = false;

    option3() {
        f = new JFrame("Style 3 to open an image");
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 200, 300);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        pp1.addPoint(130, 400);
        pp1.addPoint(180, 420);
        pp1.addPoint(150, 380);
        pp2.addPoint(300, 300);
        pp2.addPoint(340, 300);
        pp2.addPoint(340, 260);
        pp2.addPoint(300, 260);
        pp3.addPoint(200, 280);
        pp3.addPoint(230, 280);
        pp3.addPoint(240, 260);
        pp3.addPoint(230, 240);
        pp3.addPoint(210, 235);
        pp3.addPoint(180, 250);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " +
            "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem(" ");
        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);
        /*--------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " +
            "size=\"4\" face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        tf1.setEditable(false);
        tf1.setBackground(Color.white);
        p1b.add(tf1);
        /*--------------------*/
        String ht3 = "<html><p><font color=\"GREEN\" " +
            "size=\"4\" face=\"Verdana\">Feature</font> </p>";
        leb3 = new JLabel(ht3);
        p1b.add(leb3);
        tf2 = new JTextField(9);
        tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        p1b.add(tf2);
        p1.add(p1b);
        /*--------------------*/
        JPanel p1c = new JPanel();
        fcb = new JButton(" G O ");
        fcb.setBackground(Color.pink);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        fcb.addActionListener(this);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1c.add(fcb);
        p1.add(p1c);
        /*--------------------*/
        JPanel p1d = new JPanel();
        cb = new JButton(" COMPLETE ");
        cb.setBackground(Color.yellow);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        cb.addActionListener(this);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1d.add(cb);
        p1.add(p1d);
        /*--------------------*/
        c.add(desk, BorderLayout.CENTER);
        c.add(panel, BorderLayout.EAST);
        f.setSize(1024, 738);
        f.setVisible(true);
    }
```

```
    public static void main(String args[]) {
        new option3();
    }

    public void mouseClicked(MouseEvent me) {
        boolean contain1, contain2, contain3;
        ColorModel md = pg.getColorModel();
        x = me.getX();
        y = me.getY();
        contain1 = pp1.contains(x, y);
        contain2 = pp2.contains(x, y);
        contain3 = pp3.contains(x, y);
        System.out.print("\nPoint inside polygon:" + (contain1 || contain2 || contain3) +
"\n");
        int a1 = (y - 1) * ii.getIconWidth() + x;
        System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + "" + pixels[a1]);
        rgb = (md.getRGB(pixels[a1]));
        System.out.print(rgb);
        if (contain1 == true) {
            Color color = new Color(255, 0, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(255, 0, 0));
        } else if (contain2 == true) {
            Color color = new Color(0, 255, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 255, 0));
        } else if (contain3 == true) {
            Color color = new Color(0, 0, 255);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 0, 255));
        } else
            tf1.setBackground(new Color(rgb));
        for (int i = 0; i < count; i++) {
            if (rgb == frgb[i]) {
                System.out.print(" " + fea[i] + "\n\n");
                break;
            }
        }
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }
```

```java
public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}

public void mouseMoved(MouseEvent me) {
    ColorModel cm = pg.getColorModel();
    gg1 = (Graphics2D) l.getGraphics();
    gg1.setColor(new Color(255, 0, 0));
    gg1.fillPolygon(pp1);
    gg2 = (Graphics2D) l.getGraphics();
    gg2.setColor(new Color(0, 255, 0));
    gg2.fillPolygon(pp2);
    gg3 = (Graphics2D) l.getGraphics();
    gg3.setColor(new Color(0, 0, 255));
    gg3.fillPolygon(pp3);
    boolean contain1, contain2, contain3;
    // System.out.println("lol");
    int mx = me.getX();
    int my = me.getY();
    contain1 = pp1.contains(mx, my);
    contain2 = pp2.contains(mx, my);
    contain3 = pp3.contains(mx, my);
    if (contain1 || contain2 || contain3) {
        startHovercurrent = true;
        if (startHovercurrent != startHoverprev) {
            if (contain1 == true) {
                Color color = new Color(255, 0, 0);
                rgb = color.getRGB();
                tf1.setBackground(new Color(255, 0, 0));
            } else if (contain2 == true) {
                Color color = new Color(0, 255, 0);
                rgb = color.getRGB();
                tf1.setBackground(new Color(0, 255, 0));
            } else if (contain3 == true) {
                Color color = new Color(0, 0, 255);
                rgb = color.getRGB();
                tf1.setBackground(new Color(0, 0, 255));
            } else
                tf1.setBackground(new Color(rgb));
            for (int i = 0; i < count; i++) {
                if (rgb == frgb[i]) {
                    System.out.print(" " + fea[i] + "\n\n");
                    break;
                }
                }
            }
            startHoverprev = startHovercurrent;
        } else {
            startHovercurrent = false;
            // if(startHovercurrent!=startHoverprev)
            // System.out.println("Outside region");
            startHoverprev = startHovercurrent;
        }
    }

    public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }

            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            // allowing only three attribute here
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
        }
    }
}
```

OUTPUT Option3:



5. Write a program to cascade multiple designed pages(at least three) with resizing tool, take some tools to input and output data.

PROGRAM:

Page 1:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;

class page1 extends JFrame implements ActionListener,ItemListener
{
  JFrame f;
  JPanel panel,p1,p2,p3,p31,p32;
  JLabel leb1,leb2,leb3;
  JTextField tf1;
  JComboBox cb;
  JButton b;
  Container c;
  Color c1=new Color(160,160,220,200);
  Color c2=new Color(160,100,10,100);
  Color c3=new Color(20,160,10,100);
  String ss="Select";

  page1()
  {
        f=new JFrame("Page Design 1");
        f.setSize(500,300);
        f.setLocation(200,150);
        panel=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p1=new JPanel();
        p1.setBackground(c1);
        panel.add(p1);
        String ht1 =
       "<html><p><font color=\"GREEN\" "+
       "size=\"6\" face=\"Verdana\">Enter Your Name</font> </p>";
                leb1=new JLabel(ht1);
                p1.add(leb1);
        tf1=new JTextField(15);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.PLAIN,14));
        p1.add(tf1);
        p2=new JPanel();
        p2.setBackground(c2);
        panel.add(p2);
        String ht2 =
      "<html><p><font color=\"BLUE\" "+
      "size=\"6\" face=\"Verdana\">Choose Your Course</font> </p>";
              leb2=new JLabel(ht2);
        p2.add(leb2);
        cb= new JComboBox();
        cb.setEditable(false);
        cb.setMaximumRowCount(3);
        cb.addItem("Select");
        cb.addItem("M.Sc");
        cb.addItem("MCA");
        cb.addItem("M. Tech");
        cb.addItemListener(this);
        p2.add(cb);

        p3=new JPanel();
        panel.add(p3);
        p3.setLayout(new GridLayout(2,1));
        p31=new JPanel();
        p31.setBackground(c3);
        p3.add(p31);
        String ht3 =
      "<html><p><font color=\"RED\" "+
      "size=\"6\" face=\"Verdana\">Press the Button for Page 2</font> </p>";
              leb3=new JLabel(ht3);
        p31.add(leb3);
        p32=new JPanel();
        p32.setBackground(c3);
        p3.add(p32);
        b=new JButton(" BUTTON ");
        b.setBackground(Color.pink);
        b.setBorder(new BevelBorder (BevelBorder.RAISED));
        b.addActionListener(this);
        p32.add(b);

        c=f.getContentPane();
        c.add(panel);
        f.setVisible(true);
}
public static void main(String aa[])
{
        new page1();
```

```
 }
 public void actionPerformed(ActionEvent ae)
 {
        if(ae.getSource()==b)
        {
                int flag=0;
                String st=tf1.getText();
                if(st.length()==0)
                {
                JOptionPane.showMessageDialog((Component)null,"Please  Write your
Name",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(ss.equals("Select"))
                {
                        JOptionPane.showMessageDialog((Component)null,"Please
Select The Course",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(flag==0)
                {
                        f.setVisible(false);
                        new page2(st,ss);

                }
        }
 }
 public void itemStateChanged(ItemEvent ie)
 {
        ss =(String)cb.getSelectedItem();
 }

}
```

Page 2:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;

class page2 extends JFrame implements ActionListener,ItemListener
{
```

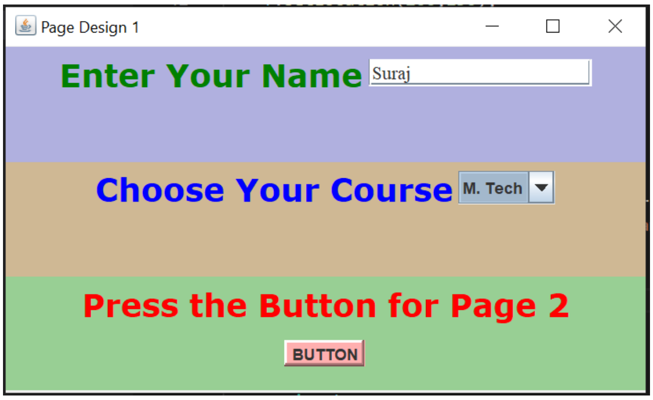
```
JFrame f;
JPanel panel,p1,p2,p3;
JLabel leb1,leb2,leb3;
Container c;
Color c1=new Color(160,160,220,200);
Color c2=new Color(160,100,10,100);
Color c3=new Color(20,160,10,100);

String nam;
String course;

JFrame f1;
JPanel panel1,p11,p21,p31,p311,p321;
JLabel leb11,leb21,leb31;
JTextField tf11;
JComboBox cb1;
JButton b1;
Container co;
Color c11=new Color(160,160,220,200);
Color c21=new Color(160,100,10,100);
Color c31=new Color(20,160,10,100);
String ss1="Select";
page2(String aa, String bb)
{
        nam=aa;
        course=bb;
        f=new JFrame("Page Design 2");
        f.setSize(500,200);
        f.setLocation(200,150);
        panel=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p1=new JPanel();
        p1.setBackground(c1);
        panel.add(p1);
        String ht1 =
      "<html><p><font color=\"GREEN\" "+
      "size=\"6\" face=\"Verdana\">My Name is "+nam+"</font> </p>";
                leb1=new JLabel(ht1);
                p1.add(leb1);

        p2=new JPanel();
        p2.setBackground(c2);
        panel.add(p2);
        String ht2 =
      "<html><p><font color=\"BLUE\" "+
      "size=\"6\" face=\"Verdana\">I am a Student of "+course+"</font> </p>";
                leb2=new JLabel(ht2);
        p2.add(leb2);
        f.setVisible(true);

        f1=new JFrame("For Page Design ");
        f1.setSize(500,300);
        f1.setLocation(200,150);
        panel1=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p11=new JPanel();
        p11.setBackground(c11);
        panel.add(p11);
        String ht11 =
      "<html><p><font color=\"GREEN\" "+
      "size=\"6\" face=\"Verdana\">Enter Country</font> </p>";
                leb11=new JLabel(ht11);
                p11.add(leb11);
        tf11=new JTextField(15);
        tf11.setFont(new Font("TIMES NEW ROMAN", Font.PLAIN,14));
        p11.add(tf11);

        p21=new JPanel();
        p21.setBackground(c21);
        panel.add(p21);
        String ht21 =
      "<html><p><font color=\"YELLOW\" "+
      "size=\"6\" face=\"Verdana\">Choose Gender</font> </p>";
                leb21=new JLabel(ht21);
        p21.add(leb21);
        cb1= new JComboBox();
        cb1.setEditable(false);
        cb1.setMaximumRowCount(3);
        cb1.addItem("Select");
        cb1.addItem("Male");
        cb1.addItem("Female");
        cb1.addItemListener(this);
        p21.add(cb1);

        p31=new JPanel();
        p31.setLayout(new GridLayout(2,1));
        p31.setBackground(c31);
        p21.add(p31);
        String ht31 =
      "<html><p><font color=\"RED\" "+
      "size=\"6\" face=\"Verdana\">Press the Button for Page 3</font> </p>";
                leb31=new JLabel(ht31);
        p31.add(leb31);
        b1=new JButton(" BUTTON ");
        b1.setBackground(Color.pink);
        b1.setBorder(new BevelBorder (BevelBorder.RAISED));
        b1.addActionListener(this);
        p31.add(b1);

        co=f.getContentPane();
        co.add(panel);

        String st1=tf11.getText();
        ss1 =(String)cb1.getSelectedItem();
        new page3(st1,ss1);
        f.setVisible(true);
  }

  public void actionPerformed(ActionEvent ae)
   {
        if(ae.getSource()==b1)
                {
                int flag=0;
                String st1=tf11.getText();
                if(st1.length()==0)
                {
                JOptionPane.showMessageDialog((Component)null,"Please  Write your
Name",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(ss1.equals("Select"))
                {
                       JOptionPane.showMessageDialog((Component)null,"Please
Select The Course",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(flag==0)
                {
                       f1.setVisible(false);
                       new page3(st1,ss1);

                }
        }
  }
```

```
  public void itemStateChanged(ItemEvent ie)
  {
        ss1 =(String)cb1.getSelectedItem();
  }
}
```

Page 3:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;

class page3 extends JFrame
{
  JFrame f;
  JPanel panel,p1,p2,p3;
  JLabel leb1,leb2,leb3;
  Container c;
  Color c1=new Color(160,160,220,200);
  Color c2=new Color(160,100,10,100);
  Color c3=new Color(20,160,10,100);
  String nam;
  String course;
  page3(String aa, String bb)

  {
        nam=aa;
        course=bb;
        f=new JFrame("Page Design 2");
        f.setSize(500,200);
        f.setLocation(200,150);
        panel=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p1=new JPanel();
        p1.setBackground(c1);
        panel.add(p1);
        String ht1 =
       "<html><p><font color=\"GREEN\" "+
       "size=\"6\" face=\"Verdana\">Country is "+nam+"</font> </p>";
                leb1=new JLabel(ht1);
                p1.add(leb1);

        p2=new JPanel();
        p2.setBackground(c2);
        panel.add(p2);
        String ht2 =
       "<html><p><font color=\"BLUE\" "+
       "size=\"6\" face=\"Verdana\">Gender "+course+"</font> </p>";
                leb2=new JLabel(ht2);
        p2.add(leb2);

        p3=new JPanel();
        panel.add(p3);
        p3.setBackground(c3);
        String ht3 =
       "<html><p><font color=\"RED\" "+
       "size=\"10\" face=\"Verdana\"> Thank  You </font> </p>";
        leb3=new JLabel(ht3);
        p3.add(leb3);
        c=f.getContentPane();
        c.add(panel);
        f.setVisible(true);
   }
 }
```

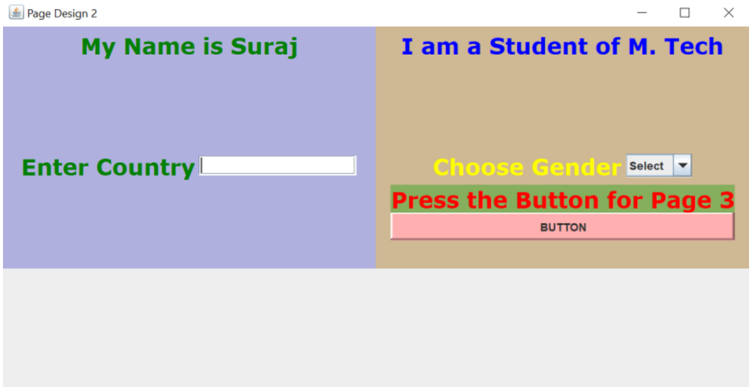
OUTPUT:



Enter Your Name Suraj

Choose Your Course M. Tech

Press the Button for Page 2



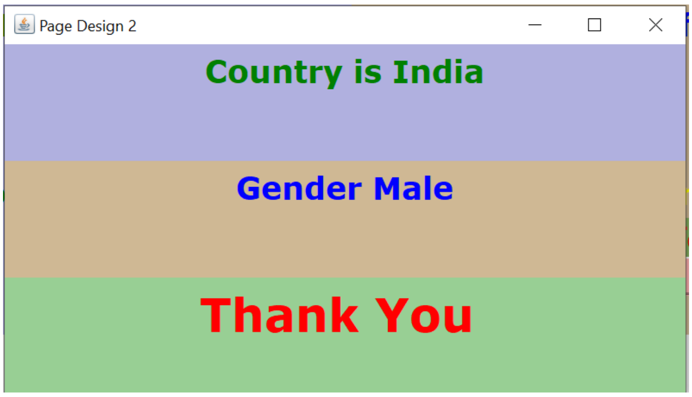
My Name is Suraj

I am a Student of M. Tech

Enter Country

Choose Gender Select

Press the Button for Page 3



Country is India

Gender Male

Thank You

6. Write a program to assign three features in RS-GIS image. Compute how many segment has features of a type.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import java.util.Arrays;
import java.lang.Math;

class Program extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3, leb4;
    JComboBox cb1a;
    JTextField tf1, tf2, tf4;
    JButton fcb, cb, fs;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb = 0;
    int frgb[] = new int[3];
    String fea[] = new String[3];
    int count = 0;
    String ss;
    int ccb = 0;
    int[][] imageArray = new int[1000][1000];
    Program() {
        f = new JFrame("Data Association Page");
        ii = new ImageIcon("session1.gif");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[(ii.getIconWidth() + 1) * (ii.getIconHeight() + 1)];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 1,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 740, 600);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(6, 1));
        panel.add(p1);
        /*---------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem(" ");
        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);
        /*---------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        tf1.setEditable(false);
        tf1.setBackground(Color.white);
        p1b.add(tf1);
        /*--------------------*/
        String ht3 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature</font> </p>";
        leb3 = new JLabel(ht3);
        p1b.add(leb3);
        tf2 = new JTextField(9);
        tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        p1b.add(tf2);
        p1.add(p1b);
        /*--------------------*/
        JPanel p1c = new JPanel();
        fcb = new JButton(" G O ");
        fcb.setBackground(Color.pink);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        fcb.addActionListener(this);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1c.add(fcb);
        p1.add(p1c);
        /*--------------------*/
        JPanel p1d = new JPanel();
        cb = new JButton(" COMPLETE ");
        cb.setBackground(Color.yellow);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        cb.addActionListener(this);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1d.add(cb);
        p1.add(p1d);
        /*--------------------*/
        /*--------------------*/
        JPanel p1e = new JPanel();
        fs = new JButton("Find No of segment");
        fs.setBackground(Color.yellow);
        fs.setBorder(new BevelBorder(BevelBorder.RAISED));
        fs.addActionListener(this);
        fs.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1e.add(fs);
        tf4 = new JTextField(15);
        tf4.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
```

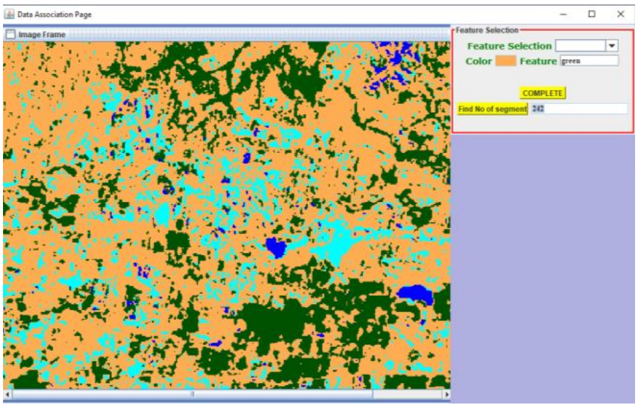
```
    tf4.setEditable(false);
    tf4.setText("Select feature first");
    tf4.setBackground(Color.white);
    p1e.add(tf4);
    tf4.setVisible(false);
    fs.setVisible(false);
    p1.add(p1e);
    c.add(desk, BorderLayout.CENTER);
    c.add(panel, BorderLayout.EAST);
    f.setSize(1024, 738);
    f.setVisible(true);
}

public static void main(String args[]) {
    new Program();
}

public void mouseClicked(MouseEvent me) {
    ColorModel md = pg.getColorModel();
    x = me.getX();
    y = me.getY();
    int a1 = (y - 1) * ii.getIconWidth() + x;
    System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + pixels[a1]);
    // System.out.println(a1);
    rgb = (md.getRGB(pixels[a1]));
    System.out.print(rgb);
    tf1.setBackground(new Color(rgb));
    for (int i = 0; i < count; i++) {
        if (rgb == frgb[i]) {
            System.out.print(" " + fea[i] + "\n\n");
            break;
        }
    }
}

public void mouseEntered(MouseEvent me) {
}

public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}
```

```
    public void mouseMoved(MouseEvent me) {
    }

    public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        ColorModel md = pg.getColorModel();
        int rgb1;
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count >= 3) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }

            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 3 && ccb == 0) {
                for (int i = 0; i < 3; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.print("\n" + frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
            tf4.setVisible(true);
            fs.setVisible(true);
            rgb = 0;
        }
        if (ae.getSource() == fs) {
            int a1, i, j;
            int count1 = 0;
            int current_label = 1;
            int k, l, row, col;
            int front = -1, rear = -1;
            int[] visited = new int[(width * height) + 1];
            int[] labelled = new int[(width * height) + 1];
            int[] queue = new int[(width * height) + 1];
            double temp, temp1;
            tf4.setText("loading...");
            if (rgb == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Click on
Feature to Find Number of Segments",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
```

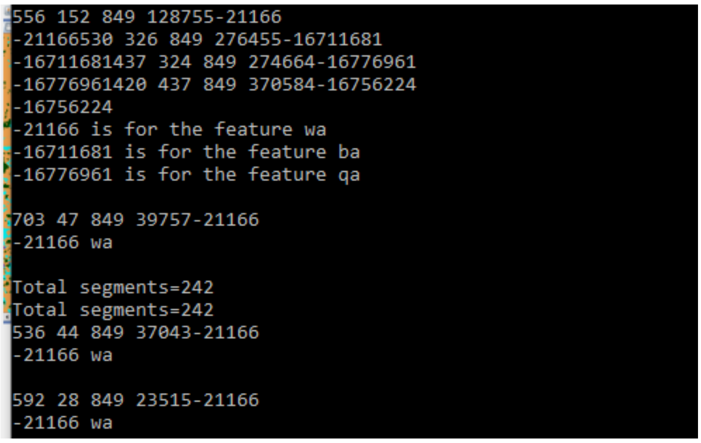
```
} else {
    for (i = 1; i <= height; i++) {
        for (j = 1; j <= width; j++) {
            a1 = (i - 1) * width + j;
            rgb1 = (md.getRGB(pixels[a1]));
            if (rgb1 == rgb) {
                imageArray[i][j] = 1;
            } else
                imageArray[i][j] = 0;
        }
    }
    for (i = 1; i <= height; i++) {
        for (j = 1; j <= width; j++) {
            if (imageArray[i][j] == 1 && visited[(i - 1) * width + j] == 0
                    && labelled[(i - 1) * width + j] == 0) {
                rear++;
                front++;
                queue[rear] = (i - 1) * width + j;
                visited[(i - 1) * width + j] = 1;
                current_label++;
                labelled[(i - 1) * width + j] = current_label;
                // now insert adjacent pixel one by one
                while (front <= rear) {
                    temp = queue[front];
                    temp1 = width;
                    row = (int) Math.ceil(temp / temp1);
                    col = queue[front] % width;
                    if (col == 0)
                        col = width;
                    for (k = -1; k <= 1; k++) {
                        for (l = -1; l <= 1; l++) {
                            if (imageArray[row + k][col + l] == 1
                                    && visited[(row + k - 1) * width + col + l] == 0
                                    && labelled[(row + k - 1) * width + col + l] == 0) {
                                rear++;
                                queue[rear] = (row + k - 1) * width + col + l;
                                visited[(row + k - 1) * width + col + l] = 1;
                                labelled[(row + k - 1) * width + col + l] = current_label;
                            }
                        }
                    }
                    front++;
                    visited[(row - 1) * width + col] = 1;
                    labelled[(row - 1) * width + col] = current_label;
                }
            }
```

```
                        // initialize queue
                        for (k = 0; k < height * width + 1; k++) {
                            queue[k] = 0;
                            front = -1;
                            rear = -1;
                        }
                    }
                }
                System.out.println("Total segments=" + (current_label - 1));
                tf4.setText(Integer.toString(current_label - 1));
            }
        }
    }
}
```

OUTPUT:



Feature Selection
Color Feature green
Find No of segment 242



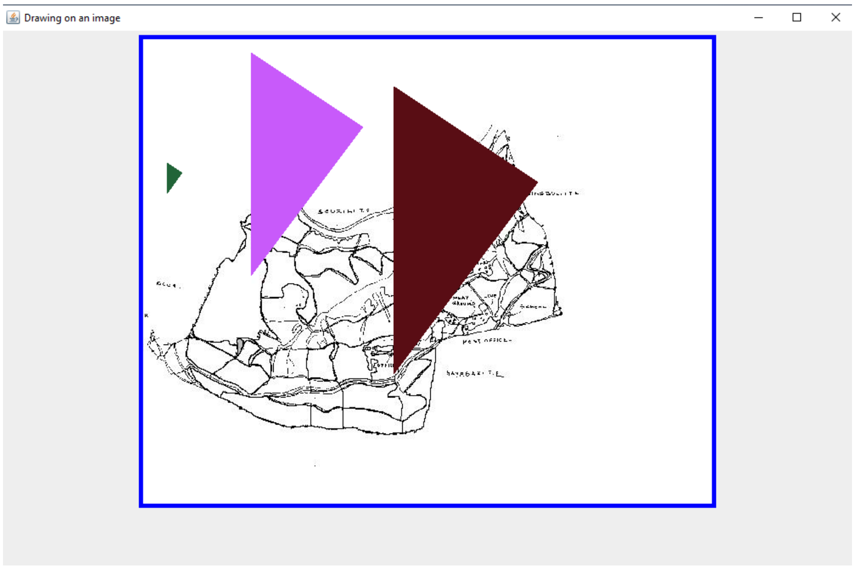
```
556 152 849 128755-21166
-21166530 326 849 276455-16711681
-16711681437 324 849 274664-16776961
-16776961420 437 849 370584-16756224
-16756224
-21166 is for the feature wa
-16711681 is for the feature ba
-16776961 is for the feature qa

703 47 849 39757-21166
-21166 wa

Total segments=242
Total segments=242
536 44 849 37043-21166
-21166 wa

592 28 849 23515-21166
-21166 wa
```

7. Write a GIS based program to draw a polygon(four different places with 5 cm,7cm,9cm width) into a GIS image and designate the mark with some attribute.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    public static double getDPI(){
        double dpi = Toolkit.getDefaultToolkit().getScreenResolution();
        // System.out.println("Screen DPI: " + dpi);
        return dpi;
    }

    public static double cmToPixel(double cm){
        double dpi = getDPI();
        double pixel = cm * dpi / 2.54;
        // System.out.println("cm: " + cm + " pixel: " + pixel);
        return pixel;
    }

    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);
    }

    public void mouseClicked(MouseEvent me) {
        gg = (Graphics2D) l.getGraphics();

        // 5cm 7cm 9cm polygon
        Polygon p1 = drawPolygon(cmToPixel(5), 25, 50);
        gg.setColor(new Color(200, 13, 40));
        gg.fillPolygon(p1);

        Polygon p2 = drawPolygon(cmToPixel(7), 130, 20);
        gg.setColor(new Color(100, 13, 40));
        gg.fillPolygon(p2);

        Polygon p3 = drawPolygon(cmToPixel(9), 300, 20);
        gg.setColor(new Color(0, 13, 40));
        gg.fillPolygon(p3);
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }
    public void mousePressed(MouseEvent me) {
    }

    public void mouseReleased(MouseEvent me) {
    }

    public Polygon drawPolygon (double width, double x, double y) {
        Polygon p = new Polygon();
        p.addPoint((int)x, (int)y);
        p.addPoint((int)(x), (int)(y+ width));
        p.addPoint((int)(x + width/2), (int)(y + width/2));
        return p;
    }

    public void mouseMoved(MouseEvent me) {

    }

    public void mouseDragged(MouseEvent me) {
    }

    public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:

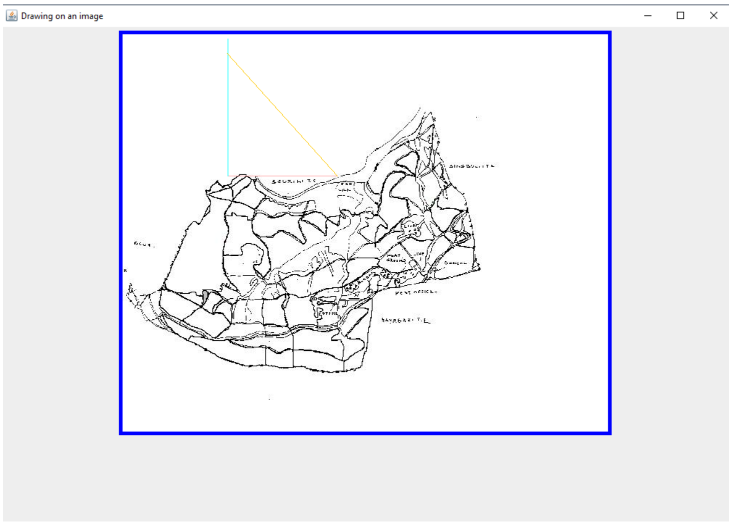


8. Write a program to draw line(4cm,5cm,6cm) in three different places and point into a GIS image so that they can form a triangle.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    public static double getDPI(){
        double dpi = Toolkit.getDefaultToolkit().getScreenResolution();
        // System.out.println("Screen DPI: " + dpi);
        return dpi;
    }

    public static double cmToPixel(double cm){
        double dpi = getDPI();
        double pixel = cm * dpi / 2.54;
        // System.out.println("cm: " + cm + " pixel: " + pixel);
        return pixel;
    }
    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);

    }

    public static double[] drawLine(double length, double angle, int x, int y,
Graphics2D gg, Color color) {
        double radians = Math.toRadians(angle);
        double x1 = x + length * Math.cos(radians);
        double y1 = y - length * Math.sin(radians);
        gg.setColor(color);
        gg.drawLine((int) x, (int) y, (int) x1, (int) y1);
        double[] end = new double[2];
        end[0] = x1;
        end[1] = y1;
        return end;
    }

    public void mouseClicked(MouseEvent me) {
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }
    public void mouseReleased(MouseEvent me) {
    }

    public void mouseMoved(MouseEvent me) {

        ColorModel cm = pg.getColorModel();
        gg = (Graphics2D) l.getGraphics();

        // Drawing line 2cm, 6cm, 9cm long

        int startX = 150;
        int startY = 200;
        int angleA = 0;
        int angleB = 90;
        int angleC = 132;

        double[] endA = drawLine(cmToPixel(4), angleA, startX, startY, gg, Color.PINK);
        double[] endB = drawLine(cmToPixel(5), angleB, startX, startY, gg, Color.CYAN);
        drawLine(cmToPixel(6), angleC, (int)endA[0], (int)endA[1], gg, Color.ORANGE);
    }

    public void mouseDragged(MouseEvent me) {
    }

    public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:



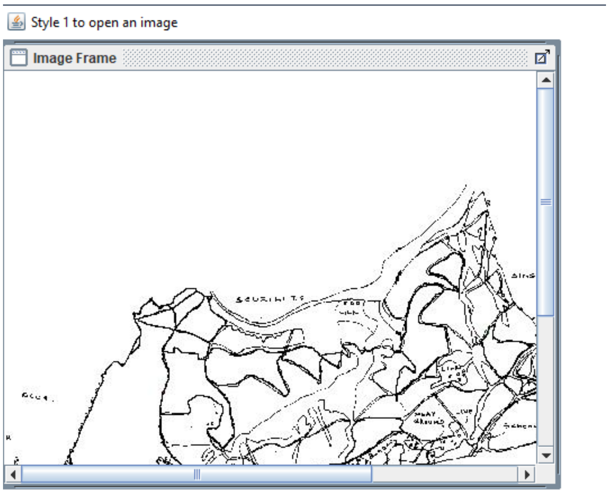
9. Write a program to show various process to open GIS image into desktop(at least three) with resizing tool.

PROGRAM:

Option 1:

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

class option1 extends JFrame {
    JFrame f;
    JLabel l;
    JPanel panel;
    ImageIcon ii;
    Image img;
    Container c;

    option1() {
        f = new JFrame("Style 1 to open an image");
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", true, false,

true, false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 500, 400);
        desk.add(p);
        p.setVisible(true);
        c.add(desk, BorderLayout.CENTER);
        f.setSize(1024, 738);
        f.setVisible(true);
    }

    public static void main(String aa[]) {
        new option1();
    }
}
```

OUTPUT of Option1:

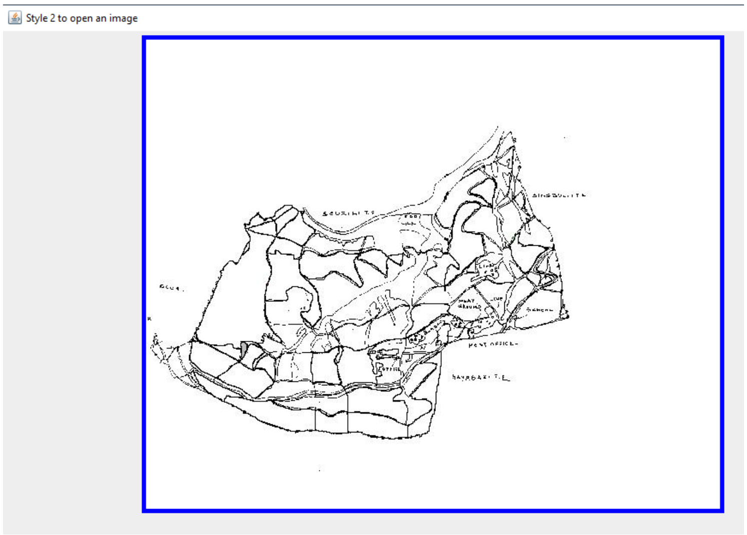


Option 2:

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

class option2 extends JFrame {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;

    option2() {
        f = new JFrame("Style 2 to open an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        l = new JLabel(ii, JLabel.CENTER);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));

        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);
    }
    public static void main(String aa[]) {
```

```
new option2();
    }
}
```

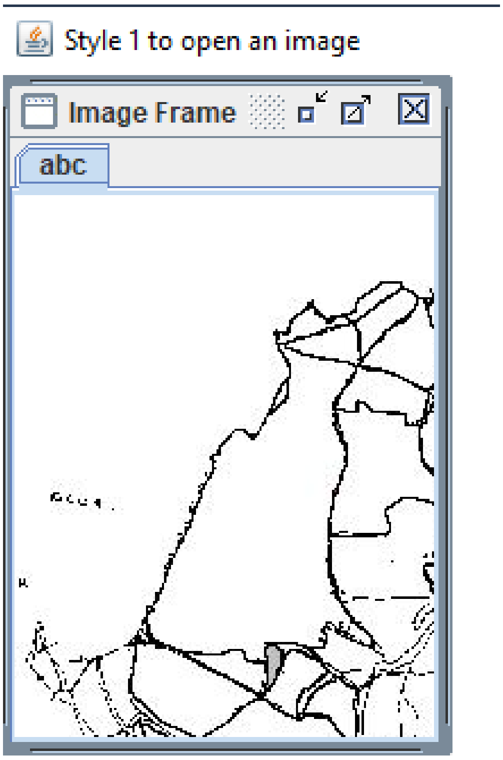
OUTPUT of Option2:



Option 3:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.image.*;
import javax.swing.JFrame;
class option3 extends JFrame {
    JFrame f;
    JLabel l;
    JPanel panel;
    ImageIcon ii;
    Image img;
    Container c;
    option3() {
        f = new JFrame("Style 1 to open an image");
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        l = new JLabel(ii, JLabel.LEFT);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", true, true, true, true);
        JTabbedPane jtp = new JTabbedPane();
        jtp.addTab("abc", l);
        p.setContentPane(jtp);
        p.setBounds(0, 0, 200, 300);
        desk.add(p);
        p.setVisible(true);
        c.add(desk, BorderLayout.CENTER);
        f.setSize(1024, 738);
        f.setVisible(true);
    }
    public static void main(String aa[]) {
        new option3();
    }
}
```

OUTPUT of Option3:



10. Write a program to cascade multiple designed pages(at least three) with resizing tool.

PROGRAM:

Page 1:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;

class page1 extends JFrame implements ActionListener,ItemListener
{
  JFrame f;
  JPanel panel,p1,p2,p3,p31,p32;
  JLabel leb1,leb2,leb3;
  JTextField tf1;
  JComboBox cb;
  JButton b;
  Container c;
  Color c1=new Color(160,160,220,200);
  Color c2=new Color(160,100,10,100);
  Color c3=new Color(20,160,10,100);
  String ss="Select";

  page1()
  {
        f=new JFrame("Page Design 1");
        f.setSize(500,300);
        f.setLocation(200,150);
        panel=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p1=new JPanel();
        p1.setBackground(c1);
        panel.add(p1);
        String ht1 =
       "<html><p><font color=\"GREEN\" "+
       "size=\"6\" face=\"Verdana\">Enter Your Name</font> </p>";
              leb1=new JLabel(ht1);
              p1.add(leb1);
        tf1=new JTextField(15);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.PLAIN,14));
        p1.add(tf1);
        p2=new JPanel();
        p2.setBackground(c2);
        panel.add(p2);
        String ht2 =
      "<html><p><font color=\"BLUE\" "+
      "size=\"6\" face=\"Verdana\">Choose Your Course</font> </p>";
                leb2=new JLabel(ht2);
        p2.add(leb2);
        cb= new JComboBox();
        cb.setEditable(false);
        cb.setMaximumRowCount(3);
        cb.addItem("Select");
        cb.addItem("M.Sc");
        cb.addItem("MCA");
        cb.addItem("M. Tech");
        cb.addItemListener(this);
        p2.add(cb);

        p3=new JPanel();
        panel.add(p3);
        p3.setLayout(new GridLayout(2,1));
        p31=new JPanel();
        p31.setBackground(c3);
        p3.add(p31);
        String ht3 =
      "<html><p><font color=\"RED\" "+
      "size=\"6\" face=\"Verdana\">Press the Button for Page 2</font> </p>";
                leb3=new JLabel(ht3);
        p31.add(leb3);
        p32=new JPanel();
        p32.setBackground(c3);
        p3.add(p32);
        b=new JButton(" BUTTON ");
        b.setBackground(Color.pink);
        b.setBorder(new BevelBorder (BevelBorder.RAISED));
        b.addActionListener(this);
        p32.add(b);

        c=f.getContentPane();
        c.add(panel);
        f.setVisible(true);
}
public static void main(String aa[])
{
        new page1();
```

```
 }
 public void actionPerformed(ActionEvent ae)
 {
        if(ae.getSource()==b)
        {
                int flag=0;
                String st=tf1.getText();
                if(st.length()==0)
                {
                JOptionPane.showMessageDialog((Component)null,"Please  Write your
Name",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(ss.equals("Select"))
                {
                        JOptionPane.showMessageDialog((Component)null,"Please
Select The Course",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(flag==0)
                {
                        f.setVisible(false);
                        new page2(st,ss);

                }
        }
 }
 public void itemStateChanged(ItemEvent ie)
 {
        ss =(String)cb.getSelectedItem();
 }

}
```

Page 2:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;

class page2 extends JFrame implements ActionListener,ItemListener
```

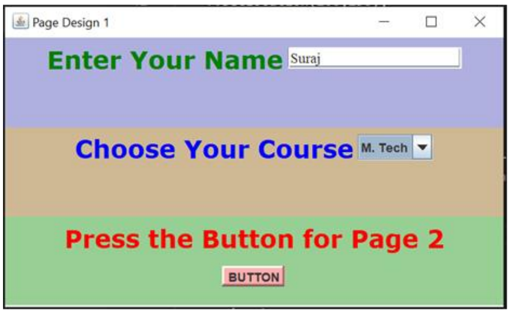
```
{
  JFrame f;
  JPanel panel,p1,p2,p3;
  JLabel leb1,leb2,leb3;
  Container c;
  Color c1=new Color(160,160,220,200);
  Color c2=new Color(160,100,10,100);
  Color c3=new Color(20,160,10,100);

  String nam;
  String course;

  JFrame f1;
  JPanel panel1,p11,p21,p31,p311,p321;
  JLabel leb11,leb21,leb31;
  JTextField tf11;
  JComboBox cb1;
  JButton b1;
  Container co;
  Color c11=new Color(160,160,220,200);
  Color c21=new Color(160,100,10,100);
  Color c31=new Color(20,160,10,100);
  String ss1="Select";
  page2(String aa, String bb)
  {
        nam=aa;
        course=bb;
        f=new JFrame("Page Design 2");

        f.setSize(500,200);
        f.setLocation(200,150);
        panel=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p1=new JPanel();
        p1.setBackground(c1);
        panel.add(p1);
        String ht1 =
       "<html><p><font color=\"GREEN\" "+
       "size=\"6\" face=\"Verdana\">My Name is "+nam+"</font> </p>";
              leb1=new JLabel(ht1);
              p1.add(leb1);

        p2=new JPanel();
        p2.setBackground(c2);
        panel.add(p2);
        String ht2 =
       "<html><p><font color=\"BLUE\" "+
       "size=\"6\" face=\"Verdana\">I am a Student of "+course+"</font> </p>";
              leb2=new JLabel(ht2);
        p2.add(leb2);
        f.setVisible(true);
        f1=new JFrame("For Page Design ");
        f1.setSize(500,300);
        f1.setLocation(200,150);
        panel1=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p11=new JPanel();
        p11.setBackground(c11);
        panel.add(p11);
        String ht11 =
       "<html><p><font color=\"GREEN\" "+
       "size=\"6\" face=\"Verdana\">Enter Country</font> </p>";
              leb11=new JLabel(ht11);
              p11.add(leb11);
        tf11=new JTextField(15);
        tf11.setFont(new Font("TIMES NEW ROMAN", Font.PLAIN,14));
        p11.add(tf11);

        p21=new JPanel();
        p21.setBackground(c21);
        panel.add(p21);
        String ht21 =
       "<html><p><font color=\"YELLOW\" "+
       "size=\"6\" face=\"Verdana\">Choose Gender</font> </p>";
              leb21=new JLabel(ht21);
        p21.add(leb21);
        cb1= new JComboBox();
        cb1.setEditable(false);
        cb1.setMaximumRowCount(3);
        cb1.addItem("Select");
        cb1.addItem("Male");
        cb1.addItem("Female");

        cb1.addItemListener(this);
        p21.add(cb1);

        p31=new JPanel();
        p31.setLayout(new GridLayout(2,1));
        p31.setBackground(c31);
        p21.add(p31);
```

```
        String ht31 =
       "<html><p><font color=\"RED\" "+
       "size=\"6\" face=\"Verdana\">Press the Button for Page 3</font> </p>";
                leb31=new JLabel(ht31);
        p31.add(leb31);
        b1=new JButton(" BUTTON ");
        b1.setBackground(Color.pink);
        b1.setBorder(new BevelBorder (BevelBorder.RAISED));
        b1.addActionListener(this);
        p31.add(b1);

        co=f.getContentPane();
        co.add(panel);

        String st1=tf11.getText();
        ss1 =(String)cb1.getSelectedItem();
        new page3(st1,ss1);
        f.setVisible(true);
  }
  public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==b1)
                {
                int flag=0;
                String st1=tf11.getText();
                if(st1.length()==0)
                {
                JOptionPane.showMessageDialog((Component)null,"Please  Write your
Name",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(ss1.equals("Select"))
                {
                        JOptionPane.showMessageDialog((Component)null,"Please
Select The Course",
                "NOTIFICATION",JOptionPane.ERROR_MESSAGE);
                flag=1;
                }
                if(flag==0)
                {
                        f1.setVisible(false);
                        new page3(st1,ss1);

                }
        }
  }

  public void itemStateChanged(ItemEvent ie)
  {
        ss1 =(String)cb1.getSelectedItem();
  }
}
```

Page 3:

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

import javax.swing.border.*;

class page3 extends JFrame
{
  JFrame f;
  JPanel panel,p1,p2,p3;
  JLabel leb1,leb2,leb3;
  Container c;
  Color c1=new Color(160,160,220,200);
  Color c2=new Color(160,100,10,100);
  Color c3=new Color(20,160,10,100);
  String nam;
  String course;
  page3(String aa, String bb)
  {
        nam=aa;
        course=bb;
        f=new JFrame("Page Design 2");
        f.setSize(500,200);
        f.setLocation(200,150);
        panel=new JPanel();
        panel.setLayout(new GridLayout(3,1));

        p1=new JPanel();
        p1.setBackground(c1);
        panel.add(p1);
        String ht1 =
       "<html><p><font color=\"GREEN\" "+
       "size=\"6\" face=\"Verdana\">Country is "+nam+"</font> </p>";
                leb1=new JLabel(ht1);
                p1.add(leb1);
        p2=new JPanel();
        p2.setBackground(c2);
        panel.add(p2);
        String ht2 =
       "<html><p><font color=\"BLUE\" "+
       "size=\"6\" face=\"Verdana\">Gender "+course+"</font> </p>";
              leb2=new JLabel(ht2);
        p2.add(leb2);

        p3=new JPanel();
        panel.add(p3);
        p3.setBackground(c3);
        String ht3 =
       "<html><p><font color=\"RED\" "+
       "size=\"10\" face=\"Verdana\"> Thank  You </font> </p>";
        leb3=new JLabel(ht3);
        p3.add(leb3);


        c=f.getContentPane();
        c.add(panel);
        f.setVisible(true);
    }
 }
```

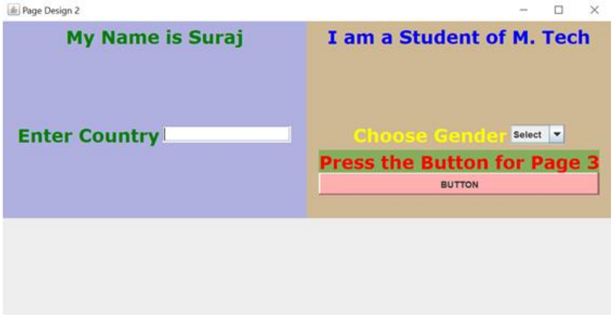
OUTPUT:


Page Design 1

Enter Your Name Suraj

Choose Your Course M. Tech

Press the Button for Page 2


Page Design 2

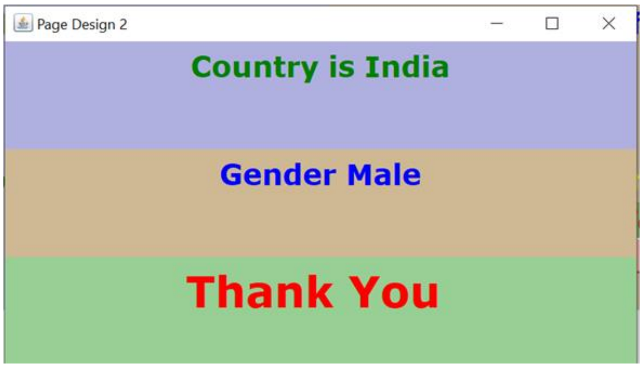
My Name is Suraj

I am a Student of M. Tech

Enter Country

Choose Gender Select

Press the Button for Page 3


Page Design 2

Country is India

Gender Male

Thank You

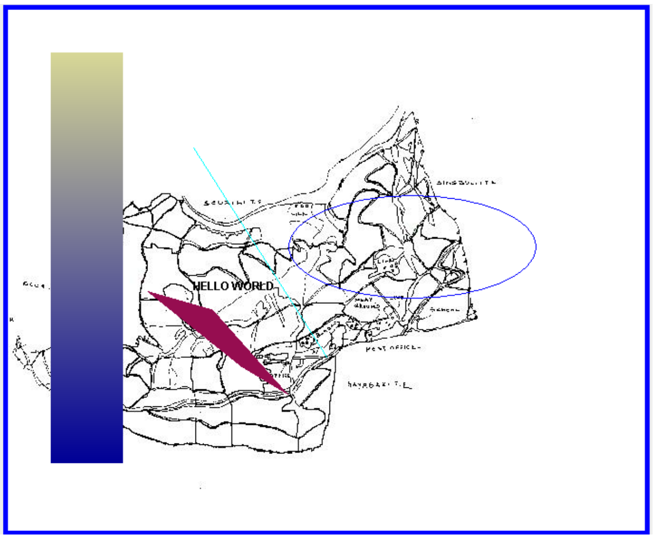
11. Draw Point, Line, Polygon, text, oval and gradient on the image.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);
    c = f.getContentPane();
    c.add(panel);
    f.setVisible(true);
}

public void mouseClicked(MouseEvent me) {
}

public void mouseEntered(MouseEvent me) {
}

public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}
public void mouseMoved(MouseEvent me) {
    ColorModel cm = pg.getColorModel();
    gg = (Graphics2D) l.getGraphics();
    // Drawing line
    // draw oval
    gg.setColor(Color.blue);
    gg.drawOval(300, 200, 260, 108);
    // draw line
    gg.setColor(Color.cyan);
    gg.drawLine(200, 150, 340, 370);
    // draw text
    gg.setColor(Color.black);
    gg.drawString("HELLO WORLD", 200, 300);
    // Draw Polygons
    Polygon pp1 = new Polygon();
    pp1.addPoint(150, 300);
    pp1.addPoint(250, 380);
    pp1.addPoint(300, 410);
    pp1.addPoint(220, 320);
    gg.setColor(new Color(150, 13, 80));
    gg.fillPolygon(pp1);
    // Draw gradient
    int k = 0, l = 0;
    for (int i = 480; i >= 50; i -= 2) {
        gg.setColor(new Color(k, l, 150));
        gg.drawRect(50, i, 75, 1);
        k = k + 1;
```

```
            l = l + 1;
        }
    }

    public void mouseDragged(MouseEvent me) {
    }

    public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:



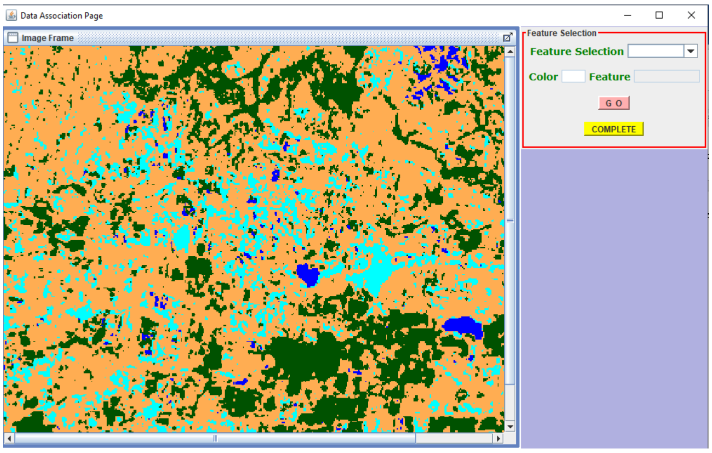
12. Write a program to perform the following operations click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the corresponding text field...session1.gif image has four features they are -

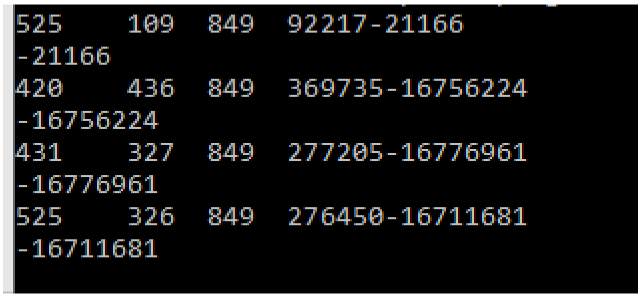
1. Vegetation
2. water body
3. moist land
4. bare soil

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.io.Console;
import java.awt.Graphics.*;

class Program extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3, leb4;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;

    Program() {
        f = new JFrame("Data Association Page");
        ii = new ImageIcon("session1.gif");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();

        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 740, 600);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);

        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);

        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);

        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem("            ");

        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);

        /*--------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        tf1.setEditable(false);
        tf1.setBackground(Color.white);
        p1b.add(tf1);

        /*--------------------*/

        String ht3 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature</font> </p>";
        leb3 = new JLabel(ht3);
        p1b.add(leb3);
        tf2 = new JTextField(9);
        tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        p1b.add(tf2);

        p1.add(p1b);
        /*--------------------*/

        JPanel p1c = new JPanel();
        fcb = new JButton("   G  O   ");
        fcb.setBackground(Color.pink);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        fcb.addActionListener(this);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1c.add(fcb);

        p1.add(p1c);
        /*--------------------*/
        JPanel p1d = new JPanel();
        cb = new JButton("   COMPLETE   ");
        cb.setBackground(Color.yellow);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        cb.addActionListener(this);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1d.add(cb);

        p1.add(p1d);
        /*--------------------*/
        c.add(desk, BorderLayout.CENTER);
        c.add(panel, BorderLayout.EAST);
        f.setSize(1024, 738);
        f.setVisible(true);

    }

    public static void main(String args[]) {
        new Program();
    }

    public void mouseClicked(MouseEvent me) {
        ColorModel md = pg.getColorModel();
        x = me.getX();
        y = me.getY();
        int a1 = (y - 1) * ii.getIconWidth() + x;
        System.out.println(x + "    " + y + "  " + ii.getIconWidth() + "  " + a1 + pixels[a1]);
        // System.out.println(a1);
        rgb = (md.getRGB(pixels[a1]));
        System.out.println(rgb);
        tf1.setBackground(new Color(rgb));
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }

    public void mouseReleased(MouseEvent me) {
    }

    public void mouseMoved(MouseEvent me) {
    }
    public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated before", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }

            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }

                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }
            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            if (count == 4 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                tf1.setBackground(Color.white);
                tf2.setText(null);
                tf2.setEditable(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + "   is for the feature " + fea[i]);
                }
                ccb = 1;
            }
        }
    }
}
```

OUTPUT:





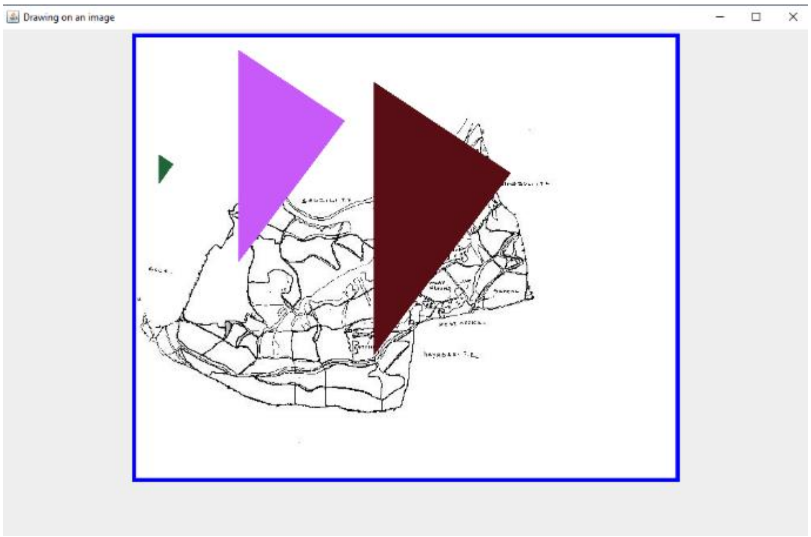
```
525     109   849   92217-21166
-21166
420     436   849   369735-16756224
-16756224
431     327   849   277205-16776961
-16776961
525     326   849   276450-16711681
-16711681
```

13. Write a GIS based program to draw a polygon(four different places with 2 cm,6cm,8cm width) into a GIS image.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    public static double getDPI(){
        double dpi = Toolkit.getDefaultToolkit().getScreenResolution();
        // System.out.println("Screen DPI: " + dpi);
        return dpi;
    }

    public static double cmToPixel(double cm){
        double dpi = getDPI();
        double pixel = cm * dpi / 2.54;
        // System.out.println("cm: " + cm + " pixel: " + pixel);
        return pixel;
    }
    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);
    }

    public void mouseClicked(MouseEvent me) {
        gg = (Graphics2D) l.getGraphics();

        // 1cm 7cm 9cm polygon
        Polygon p1 = drawPolygon(cmToPixel(1), 33, 150);
        gg.setColor(new Color(33, 100, 56));
        gg.fillPolygon(p1);

        Polygon p2 = drawPolygon(cmToPixel(7), 132, 20);
        gg.setColor(new Color(200, 90, 250));
        gg.fillPolygon(p2);

        Polygon p3 = drawPolygon(cmToPixel(9), 300, 60);
        gg.setColor(new Color(89, 13, 20));
        gg.fillPolygon(p3);
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }
    public void mouseReleased(MouseEvent me) {
    }

    public Polygon drawPolygon (double width, double x, double y) {
        Polygon p = new Polygon();
        p.addPoint((int)x, (int)y);
        p.addPoint((int)(x), (int)(y+ width));
        p.addPoint((int)(x + width/2), (int)(y + width/3));
        return p;
    }

    public void mouseMoved(MouseEvent me) {

    }

    public void mouseDragged(MouseEvent me) {
    }

    public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:



14. Write a GIS application program that will do the following :

a. it will open an image in a panel.

b. an option will be there for labeling region in the current image depending on the pixel color.

c. after completion of step 2, the user can ask for percentage of particular region in the current image.

Program should be able to show the percentage value of the region with respect to the whole image.

d. the above stated problem in step 3, should be implemented for an image where same color pixels are scattered.

PROGRAM:

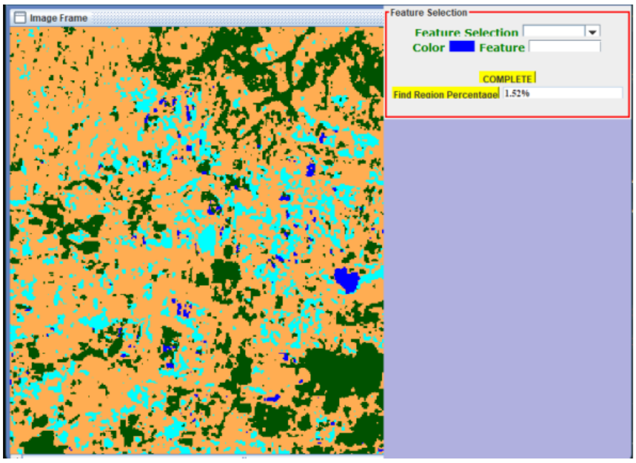
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import java.util.Arrays;
import java.lang.Math;

class Program extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3, leb4;
    JComboBox cb1a;
    JTextField tf1, tf2, tf4;
    JButton fcb, cb, fs;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb = 0;
    int frgb[] = new int[3];
    String fea[] = new String[3];
    int count = 0;
    String ss;
    int ccb = 0;
    int[][] imageArray = new int[1000][1000];

    Program() {
        f = new JFrame("Data Association Page");
        ii = new ImageIcon("session1.gif");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[(ii.getIconWidth() + 1) * (ii.getIconHeight() + 1)];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 1,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 740, 600);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(6, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
```

```
cb1a.setPreferredSize(new Dimension(100, 20));
cb1a.setEditable(false);
cb1a.setBackground(Color.WHITE);
cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
cb1a.addItem(" ");
// cb1a.addActionListener(this);
p1a.add(cb1a);
p1.add(p1a);
/*--------------------*/
JPanel p1b = new JPanel();
String ht2 = "<html><p><font color=\"GREEN\" " +
      "size=\"4\" face=\"Verdana\">Color</font> </p>";
leb2 = new JLabel(ht2);
p1b.add(leb2);
tf1 = new JTextField(3);
tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
tf1.setEditable(false);
tf1.setBackground(Color.white);
p1b.add(tf1);
/*--------------------*/
String ht3 = "<html><p><font color=\"GREEN\" " +
      "size=\"4\" face=\"Verdana\">Feature</font> </p>";
leb3 = new JLabel(ht3);
p1b.add(leb3);
tf2 = new JTextField(9);
tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
p1b.add(tf2);
p1.add(p1b);
/*--------------------*/
JPanel p1c = new JPanel();
fcb = new JButton(" G O ");
fcb.setBackground(Color.pink);
fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
fcb.addActionListener(this);
fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
p1c.add(fcb);
p1.add(p1c);
/*--------------------*/
JPanel p1d = new JPanel();
cb = new JButton(" COMPLETE ");
cb.setBackground(Color.yellow);
cb.setBorder(new BevelBorder(BevelBorder.RAISED));
cb.addActionListener(this);
cb.setBorder(new BevelBorder(BevelBorder.RAISED));
p1d.add(cb);
p1.add(p1d);
```

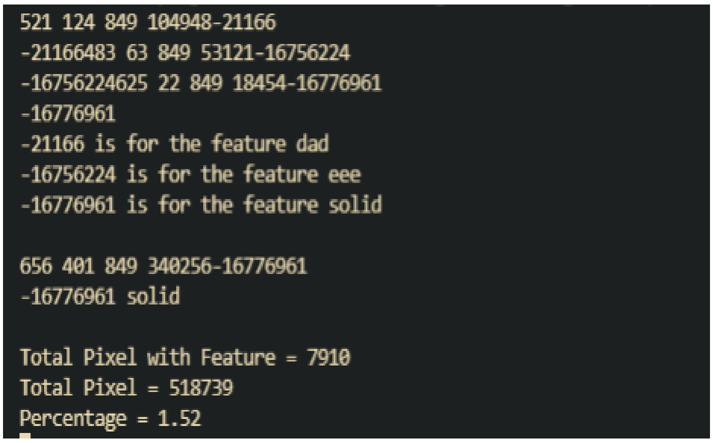
```
    /*--------------------*/
    /*--------------------*/
    JPanel p1e = new JPanel();
    fs = new JButton("Find Region Percentage");
    fs.setBackground(Color.yellow);
    fs.setBorder(new BevelBorder(BevelBorder.RAISED));
    fs.addActionListener(this);
    fs.setBorder(new BevelBorder(BevelBorder.RAISED));
    p1e.add(fs);
    tf4 = new JTextField(15);
    tf4.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
    tf4.setEditable(false);
    tf4.setText("Select feature first");
    tf4.setBackground(Color.white);
    p1e.add(tf4);
    tf4.setVisible(false);
    fs.setVisible(false);
    p1.add(p1e);
    c.add(desk, BorderLayout.CENTER);
    c.add(panel, BorderLayout.EAST);
    f.setSize(1024, 738);
    f.setVisible(true);
}

public static void main(String args[]) {
    new Program();
}

public void mouseClicked(MouseEvent me) {
    ColorModel md = pg.getColorModel();
    x = me.getX();
    y = me.getY();
    int a1 = (y - 1) * ii.getIconWidth() + x;
    System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + pixels[a1]);
    // System.out.println(a1);
    rgb = (md.getRGB(pixels[a1]));
    System.out.print(rgb);
    tf1.setBackground(new Color(rgb));
    for (int i = 0; i < count; i++) {
        if (rgb == frgb[i]) {
            System.out.print(" " + fea[i] + "\n\n");
            break;
        }
    }
}
```

```
    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }

    public void mouseReleased(MouseEvent me) {
    }

    public void mouseMoved(MouseEvent me) {
    }

    public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        ColorModel md = pg.getColorModel();
        int rgb1;
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count >= 3) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }

            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 3 && ccb == 0) {
                for (int i = 0; i < 3; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.print("\n" + frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
            tf4.setVisible(true);
            fs.setVisible(true);
            rgb = 0;
        }
        if (ae.getSource() == fs) {
            int a1, i, j;
            tf4.setText("loading...");
            if (rgb == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Click on
Feature to Find Number of Segments",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
            } else {
                int total_pixel = width * height;
                int pixel_with_feature = 0;

                for (i = 1; i <= height; i++) {
                    for (j = 1; j <= width; j++) {
                        a1 = (i - 1) * width + j;
                        rgb1 = (md.getRGB(pixels[a1]));
                        if (rgb1 == rgb) {
                            pixel_with_feature++;
                            imageArray[i][j] = 1;
                        } else
                            imageArray[i][j] = 0;
                    }
                }
                System.out.println("Total Pixel with Feature = " + pixel_with_feature);
                System.out.println("Total Pixel = " + total_pixel);
                double percentage = (double) pixel_with_feature*100 / total_pixel;
                // to 2 decimal places
                percentage = (double) Math.round(percentage * 100) / 100;
                System.out.println("Percentage = " + percentage);
                tf4.setText(Double.toString(percentage)+"%");
            }
        }
    }
}
```

OUTPUT:



Find Region Percentage 1.52%



```
521 124 849 104948-21166
-21166483 63 849 53121-16756224
-16756224625 22 849 18454-16776961
-16776961
-21166 is for the feature dad
-16756224 is for the feature eee
-16776961 is for the feature solid

656 401 849 340256-16776961
-16776961 solid

Total Pixel with Feature = 7910
Total Pixel = 518739
Percentage = 1.52
```

15. Write a program to show various process to open GIS image into desktop(at least three) with resizing it, mark some portion of the image with some color and point it with attribute.

PROGRAM:

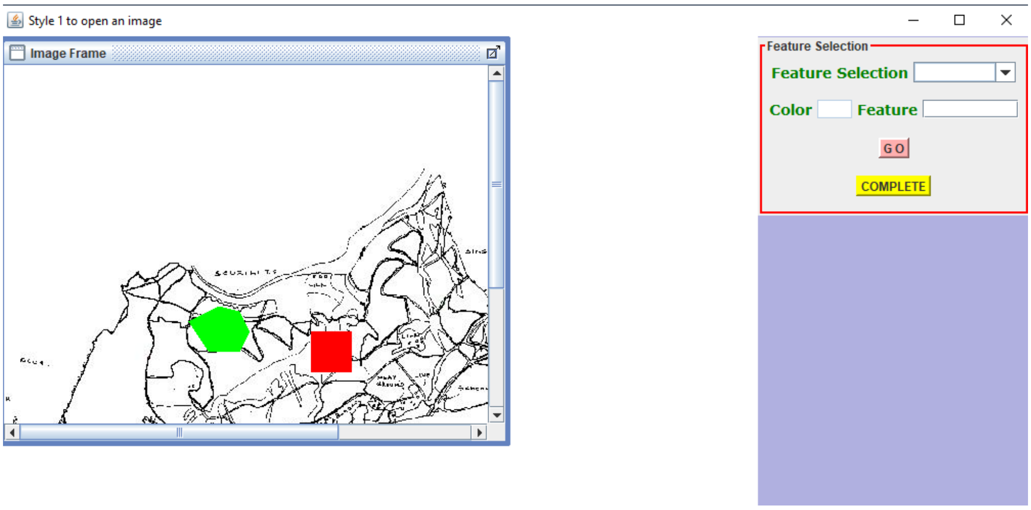
Option1:

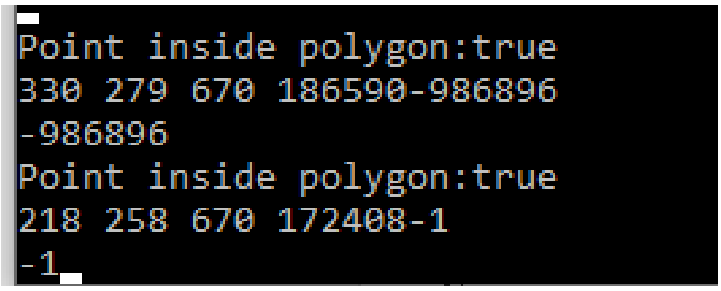
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;

class option3 extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
    Graphics2D gg1, gg2, gg3;
    Polygon pp1 = new Polygon();
    Polygon pp2 = new Polygon();
    Polygon pp3 = new Polygon();
    boolean startHovercurrent, startHoverprev = false;
    option3() {
        f = new JFrame("Style 3 to open an image");
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 200, 300);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        pp1.addPoint(130, 400);
        pp1.addPoint(180, 420);
        pp1.addPoint(150, 380);
        pp2.addPoint(300, 300);
        pp2.addPoint(340, 300);
        pp2.addPoint(340, 260);
        pp2.addPoint(300, 260);
        pp3.addPoint(200, 280);
        pp3.addPoint(230, 280);
        pp3.addPoint(240, 260);
        pp3.addPoint(230, 240);
        pp3.addPoint(210, 235);
        pp3.addPoint(180, 250);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
```

```
/*--------------------*/
JPanel p1a = new JPanel();
String ht1 = "<html><p><font color=\"GREEN\" " +
       "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
leb1 = new JLabel(ht1);
p1a.add(leb1);
cb1a = new JComboBox();
cb1a.setPreferredSize(new Dimension(100, 20));
cb1a.setEditable(false);
cb1a.setBackground(Color.WHITE);
cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
cb1a.addItem(" ");
// cb1a.addActionListener(this);
p1a.add(cb1a);
p1.add(p1a);
/*--------------------*/
JPanel p1b = new JPanel();
String ht2 = "<html><p><font color=\"GREEN\" " +
       "size=\"4\" face=\"Verdana\">Color</font> </p>";
leb2 = new JLabel(ht2);
p1b.add(leb2);
tf1 = new JTextField(3);
tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
tf1.setEditable(false);
tf1.setBackground(Color.white);
p1b.add(tf1);
/*--------------------*/
String ht3 = "<html><p><font color=\"GREEN\" " +
       "size=\"4\" face=\"Verdana\">Feature</font> </p>";
leb3 = new JLabel(ht3);
p1b.add(leb3);
tf2 = new JTextField(9);
tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
p1b.add(tf2);
p1.add(p1b);
/*--------------------*/
JPanel p1c = new JPanel();
fcb = new JButton(" G O ");
fcb.setBackground(Color.pink);
fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
fcb.addActionListener(this);
fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
p1c.add(fcb);
p1.add(p1c);
/*--------------------*/
JPanel p1d = new JPanel();
```

```
        cb = new JButton(" COMPLETE ");
        cb.setBackground(Color.yellow);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        cb.addActionListener(this);
        cb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1d.add(cb);
        p1.add(p1d);
        /*--------------------*/
        c.add(desk, BorderLayout.CENTER);
        c.add(panel, BorderLayout.EAST);
        f.setSize(1024, 738);
        f.setVisible(true);
    }

    public static void main(String args[]) {
        new option3();
    }

    public void mouseClicked(MouseEvent me) {
        boolean contain1, contain2, contain3;
        ColorModel md = pg.getColorModel();
        x = me.getX();
        y = me.getY();
        contain1 = pp1.contains(x, y);
        contain2 = pp2.contains(x, y);
        contain3 = pp3.contains(x, y);
        System.out.print("\nPoint inside polygon:" + (contain1 || contain2 || contain3) +
"\n");
        int a1 = (y - 1) * ii.getIconWidth() + x;
        System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + "" + pixels[a1]);
        rgb = (md.getRGB(pixels[a1]));
        System.out.print(rgb);
        if (contain1 == true) {
            Color color = new Color(255, 0, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(255, 0, 0));
        } else if (contain2 == true) {
            Color color = new Color(0, 255, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 255, 0));
        } else if (contain3 == true) {
            Color color = new Color(0, 0, 255);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 0, 255));
        } else
            tf1.setBackground(new Color(rgb));
    for (int i = 0; i < count; i++) {
        if (rgb == frgb[i]) {
            System.out.print(" " + fea[i] + "\n\n");
            break;
        }
    }
}

public void mouseEntered(MouseEvent me) {
}

public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}

public void mouseMoved(MouseEvent me) {
    ColorModel cm = pg.getColorModel();
    gg1 = (Graphics2D) l.getGraphics();
    gg1.setColor(new Color(255, 0, 0));
    gg1.fillPolygon(pp1);
    gg2 = (Graphics2D) l.getGraphics();
    gg2.setColor(new Color(0, 255, 0));
    gg2.fillPolygon(pp2);
    gg3 = (Graphics2D) l.getGraphics();
    gg3.setColor(new Color(0, 0, 255));
    gg3.fillPolygon(pp3);
    boolean contain1, contain2, contain3;
    // System.out.println("lol");
    int mx = me.getX();
    int my = me.getY();
    contain1 = pp1.contains(mx, my);
    contain2 = pp2.contains(mx, my);
    contain3 = pp3.contains(mx, my);
    if (contain1 || contain2 || contain3) {
        startHovercurrent = true;
        if (startHovercurrent != startHoverprev) {
            if (contain1 == true) {
                Color color = new Color(255, 0, 0);
                rgb = color.getRGB();
                tf1.setBackground(new Color(255, 0, 0));
            } else if (contain2 == true) {
                    Color color = new Color(0, 255, 0);
                    rgb = color.getRGB();
                    tf1.setBackground(new Color(0, 255, 0));
                } else if (contain3 == true) {
                    Color color = new Color(0, 0, 255);
                    rgb = color.getRGB();
                    tf1.setBackground(new Color(0, 0, 255));
                } else
                    tf1.setBackground(new Color(rgb));
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        System.out.print(" " + fea[i] + "\n\n");
                        break;
                    }
                }
            }
            startHoverprev = startHovercurrent;
        } else {
            startHovercurrent = false;
            // if(startHovercurrent!=startHoverprev)
            // System.out.println("Outside region");
            startHoverprev = startHovercurrent;
        }
    }

    public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }

            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            // allowing only three attribute here
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
        }
    }
}
```

OUTPUT of Option1:





```
Point inside polygon:true
330 279 670 186590-986896
-986896
Point inside polygon:true
218 258 670 172408-1
-1
```

Option2:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;

class option2 extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
    Graphics2D gg1, gg2, gg3;
    Polygon pp1 = new Polygon();
    Polygon pp2 = new Polygon();
    Polygon pp3 = new Polygon();
    boolean startHovercurrent, startHoverprev = false;

    option2() {
        f = new JFrame("Style 2 to open an image");
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();

        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 600, 600);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        pp1.addPoint(300, 300);
        pp1.addPoint(340, 300);
        pp1.addPoint(340, 260);
        pp1.addPoint(300, 260);
        pp2.addPoint(130, 400);
        pp2.addPoint(180, 420);
        pp2.addPoint(150, 380);
        pp3.addPoint(200, 280);
        pp3.addPoint(230, 280);
        pp3.addPoint(240, 260);
        pp3.addPoint(230, 240);
        pp3.addPoint(210, 235);
        pp3.addPoint(180, 250);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem(" ");
        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);
        /*--------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
```

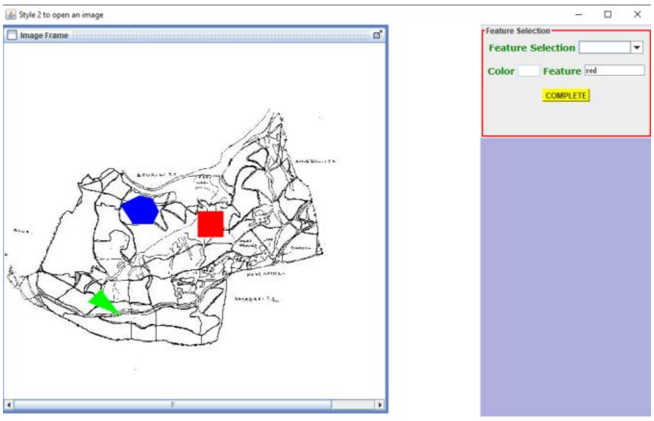
```
    tf1.setEditable(false);
    tf1.setBackground(Color.white);
    p1b.add(tf1);
    /*--------------------*/
    String ht3 = "<html><p><font color=\"GREEN\" " +
            "size=\"4\" face=\"Verdana\">Feature</font> </p>";
    leb3 = new JLabel(ht3);
    p1b.add(leb3);
    tf2 = new JTextField(9);
    tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
    p1b.add(tf2);
    p1.add(p1b);
    /*--------------------*
    JPanel p1c=new JPanel();
    fcb=new JButton(" G O ");
    fcb.setBackground(Color.pink);
    fcb.setBorder(new BevelBorder (BevelBorder.RAISED));
    fcb.addActionListener(this);
    fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
    p1c.add(fcb);
    p1.add(p1c);
    /*--------------------*/
    JPanel p1d = new JPanel();
    cb = new JButton(" COMPLETE ");
    cb.setBackground(Color.yellow);
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    cb.addActionListener(this);
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    p1d.add(cb);
    p1.add(p1d);
    /*--------------------*/
    c.add(desk, BorderLayout.CENTER);
    c.add(panel, BorderLayout.EAST);
    f.setSize(1024, 738);
    f.setVisible(true);
}

public static void main(String args[]) {
    new option2();
}

public void mouseClicked(MouseEvent me) {
    boolean contain1, contain2, contain3;
    ColorModel md = pg.getColorModel();
    x = me.getX();
    y = me.getY();
```

```
        contain1 = pp1.contains(x, y);
        contain2 = pp2.contains(x, y);
        contain3 = pp3.contains(x, y);
        System.out.print("\nPoint inside polygon:" + (contain1 || contain2 || contain3) +
"\n");
        int a1 = (y - 1) * ii.getIconWidth() + x;
        System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + "" + pixels[a1]);
        rgb = (md.getRGB(pixels[a1]));
        System.out.print(rgb);
        if (contain1 == true) {
            Color color = new Color(255, 0, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(255, 0, 0));
        } else if (contain2 == true) {
            Color color = new Color(0, 255, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 255, 0));
        } else if (contain3 == true) {
            Color color = new Color(0, 0, 255);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 0, 255));
        } else
            tf1.setBackground(new Color(rgb));
        for (int i = 0; i < count; i++) {
            if (rgb == frgb[i]) {
                System.out.print(" " + fea[i] + "\n\n");
                break;
            }
        }
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }

    public void mouseReleased(MouseEvent me) {
    }

    public void mouseMoved(MouseEvent me) {
        ColorModel cm = pg.getColorModel();
        gg1 = (Graphics2D) l.getGraphics();
```

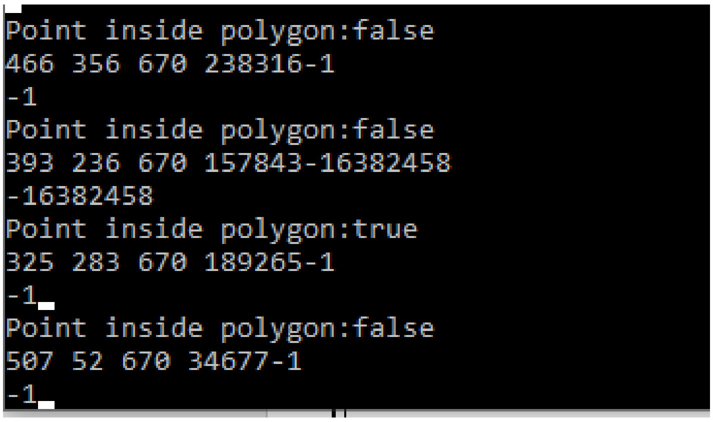
```
gg1.setColor(new Color(255, 0, 0));
gg1.fillPolygon(pp1);
gg2 = (Graphics2D) l.getGraphics();
gg2.setColor(new Color(0, 255, 0));
gg2.fillPolygon(pp2);
gg3 = (Graphics2D) l.getGraphics();
gg3.setColor(new Color(0, 0, 255));
gg3.fillPolygon(pp3);
boolean contain1, contain2, contain3;
// System.out.println("lol");
int mx = me.getX();
int my = me.getY();
contain1 = pp1.contains(mx, my);
contain2 = pp2.contains(mx, my);
contain3 = pp3.contains(mx, my);
if (contain1 || contain2 || contain3) {
    startHovercurrent = true;
    if (startHovercurrent != startHoverprev) {
        if (contain1 == true) {
            Color color = new Color(255, 0, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(255, 0, 0));
        } else if (contain2 == true) {
            Color color = new Color(0, 255, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 255, 0));
        } else if (contain3 == true) {
            Color color = new Color(0, 0, 255);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 0, 255));
        } else
            tf1.setBackground(new Color(rgb));
        for (int i = 0; i < count; i++) {
            if (rgb == frgb[i]) {
                System.out.print(" " + fea[i] + "\n\n");
                break;
            }
        }
    }
    startHoverprev = startHovercurrent;
} else {
    startHovercurrent = false;
    // if(startHovercurrent!=startHoverprev)
    // System.out.println("Outside region");
    startHoverprev = startHovercurrent;
}
```

```
    }

    public void mouseDragged(MouseEvent me) {
    }
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }
            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            // allowing only three attribute here
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
        }
    }
}
```

OUTPUT of Option2:



Feature red



```
Point inside polygon:false
466 356 670 238316-1
-1
Point inside polygon:false
393 236 670 157843-16382458
-16382458
Point inside polygon:true
325 283 670 189265-1
-1
Point inside polygon:false
507 52 670 34677-1
-1
```

Option3:

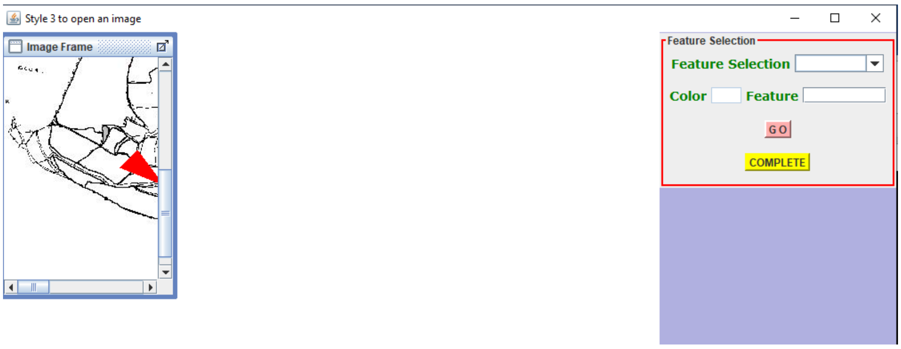
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;

class option3 extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
    Graphics2D gg1, gg2, gg3;
    Polygon pp1 = new Polygon();
    Polygon pp2 = new Polygon();
    Polygon pp3 = new Polygon();
    boolean startHovercurrent, startHoverprev = false;

    option3() {
        f = new JFrame("Style 3 to open an image");

        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();
        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 200, 300);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        pp1.addPoint(130, 400);
        pp1.addPoint(180, 420);
        pp1.addPoint(150, 380);
        pp2.addPoint(300, 300);
        pp2.addPoint(340, 300);
        pp2.addPoint(340, 260);
        pp2.addPoint(300, 260);
        pp3.addPoint(200, 280);
        pp3.addPoint(230, 280);
        pp3.addPoint(240, 260);
        pp3.addPoint(230, 240);
        pp3.addPoint(210, 235);
        pp3.addPoint(180, 250);
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);
        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " +
                "size=\"4\" face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
```

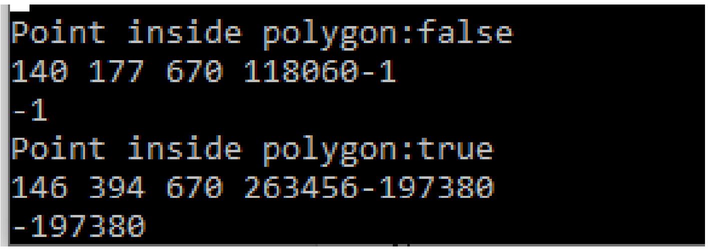
```
cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
cb1a.addItem(" ");
// cb1a.addActionListener(this);
p1a.add(cb1a);
p1.add(p1a);
/*--------------------*/
JPanel p1b = new JPanel();
String ht2 = "<html><p><font color=\"GREEN\" " +
       "size=\"4\" face=\"Verdana\">Color</font> </p>";
leb2 = new JLabel(ht2);
p1b.add(leb2);
tf1 = new JTextField(3);
tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
tf1.setEditable(false);
tf1.setBackground(Color.white);
p1b.add(tf1);
/*--------------------*/
String ht3 = "<html><p><font color=\"GREEN\" " +
       "size=\"4\" face=\"Verdana\">Feature</font> </p>";
leb3 = new JLabel(ht3);
p1b.add(leb3);
tf2 = new JTextField(9);
tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
p1b.add(tf2);
p1.add(p1b);
/*--------------------*/
JPanel p1c = new JPanel();
fcb = new JButton(" G O ");
fcb.setBackground(Color.pink);
fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
fcb.addActionListener(this);
fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
p1c.add(fcb);
p1.add(p1c);
/*--------------------*/
JPanel p1d = new JPanel();
cb = new JButton(" COMPLETE ");
cb.setBackground(Color.yellow);
cb.setBorder(new BevelBorder(BevelBorder.RAISED));
cb.addActionListener(this);
cb.setBorder(new BevelBorder(BevelBorder.RAISED));
p1d.add(cb);
p1.add(p1d);
/*--------------------*/
c.add(desk, BorderLayout.CENTER);
c.add(panel, BorderLayout.EAST);
```

```
        f.setSize(1024, 738);
        f.setVisible(true);
    }

    public static void main(String args[]) {
        new option3();
    }

    public void mouseClicked(MouseEvent me) {
        boolean contain1, contain2, contain3;
        ColorModel md = pg.getColorModel();
        x = me.getX();
        y = me.getY();
        contain1 = pp1.contains(x, y);
        contain2 = pp2.contains(x, y);
        contain3 = pp3.contains(x, y);
        System.out.print("\nPoint inside polygon:" + (contain1 || contain2 || contain3) +
"\n");
        int a1 = (y - 1) * ii.getIconWidth() + x;
        System.out.println(x + " " + y + " " + ii.getIconWidth() + " " + a1 + "" + pixels[a1]);
        rgb = (md.getRGB(pixels[a1]));
        System.out.print(rgb);
        if (contain1 == true) {
            Color color = new Color(255, 0, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(255, 0, 0));
        } else if (contain2 == true) {
            Color color = new Color(0, 255, 0);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 255, 0));
        } else if (contain3 == true) {
            Color color = new Color(0, 0, 255);
            rgb = color.getRGB();
            tf1.setBackground(new Color(0, 0, 255));
        } else
            tf1.setBackground(new Color(rgb));
        for (int i = 0; i < count; i++) {
            if (rgb == frgb[i]) {
                System.out.print(" " + fea[i] + "\n\n");
                break;
            }
        }
    }

    public void mouseEntered(MouseEvent me) {
    }
```

```
public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}

public void mouseMoved(MouseEvent me) {
    ColorModel cm = pg.getColorModel();
    gg1 = (Graphics2D) l.getGraphics();
    gg1.setColor(new Color(255, 0, 0));
    gg1.fillPolygon(pp1);
    gg2 = (Graphics2D) l.getGraphics();
    gg2.setColor(new Color(0, 255, 0));
    gg2.fillPolygon(pp2);
    gg3 = (Graphics2D) l.getGraphics();
    gg3.setColor(new Color(0, 0, 255));
    gg3.fillPolygon(pp3);
    boolean contain1, contain2, contain3;
    // System.out.println("lol");
    int mx = me.getX();
    int my = me.getY();
    contain1 = pp1.contains(mx, my);
    contain2 = pp2.contains(mx, my);
    contain3 = pp3.contains(mx, my);
    if (contain1 || contain2 || contain3) {
        startHovercurrent = true;
        if (startHovercurrent != startHoverprev) {
            if (contain1 == true) {
                Color color = new Color(255, 0, 0);
                rgb = color.getRGB();
                tf1.setBackground(new Color(255, 0, 0));
            } else if (contain2 == true) {
                Color color = new Color(0, 255, 0);
                rgb = color.getRGB();
                tf1.setBackground(new Color(0, 255, 0));
            } else if (contain3 == true) {
                Color color = new Color(0, 0, 255);
                rgb = color.getRGB();
                tf1.setBackground(new Color(0, 0, 255));
            } else
                tf1.setBackground(new Color(rgb));
            for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        System.out.print(" " + fea[i] + "\n\n");
                        break;
                    }
                }
            }
            startHoverprev = startHovercurrent;
        } else {
            startHovercurrent = false;
            // if(startHovercurrent!=startHoverprev)
            // System.out.println("Outside region");
            startHoverprev = startHovercurrent;
        }
    }

    public void mouseDragged(MouseEvent me) {
    }
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
completed before",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature",
                        "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }
                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before",
                                "NOTIFICATION", JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }

            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            // allowing only three attribute here
            if (count == 3) {
                fcb.setVisible(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + " is for the feature " + fea[i]);
                }
                System.out.print("\n\n");
                ccb = 1;
            }
        }
    }
}
```

OUTPUT of Option2:





```
Point inside polygon:false
140 177 670 118060-1
-1
Point inside polygon:true
146 394 670 263456-197380
-197380
```

16. Write a program to perform the following operations click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the
corresponding text field...session1.gif image has four features they are -
1. Rock
2. River
3. moist land
4. bare soil

PROGRAM:

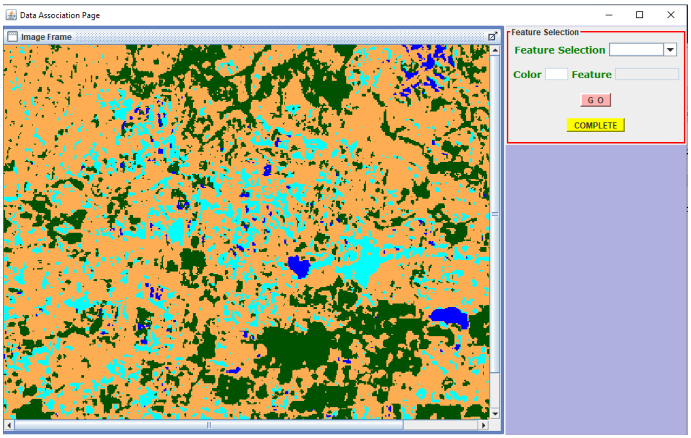
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.io.Console;
import java.awt.Graphics.*;

class Program extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3, leb4;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
```

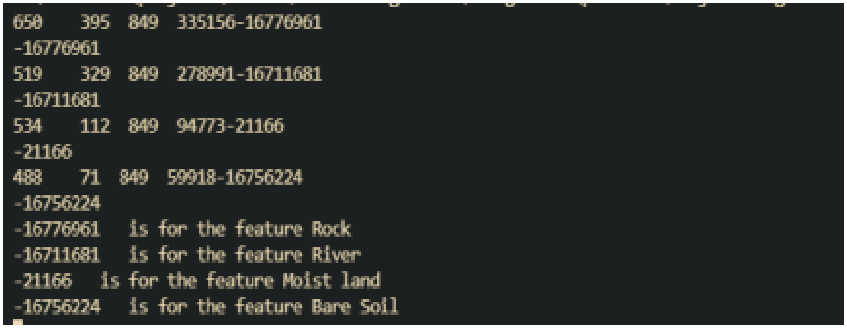
```
  Program() {
    f = new JFrame("Data Association Page");
    ii = new ImageIcon("session1.gif");
    img = ii.getImage();
    height = ii.getIconHeight();
    width = ii.getIconWidth();
    pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
    pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
    try {
      pg.grabPixels();
    } catch (InterruptedException k) {
    }
    l = new JLabel(ii, JLabel.CENTER);
    c = f.getContentPane();
    JDesktopPane desk = new JDesktopPane();

    JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
    JScrollPane scroll = new JScrollPane(l);
    p.setContentPane(scroll);
    p.setBounds(0, 0, 740, 600);
    desk.add(p);
    p.setVisible(true);
    l.addMouseListener(this);
    l.addMouseMotionListener(this);

    panel = new JPanel();
    panel.setLayout(new GridLayout(4, 1));
    panel.setBackground(cc);

    p1 = new JPanel();
    p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
    p1.setLayout(new GridLayout(4, 1));
    panel.add(p1);
    /*--------------------*/
    JPanel p1a = new JPanel();
    String ht1 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature Selection</font> </p>";
    leb1 = new JLabel(ht1);
    p1a.add(leb1);

    cb1a = new JComboBox();
    cb1a.setPreferredSize(new Dimension(100, 20));
    cb1a.setEditable(false);
```

```
	cb1a.setBackground(Color.WHITE);
	cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
	cb1a.addItem("           ");

	// cb1a.addActionListener(this);
	p1a.add(cb1a);
	p1.add(p1a);

	/*--------------------*/
	JPanel p1b = new JPanel();
	String ht2 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Color</font> </p>";
	leb2 = new JLabel(ht2);
	p1b.add(leb2);
	tf1 = new JTextField(3);
	tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
	tf1.setEditable(false);
	tf1.setBackground(Color.white);
	p1b.add(tf1);

	/*--------------------*/

	String ht3 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature</font> </p>";
	leb3 = new JLabel(ht3);
	p1b.add(leb3);
	tf2 = new JTextField(9);
	tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
	p1b.add(tf2);

	p1.add(p1b);
	/*--------------------*/

	JPanel p1c = new JPanel();
	fcb = new JButton("   G  O   ");
	fcb.setBackground(Color.pink);
	fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
	fcb.addActionListener(this);
	fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
	p1c.add(fcb);

	p1.add(p1c);
	/*--------------------*/
	JPanel p1d = new JPanel();
	cb = new JButton("   COMPLETE   ");
	cb.setBackground(Color.yellow);
```

```
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    cb.addActionListener(this);
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    p1d.add(cb);

    p1.add(p1d);
    /*---------------------*/
    c.add(desk, BorderLayout.CENTER);
    c.add(panel, BorderLayout.EAST);
    f.setSize(1024, 738);
    f.setVisible(true);

}

public static void main(String args[]) {
    new Program();
}

public void mouseClicked(MouseEvent me) {
    ColorModel md = pg.getColorModel();
    x = me.getX();
    y = me.getY();
    int a1 = (y - 1) * ii.getIconWidth() + x;
    System.out.println(x + "    " + y + "  " + ii.getIconWidth() + "  " + a1 + pixels[a1]);
    // System.out.println(a1);
    rgb = (md.getRGB(pixels[a1]));
    System.out.println(rgb);
    tf1.setBackground(new Color(rgb));
}

public void mouseEntered(MouseEvent me) {
}

public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}

public void mouseMoved(MouseEvent me) {
}

public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated before", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }

            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }

                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }
            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            if (count == 4 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated", "NOTIFICATION",
                    JOptionPane.ERROR_MESSAGE);
                tf1.setBackground(Color.white);
                tf2.setText(null);
                tf2.setEditable(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + "   is for the feature " + fea[i]);
                }
                ccb = 1;
            }
        }
    }
}
```

OUTPUT:





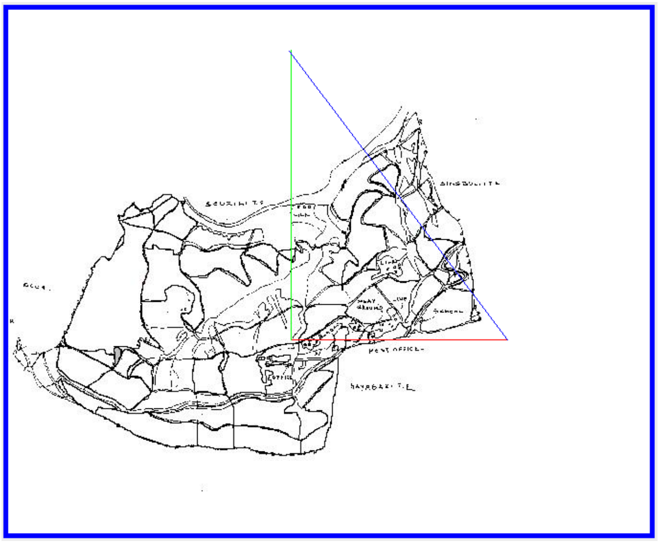
```
650    395  849  335156-16776961
-16776961
519    329  849  278991-16711681
-16711681
534    112  849  94773-21166
-21166
488    71  849  59918-16756224
-16756224
-16776961   is for the feature Rock
-16711681   is for the feature River
-21166   is for the feature Moist land
-16756224   is for the feature Bare Soil
```

17. Write a program to draw line(6cm,8cm,10cm) in three different places and point into a GIS image so that they can form a triangle.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    public static double getDPI(){
        double dpi = Toolkit.getDefaultToolkit().getScreenResolution();
        // System.out.println("Screen DPI: " + dpi);
        return dpi;
    }

    public static double cmToPixel(double cm){
        double dpi = getDPI();
        double pixel = cm * dpi / 2.54;
        // System.out.println("cm: " + cm + " pixel: " + pixel);
        return pixel;
    }
    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);

    }

    public static double[] drawLine(double length, double angle, int x, int y,
Graphics2D gg, Color color) {
        double radians = Math.toRadians(angle);
        double x1 = x + length * Math.cos(radians);
        double y1 = y - length * Math.sin(radians);
        gg.setColor(color);
        gg.drawLine((int) x, (int) y, (int) x1, (int) y1);
        double[] end = new double[2];
        end[0] = x1;
        end[1] = y1;
        return end;
    }

    public void mouseClicked(MouseEvent me) {
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }

    public void mousePressed(MouseEvent me) {
    }
    public void mouseReleased(MouseEvent me) {
    }

    public void mouseMoved(MouseEvent me) {

        ColorModel cm = pg.getColorModel();
        gg = (Graphics2D) l.getGraphics();

        // Drawing line 2cm, 6cm, 9cm long

        int startX = 300;
        int startY = 350;
        int angleA = 0;
        int angleB = 90;
        int angleC = 127;

        double[] endA = drawLine(cmToPixel(6), angleA, startX, startY, gg, Color.RED);
        double[] endB = drawLine(cmToPixel(8), angleB, startX, startY, gg,
Color.GREEN);
        drawLine(cmToPixel(10), angleC, (int)endA[0], (int)endA[1], gg, Color.BLUE);
    }

    public void mouseDragged(MouseEvent me) {
    }

    public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:



18. Write a program to perform the following operations click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the corresponding text field...session1.gif image has four features they are –

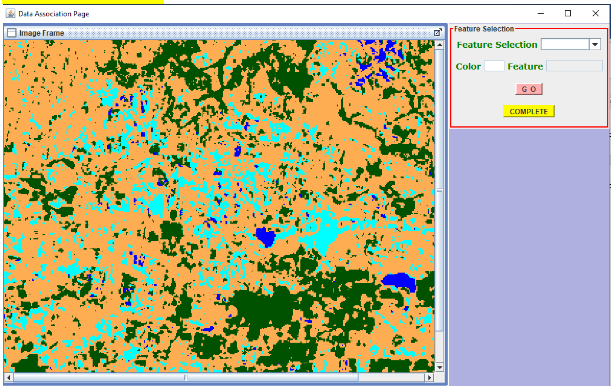
1. Vegetation
2. water body
3. moist land
4. bare soil

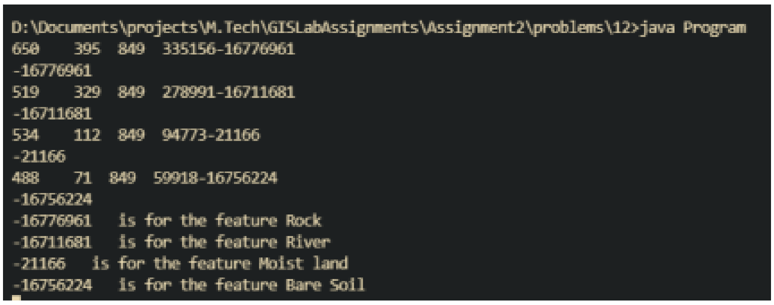
PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.io.Console;
import java.awt.Graphics.*;

class Program extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3, leb4;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;

    Program() {
        f = new JFrame("Data Association Page");
        ii = new ImageIcon("session1.gif");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();

        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 740, 600);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);

        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);

        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);
        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem("            ");

        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);

        /*--------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        tf1.setEditable(false);
        tf1.setBackground(Color.white);
        p1b.add(tf1);

        /*--------------------*/

        String ht3 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature</font> </p>";
        leb3 = new JLabel(ht3);
        p1b.add(leb3);
        tf2 = new JTextField(9);
        tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        p1b.add(tf2);

        p1.add(p1b);
        /*--------------------*/

        JPanel p1c = new JPanel();
        fcb = new JButton("   G  O   ");
        fcb.setBackground(Color.pink);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        fcb.addActionListener(this);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1c.add(fcb);

        p1.add(p1c);
        /*--------------------*/
```

```
    JPanel p1d = new JPanel();
    cb = new JButton("   COMPLETE   ");
    cb.setBackground(Color.yellow);
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    cb.addActionListener(this);
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    p1d.add(cb);

    p1.add(p1d);
    /*---------------------*/
    c.add(desk, BorderLayout.CENTER);
    c.add(panel, BorderLayout.EAST);
    f.setSize(1024, 738);
    f.setVisible(true);

}

public static void main(String args[]) {
    new Program();
}

public void mouseClicked(MouseEvent me) {
    ColorModel md = pg.getColorModel();
    x = me.getX();
    y = me.getY();
    int a1 = (y - 1) * ii.getIconWidth() + x;
    System.out.println(x + "    " + y + "  " + ii.getIconWidth() + "  " + a1 + pixels[a1]);
    // System.out.println(a1);
    rgb = (md.getRGB(pixels[a1]));
    System.out.println(rgb);
    tf1.setBackground(new Color(rgb));
}

public void mouseEntered(MouseEvent me) {
}

public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}

public void mouseMoved(MouseEvent me) {
    }

    public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated before", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }

            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }

                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }
            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            if (count == 4 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                tf1.setBackground(Color.white);
                tf2.setText(null);
                tf2.setEditable(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + "   is for the feature " + fea[i]);
                }
                ccb = 1;
            }
        }
    }
}
```

OUTPUT:





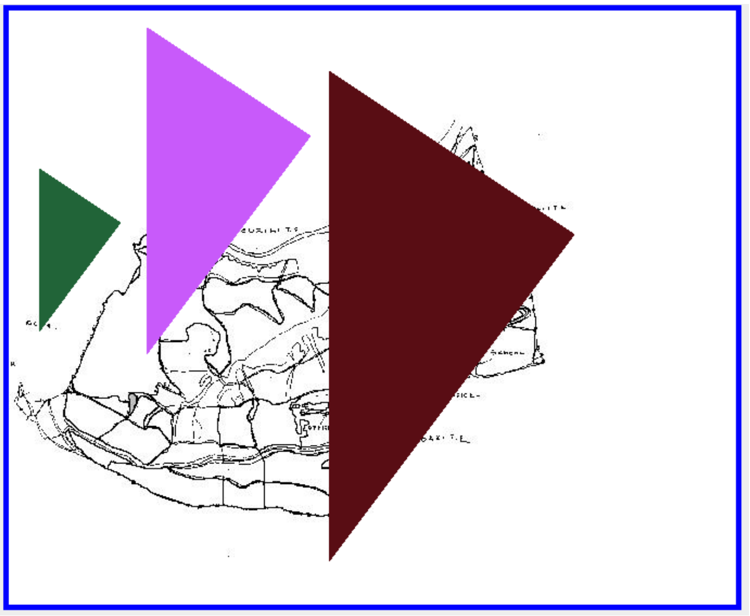
```
D:\Documents\projects\M.Tech\GISLabAssignments\Assignment2\problems\12>java Program
650    395  849  335156-16776961
-16776961
519    329  849  278991-16711681
-16711681
534    112  849  94773-21166
-21166
488    71  849  59918-16756224
-16756224
-16776961   is for the feature Rock
-16711681   is for the feature River
-21166   is for the feature Moist land
-16756224   is for the feature Bare Soil
```

19. Write a GIS based program to draw a polygon(four different places with 4 cm,8cm,12cm width) into a GIS image.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.color.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.io.File;

class Program extends JFrame implements MouseListener, MouseMotionListener {
    JFrame f;
    JLabel l;
    JPanel panel;
    Container c;
    ImageIcon ii;
    Image img;
    int pixels[];
    PixelGrabber pg;
    ColorModel cm;
    Graphics2D gg;

    public static double getDPI(){
        double dpi = Toolkit.getDefaultToolkit().getScreenResolution();
        // System.out.println("Screen DPI: " + dpi);
        return dpi;
    }

    public static double cmToPixel(double cm){
        double dpi = getDPI();
        double pixel = cm * dpi / 2.54;
        // System.out.println("cm: " + cm + " pixel: " + pixel);
        return pixel;
    }

    Program() {
        f = new JFrame("Drawing on an image");
        f.setSize(1024, 738);
        ii = new ImageIcon("prasun.jpg");
        img = ii.getImage();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);
        l.setBorder(BorderFactory.createLineBorder(Color.blue, 5));
        panel = new JPanel();
        panel.setLayout(new FlowLayout());
        panel.add(l);

        c = f.getContentPane();
        c.add(panel);
        f.setVisible(true);
    }

    public void mouseClicked(MouseEvent me) {
        gg = (Graphics2D) l.getGraphics();

        Polygon p1 = drawPolygon(cmToPixel(4), 25, 50);
        gg.setColor(new Color(200, 13, 40));
        gg.fillPolygon(p1);

        Polygon p2 = drawPolygon(cmToPixel(8), 130, 20);
        gg.setColor(new Color(100, 13, 40));
        gg.fillPolygon(p2);

        Polygon p3 = drawPolygon(cmToPixel(12), 300, 20);
        gg.setColor(new Color(0, 13, 40));
        gg.fillPolygon(p3);

        Polygon p4 = drawPolygon(cmToPixel(3), 25, 200);
        gg.setColor(new Color(0, 13, 40));
        gg.fillPolygon(p4);
    }

    public void mouseEntered(MouseEvent me) {
    }

    public void mouseExited(MouseEvent me) {
    }
    public void mousePressed(MouseEvent me) {
    }

    public void mouseReleased(MouseEvent me) {
    }

    public Polygon drawPolygon (double width, double x, double y) {
        Polygon p = new Polygon();
        p.addPoint((int)x, (int)y);
        p.addPoint((int)(x), (int)(y+ width));
        p.addPoint((int)(x + width/2), (int)(y + width/2));
        return p;
    }

    public void mouseMoved(MouseEvent me) {

    }

    public void mouseDragged(MouseEvent me) {
    }

    public static void main(String aa[]) {
        new Program();
    }
}
```

OUTPUT:



20. Write a program to perform the following operations click the mouse on a particular color(feature) of the image then the RGB value of the selected feature will be calculated. Then assign the name of the feature in the corresponding text field...session1.gif image has four features they are -

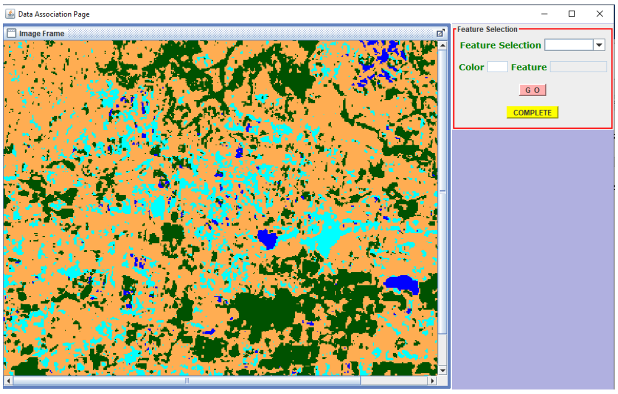
1. Vegetation
2. water body
3. moist land
4. bare soil

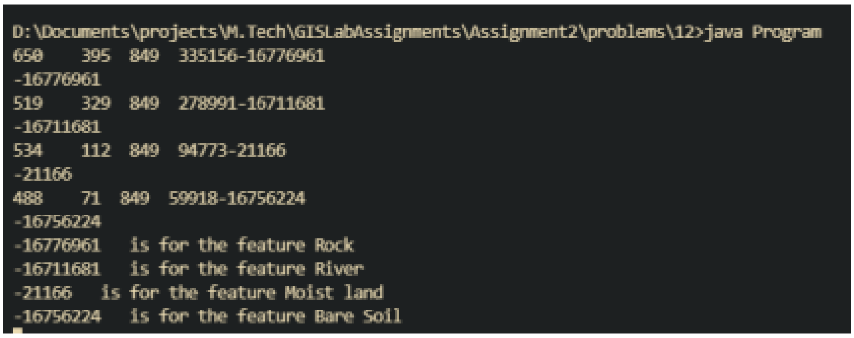
PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.image.*;
import java.io.Console;
import java.awt.Graphics.*;

class Program extends JFrame implements MouseListener, MouseMotionListener,
ActionListener {
    JFrame f;
    JLabel l;
    JPanel p1;
    ImageIcon ii;
    Image img;
    int height;
    int width;
    Container c;
    int pixels[];
    PixelGrabber pg;
    JPanel panel;
    JLabel leb1, leb2, leb3, leb4;
    JComboBox cb1a;
    JTextField tf1, tf2;
    JButton fcb, cb;
    Color cc = new Color(160, 160, 220, 200);
    int x, y;
    int rgb;
    int frgb[] = new int[4];
    String fea[] = new String[4];
    int count = 0;
    String ss;
    int ccb = 0;
    Program() {
        f = new JFrame("Data Association Page");
        ii = new ImageIcon("session1.gif");
        img = ii.getImage();
        height = ii.getIconHeight();
        width = ii.getIconWidth();
        pixels = new int[ii.getIconWidth() * ii.getIconHeight()];
        pg = new PixelGrabber(img, 0, 0, ii.getIconWidth(), ii.getIconHeight(), pixels, 0,
ii.getIconWidth());
        try {
            pg.grabPixels();
        } catch (InterruptedException k) {
        }
        l = new JLabel(ii, JLabel.CENTER);
        c = f.getContentPane();
        JDesktopPane desk = new JDesktopPane();

        JInternalFrame p = new JInternalFrame("Image Frame", false, false, true,
false);
        JScrollPane scroll = new JScrollPane(l);
        p.setContentPane(scroll);
        p.setBounds(0, 0, 740, 600);
        desk.add(p);
        p.setVisible(true);
        l.addMouseListener(this);
        l.addMouseMotionListener(this);

        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));
        panel.setBackground(cc);

        p1 = new JPanel();
        p1.setBorder(new TitledBorder(new LineBorder(Color.red, 2), "Feature
Selection"));
        p1.setLayout(new GridLayout(4, 1));
        panel.add(p1);
        /*--------------------*/
        JPanel p1a = new JPanel();
        String ht1 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature Selection</font> </p>";
        leb1 = new JLabel(ht1);
        p1a.add(leb1);

        cb1a = new JComboBox();
        cb1a.setPreferredSize(new Dimension(100, 20));
        cb1a.setEditable(false);
        cb1a.setBackground(Color.WHITE);
        cb1a.setFont(new Font("Dialog", Font.BOLD, 10));
        cb1a.addItem("            ");

        // cb1a.addActionListener(this);
        p1a.add(cb1a);
        p1.add(p1a);

        /*--------------------*/
        JPanel p1b = new JPanel();
        String ht2 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Color</font> </p>";
        leb2 = new JLabel(ht2);
        p1b.add(leb2);
        tf1 = new JTextField(3);
        tf1.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        tf1.setEditable(false);
        tf1.setBackground(Color.white);
        p1b.add(tf1);
        /*--------------------*/

        String ht3 = "<html><p><font color=\"GREEN\" " + "size=\"4\"
face=\"Verdana\">Feature</font> </p>";
        leb3 = new JLabel(ht3);
        p1b.add(leb3);
        tf2 = new JTextField(9);
        tf2.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 12));
        p1b.add(tf2);

        p1.add(p1b);
        /*--------------------*/

        JPanel p1c = new JPanel();
        fcb = new JButton("   G  O   ");
        fcb.setBackground(Color.pink);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        fcb.addActionListener(this);
        fcb.setBorder(new BevelBorder(BevelBorder.RAISED));
        p1c.add(fcb);

        p1.add(p1c);
        /*--------------------*/
        JPanel p1d = new JPanel();
        cb = new JButton("   COMPLETE   ");
        cb.setBackground(Color.yellow);
```

```
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    cb.addActionListener(this);
    cb.setBorder(new BevelBorder(BevelBorder.RAISED));
    p1d.add(cb);

    p1.add(p1d);
    /*---------------------*/
    c.add(desk, BorderLayout.CENTER);
    c.add(panel, BorderLayout.EAST);
    f.setSize(1024, 738);
    f.setVisible(true);

}

public static void main(String args[]) {
    new Program();
}

public void mouseClicked(MouseEvent me) {
    ColorModel md = pg.getColorModel();
    x = me.getX();
    y = me.getY();
    int a1 = (y - 1) * ii.getIconWidth() + x;
    System.out.println(x + "    " + y + "  " + ii.getIconWidth() + "  " + a1 + pixels[a1]);
    // System.out.println(a1);
    rgb = (md.getRGB(pixels[a1]));
    System.out.println(rgb);
    tf1.setBackground(new Color(rgb));
}

public void mouseEntered(MouseEvent me) {
}

public void mouseExited(MouseEvent me) {
}

public void mousePressed(MouseEvent me) {
}

public void mouseReleased(MouseEvent me) {
}

public void mouseMoved(MouseEvent me) {
}

public void mouseDragged(MouseEvent me) {
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == fcb) {
            int flag = 0;
            if (count == 4) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated before", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
                tf1.setBackground(Color.white);
                tf2.setText(null);
            }
            ss = tf2.getText();
            if (ss.length() == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Assign
Feature Name", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }
            if (rgb == 0 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Please Select
Feature", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                flag = 1;
            }

            if (flag == 0) {
                for (int i = 0; i < count; i++) {
                    if (rgb == frgb[i]) {
                        JOptionPane.showMessageDialog((Component) null, "Please Select
New Feature", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                    }

                    if (ss.equalsIgnoreCase(fea[i])) {
                        JOptionPane.showMessageDialog((Component) null, "Feature Name
Used Before", "NOTIFICATION",
                                JOptionPane.ERROR_MESSAGE);
                        flag = 1;
                        tf2.setText(null);
                    }
                }
            }
            if (flag == 0) {
                frgb[count] = rgb;
                // System.out.println(rgb);
                fea[count] = ss;
                count++;
                tf1.setBackground(Color.white);
                tf2.setText(null);
                rgb = 0;
            }
            if (count == 4 && flag == 0) {
                JOptionPane.showMessageDialog((Component) null, "Assignment
compleated", "NOTIFICATION",
                        JOptionPane.ERROR_MESSAGE);
                tf1.setBackground(Color.white);
                tf2.setText(null);
                tf2.setEditable(false);
            }
        }
        if (ae.getSource() == cb) {
            if (count == 4 && ccb == 0) {
                for (int i = 0; i < 4; i++) {
                    cb1a.addItem(fea[i]);
                    System.out.println(frgb[i] + "   is for the feature " + fea[i]);
                }
                ccb = 1;
            }
        }
    }
}
```

OUTPUT:





```
D:\Documents\projects\M.Tech\GISLabAssignments\Assignment2\problems\12>java Program
650    395  849  335156-16776961
-16776961
519    329  849  278991-16711681
-16711681
534    112  849  94773-21166
-21166
488    71  849  59918-16756224
-16756224
-16776961   is for the feature Rock
-16711681   is for the feature River
-21166   is for the feature Moist land
-16756224   is for the feature Bare Soil
```