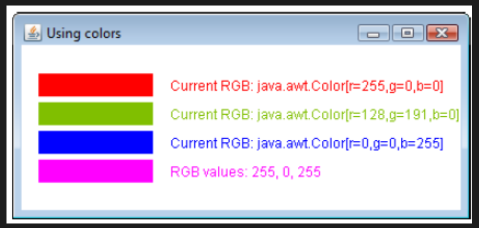
1

```
import javax.swing.JFrame;

public class ShowColors  {
 public static void main(String[] args) {
   JFrame frame = new JFrame("Using colors");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   ColorJPanel colorJPanel = new ColorJPanel();
   frame.add(colorJPanel);
   frame.setSize(400, 180);
   frame.setVisible(true);
 }
}
```



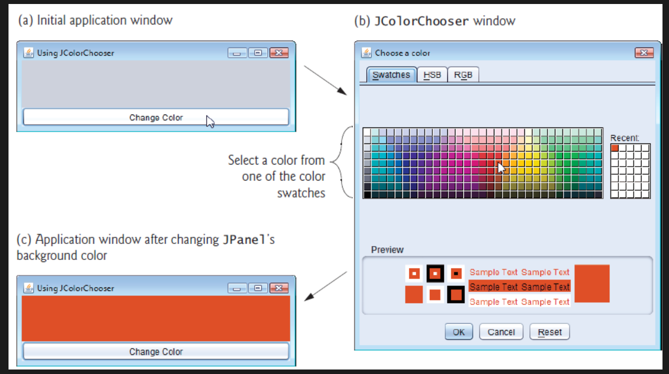
Using colors

Current RGB: java.awt.Color[r=255,g=0,b=0]
Current RGB: java.awt.Color[r=128,g=191,b=0]
Current RGB: java.awt.Color[r=0,g=0,b=255]
RGB values: 255, 0, 255

```
import java.awt.Graphics;
import java.awt.Color;
import javax.swing.JPanel;

public class ColorJPanel extends JPanel {
 @Override public void paintComponent(Graphics g) {
   super.paintComponent(g);
   this.setBackground(Color.WHITE);

   g.setColor(new Color(255, 0, 0));
   g.fillRect(15, 25, 100, 20);
   g.drawString("Current RGB: " + g.getColor(), 130, 40);

   g.setColor(new Color(0.50f, 0.75f, 0.0f));
   g.fillRect(15, 50, 100, 20);
   g.drawString("Current RGB: " + g.getColor(), 130, 65);

   g.setColor(Color.BLUE);
   g.fillRect(15, 75, 100, 20);
   g.drawString("Current RGB: " + g.getColor(), 130, 90);

   Color color = Color.MAGENTA;
   g.setColor(color);
   g.fillRect(15, 100, 100, 20);
   g.drawString("RGB values: " +
                color.getRed() +
                ", " +
                color.getGreen() +
                ", " + color.getBlue(),
                130, 115);
 }
}
```

2

```
import javax.swing.JFrame;

public class ShowColors2   {
 public static void main(String[] args) {
   ShowColors2JFrame application = new ShowColors2JFrame();
   application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }
}
```



(a) Initial application window

(b) JColorChooser window

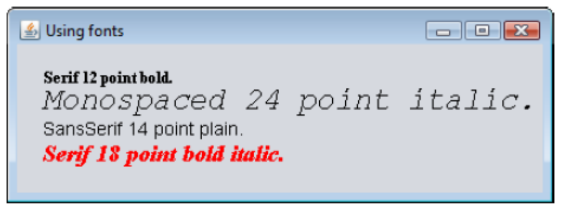
Select a color from one of the color swatches

(c) Application window after changing JPanel's background color

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JColorChooser;
import javax.swing.JPanel;

public class ShowColors2JFrame extends JFrame {
 private final JButton changeColorJButton;
 private       Color color = Color.LIGHT_GRAY;
 private final JPanel colorJPanel;

 public ShowColors2JFrame()  {
   super("Using JColorChooser");

   colorJPanel = new JPanel();
   colorJPanel.setBackground(color);

   changeColorJButton = new JButton("Change Color");
   changeColorJButton.addActionListener(
    new ActionListener() {
     @Override  public void actionPerformed(ActionEvent event) {
       color = JColorChooser.showDialog(
        ShowColors2JFrame.this, "Choose a color", color);

       if (color == null)
          color = Color.LIGHT_GRAY;

       colorJPanel.setBackground(color);
     }
    }
   );

   add(colorJPanel, BorderLayout.CENTER);
   add(changeColorJButton, BorderLayout.SOUTH);

   setSize(400, 130);
   setVisible(true);
 }
}
```

3

```
import javax.swing.JFrame;

public class Fonts {
  public static void main(String[] args) {

   JFrame frame = new JFrame("Using fonts");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   FontJPanel fontJPanel = new FontJPanel();
   frame.add(fontJPanel);
   frame.setSize(420, 150);
   frame.setVisible(true);
 }
}
```



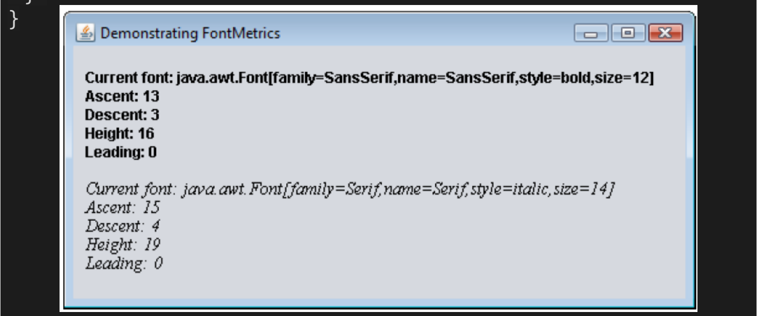
Using fonts

Serif 12 point bold.
Monospaced 24 point italic.
SansSerif 14 point plain.
Serif 18 point bold italic.

```
import java.awt.Font;
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;

public class FontJPanel extends JPanel {
 @Override public void paintComponent(Graphics g) {
   super.paintComponent(g);

   g.setFont(new Font("Serif", Font.BOLD, 12));
   g.drawString("Serif 12 point bold.", 20, 30);

   g.setFont(new Font("Monospaced", Font.ITALIC, 24));
   g.drawString("Monospaced 24 point italic.", 20, 50);

   g.setFont(new Font("SansSerif", Font.PLAIN, 14));
   g.drawString("SansSerif 14 point plain.", 20, 70);

   g.setColor(Color.RED);
   g.setFont(new Font("Serif", Font.BOLD + Font.ITALIC, 18));
   g.drawString(g.getFont().getName() + " " +
                g.getFont().getSize() +
                " point bold italic.", 20, 90);
 }
}
```

4

```
import javax.swing.JFrame;

public class Metrics {

 public static void main(String[] args) {
   JFrame frame = new JFrame("Demonstrating FontMetrics");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   MetricsJPanel metricsJPanel = new MetricsJPanel();
   frame.add(metricsJPanel);
   frame.setSize(510, 240);
   frame.setVisible(true);
 }
}
```



Demonstrating FontMetrics

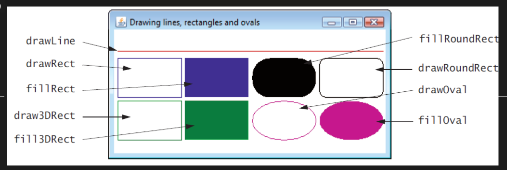
Current font: java.awt.Font[family=SansSerif,name=SansSerif,style=bold,size=12]
Ascent: 13
Descent: 3
Height: 16
Leading: 0

Current font: java.awt.Font[family=Serif,name=Serif,style=italic,size=14]
Ascent: 15
Descent: 4
Height: 19
Leading: 0

```
import java.awt.Font;
import java.awt.FontMetrics;
import java.awt.Graphics;
import javax.swing.JPanel;

public class MetricsJPanel extends JPanel {
  @Override public void paintComponent(Graphics g) {
   super.paintComponent(g);

   g.setFont(new Font("SansSerif", Font.BOLD, 12));
   FontMetrics metrics = g.getFontMetrics();

   g.drawString("Current font: " + g.getFont(), 10, 30);
   g.drawString("Ascent: " + metrics.getAscent(), 10, 45);
   g.drawString("Descent: " + metrics.getDescent(), 10, 60);
   g.drawString("Height: " + metrics.getHeight(), 10, 75);
   g.drawString("Leading: " + metrics.getLeading(), 10, 90);

   Font font = new Font("Serif", Font.ITALIC, 14);
   metrics = g.getFontMetrics(font);
   g.setFont(font);
   g.drawString("Current font: " + font, 10, 120);
   g.drawString("Ascent: " + metrics.getAscent(), 10, 135);
   g.drawString("Descent: " + metrics.getDescent(), 10, 150);
   g.drawString("Height: " + metrics.getHeight(), 10, 165);
   g.drawString("Leading: " + metrics.getLeading(), 10, 180);
  }
}
```

5

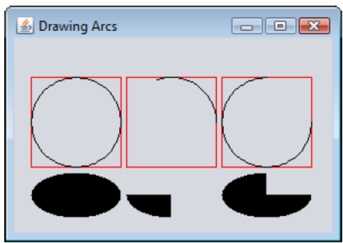
```
import java.awt.Color;
import javax.swing.JFrame;

public class LinesRectsOvals {
  public static void main(String[] args)  {
    JFrame frame =
     new JFrame("Drawing lines, rectangles and ovals");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   LinesRectsOvalsJPanel linesRectsOvalsJPanel =
    new LinesRectsOvalsJPanel();
   linesRectsOvalsJPanel.setBackground(Color.WHITE);
   frame.add(linesRectsOvalsJPanel);
   frame.setSize(400, 210);
   frame.setVisible(true);
 }
}
```



```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;

public class LinesRectsOvalsJPanel extends JPanel {

 @Override  public void paintComponent(Graphics g)
 {
   super.paintComponent(g);
   this.setBackground(Color.WHITE);

   g.setColor(Color.RED);
   g.drawLine(5, 30, 380, 30);

   g.setColor(Color.BLUE);
   g.drawRect(5, 40, 90, 55);
   g.fillRect(100, 40, 90, 55);

   g.setColor(Color.BLACK);
   g.fillRoundRect(195, 40, 90, 55, 50, 50);
   g.drawRoundRect(290, 40, 90, 55, 20, 20);

   g.setColor(Color.GREEN);
   g.draw3DRect(5, 100, 90, 55, true);
   g.fill3DRect(100, 100, 90, 55, false);

   g.setColor(Color.MAGENTA);
   g.drawOval(195, 100, 90, 55);
   g.fillOval(290, 100, 90, 55);
 }
}
```

6

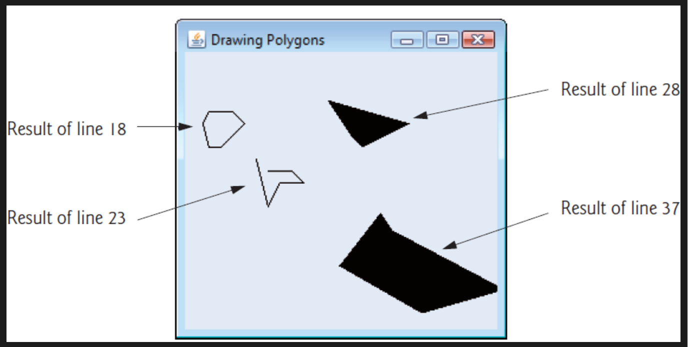
```
import javax.swing.JFrame;

public class DrawArcs {

 public static void main(String[] args) {
   JFrame frame = new JFrame("Drawing Arcs");
   frame.setDefaultCloseOperation(
           JFrame.EXIT_ON_CLOSE);

   ArcsJPanel arcsJPanel = new ArcsJPanel();
   frame.add(arcsJPanel);
   frame.setSize(300, 210);
   frame.setVisible(true);
 }
}
```



```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;

public class ArcsJPanel extends JPanel {

 @Override  public void paintComponent(Graphics g)  {
   super.paintComponent(g);

   g.setColor(Color.RED);
   g.drawRect(15, 35, 80, 80);
   g.setColor(Color.BLACK);
   g.drawArc(15, 35, 80, 80, 0, 360);

   g.setColor(Color.RED);
   g.drawRect(100, 35, 80, 80);
   g.setColor(Color.BLACK);
   g.drawArc(100, 35, 80, 80, 0, 110);

   g.setColor(Color.RED);
   g.drawRect(185, 35, 80, 80);
   g.setColor(Color.BLACK);
   g.drawArc(185, 35, 80, 80, 0, -270);

   g.fillArc(15, 120, 80, 40, 0, 360);
   g.fillArc(100, 120, 80, 40, 270, -90);
   g.fillArc(185, 120, 80, 40, 0, -270);
 }
}
```

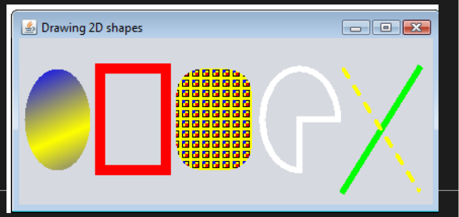
7

```
import javax.swing.JFrame;

public class DrawPolygons {
 public static void main(String[] args) {
   JFrame frame = new JFrame("Drawing Polygons");
   frame.setDefaultCloseOperation(
          Frame.EXIT_ON_CLOSE);

   PolygonsJPanel polygonsJPanel =
         new PolygonsJPanel();
   frame.add(polygonsJPanel);
   frame.setSize(280, 270);
   frame.setVisible(true);
 }
}

import java.awt.Graphics;
import java.awt.Polygon;
import javax.swing.JPanel;

public class PolygonsJPanel extends JPanel {

 @Override  public void paintComponent(Graphics g)  {
   super.paintComponent(g);

   int[] xValues = {20, 40, 50, 30, 20, 15};
   int[] yValues = {50, 50, 60, 80, 80, 60};
   Polygon polygon1 =
      new Polygon(xValues, yValues, 6);
   g.drawPolygon(polygon1);

   int[] xValues2 =
        {70, 90, 100, 80, 70, 65, 60};
   int[] yValues2 =
        {100, 100, 110, 110, 130, 110, 90};
   g.drawPolyline(xValues2, yValues2, 7);

   int[] xValues3 = {120, 140, 150, 190};
   int[] yValues3 = {40, 70, 80, 60};
   g.fillPolygon(xValues3, yValues3, 4);

   Polygon polygon2 = new Polygon();
   polygon2.addPoint(165, 135);
   polygon2.addPoint(175, 150);
   polygon2.addPoint(270, 200);
   polygon2.addPoint(200, 220);
   polygon2.addPoint(130, 180);
   g.fillPolygon(polygon2);
 }
}
```



8

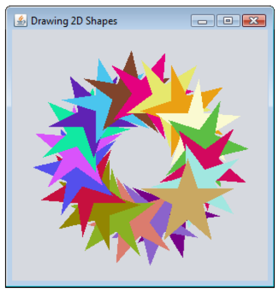
```
import javax.swing.JFrame;

public class Shapes {

 public static void main(String[] args)  {
   JFrame frame = new JFrame("Drawing 2D shapes");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   ShapesJPanel shapesJPanel = new ShapesJPanel();
   frame.add(shapesJPanel);
   frame.setSize(425, 200);
   frame.setVisible(true);
 }
}
```



```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.BasicStroke;
import java.awt.GradientPaint;
import java.awt.TexturePaint;
import java.awt.Rectangle;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Rectangle2D;
import java.awt.geom.RoundRectangle2D;
import java.awt.geom.Arc2D;
import java.awt.geom.Line2D;
import java.awt.image.BufferedImage;
import javax.swing.JPanel;

public class ShapesJPanel extends JPanel {

 @Override  public void paintComponent(Graphics g)  {
   super.paintComponent(g);
   Graphics2D g2d = (Graphics2D) g;

   g2d.setPaint(new GradientPaint(5, 30,
                         Color.BLUE, 35, 100,
                         Color.YELLOW, true));
   g2d.fill(new Ellipse2D.Double(5, 30, 65, 100));

   g2d.setPaint(Color.RED);
   g2d.setStroke(new BasicStroke(10.0f));
   g2d.draw(new Rectangle2D.Double(80, 30, 65, 100));

   BufferedImage buffImage =
     new BufferedImage(10, 10,
                       BufferedImage.TYPE_INT_RGB);

   Graphics2D gg = buffImage.createGraphics();
   gg.setColor(Color.YELLOW);
   gg.fillRect(0, 0, 10, 10);
   gg.setColor(Color.BLACK);
   gg.drawRect(1, 1, 6, 6);
   gg.setColor(Color.BLUE);
   gg.fillRect(1, 1, 3, 3);
   gg.setColor(Color.RED);
   gg.fillRect(4, 4, 3, 3);

   g2d.setPaint(new TexturePaint(buffImage,
                  new Rectangle(10, 10)));
   g2d.fill( new RoundRectangle2D.Double(155, 30,
                 75, 100, 50, 50));

   g2d.setPaint(Color.WHITE);
   g2d.setStroke(new BasicStroke(6.0f));
   g2d.draw(
    new Arc2D.Double(240, 30, 75, 100, 0, 270, Arc2D.PIE));
   g2d.setPaint(Color.GREEN);
   g2d.draw(new Line2D.Double(395, 30, 320, 150));

   float[] dashes = {10};
   g2d.setPaint(Color.YELLOW);
   g2d.setStroke(new BasicStroke(4, BasicStroke.CAP_ROUND,
    BasicStroke.JOIN_ROUND, 10, dashes, 0));
   g2d.draw(new Line2D.Double(320, 30, 395, 150));
 }
}
```

9

```
import java.awt.Color;
import javax.swing.JFrame;

public class Shapes2
{

 public static void main(String[] args)
 {

   JFrame frame = new JFrame("Drawing 2D Shapes");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   Shapes2JPanel shapes2JPanel = new Shapes2JPanel();
   frame.add(shapes2JPanel);
   frame.setBackground(Color.WHITE);
   frame.setSize(315, 330);
   frame.setVisible(true);
 }
}
```



```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.GeneralPath;
import java.security.SecureRandom;
import javax.swing.JPanel;
```

```
public class Shapes2JPanel extends JPanel {
 @Override  public void paintComponent(Graphics g)  {
   super.paintComponent(g);
   SecureRandom random = new SecureRandom();

   int[] xPoints =
     {55, 67, 109, 73, 83, 55, 27, 37, 1, 43};
   int[] yPoints =
     {0, 36, 36, 54, 96, 72, 96, 54, 36, 36};

   Graphics2D g2d = (Graphics2D) g;
   GeneralPath star = new GeneralPath();

   star.moveTo(xPoints[0], yPoints[0]);

   for (int count = 1; count < xPoints.length; count++)
    star.lineTo(xPoints[count], yPoints[count]);

   star.closePath();
   g2d.translate(150, 150);

   for (int count = 1; count <= 20; count++) {
    g2d.rotate(Math.PI / 10.0);
    g2d.setColor(
      new Color(random.nextInt(256),
                random.nextInt(256),
               random.nextInt(256)));
    g2d.fill(star);
   }
 }
}
```