

Assignment Purpose:

The purpose of this program is to demonstrate knowledge of using the main methods of the **Graphics** class in the **java.awt** package.

Write a program, which displays twelve different features of the **Graphics** class. Your program will show a graphics grid of three rows by four columns. Each grid location will display a brief text message and a display of various graphics features. The **main** method, method **grid** and stubs for the other methods are provided.

This lab assignment is not complex, logically speaking, but that does not make it easy. Each one of the basic AWT Graphics command that are demonstrated in the first unit program examples needs to be used. Your biggest challenge will to determine the proper coordinate value to use. You will create twelve separate graphics displays in twelve different cell locations. Close attention to coordinate values is the main requirement. An attempt at advanced graphics is impossible without a very clear understanding of coordinate point values on the graphics window.

Provided Student Program for Unit01vST

```
// Unit01vST.java
// Student starting version of the Unit01 lab assignment.

import java.awt.*;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.util.Random;

public class Unit01vST extends Frame implements WindowListener
{
    public Unit01vST() {
        this.addWindowListener(this);
        repaint();
    }
    public void paint(Graphics g)
    {
        grid(g);
        lines(g);
        openRectangles(g);
        filledRectangles(g);
        openOvals(g);
        filledOvals(g);
        openArcs(g);
        filledArcs(g);
        settingColors(g);
        openPolygons(g);
        filledPolygons(g);
        polyLines(g);
        randomPixels(g);
    }
    public void grid(Graphics g)
```

```

{
    g.drawRect(50,50,800,600);
    for (int x = 250; x < 850; x+= 200)
        g.drawLine(x,50,x,650);
    for (int y = 250; y < 650; y+= 200)
        g.drawLine(50,y,850,y);
}

public void lines(Graphics g)
{
}

public void openRectangles(Graphics g)
{
}

public void filledRectangles(Graphics g)
{
}

public void openOvals(Graphics g)
{
}

public void filledOvals(Graphics g)
{
}

public void openArcs(Graphics g)
{
}

public void filledArcs(Graphics g)
{
}

public void settingColors(Graphics g)
{
}

public void openPolygons(Graphics g)
{
}

public void filledPolygons(Graphics g)
{
}

public void polyLines(Graphics g)
{
}

public void randomPixels(Graphics g)
{
}
}

```

80-Point, 90-Point and 100-Point Versions

This uses many of the common **Graphics** methods. For the **80-point** version you will need to complete the first four grid locations. This will demonstrate **Lines**, **Open Rectangles**, **Closed Rectangles** and **Open Ovals**.

The **90-point** version will add four additional displays for **Filled Ovals**, **Open Arcs**, **Filled Arcs** and **Setting Colors**.

The **100-point** version completes the entire grid with the last row adding displays for **Open Polygons**, **Filled Polygons**, **Poly Lines** and **Random Pixels**.

Methods of the Graphics class used for GFX-Unit01Lab

drawLine Method Class: Graphics

```
drawLine(int x1, int y1, int x2, int y2)
```

Draws a line from coordinate (x1,y1) to coordinate (x2,y2)

drawRect and fillRect Methods Class: Graphics

```
drawRect(int x, int y, int width, int height)
```

Draws a rectangle with top-left corner at coordinate (x,y) using width and height dimensions.

fillRect uses identical parameters, but fills in the rectangle.

drawRoundRect and fillRoundRect Methods

```
drawRoundRect(int x, int y, int width, int height, int cornerW, int cornerH)
```

Draws a rectangle with top-left corner at coordinate (x,y) using width and height dimensions. cornerW and cornerH are the width and height values of the rectangle that dimensions the size of the rounded corner.

fillRoundRect uses the same parameters, but fills the rounded rectangle completely.

drawOval and fillOval Methods Class: Graphics

drawOval(int x, int y, int width, int height)

Draws an oval that is circumscribed by the rectangle with top-left corner at coordinate (x,y) using width and height dimensions.

fillOval uses identical parameters, but fills in the oval.

drawArc and fillArc Methods Class: Graphics

drawArc(int x, int y, int width, int height, int degrees)

Draws an arc that is circumscribed by the rectangle with top-left corner at coordinate (X,Y) using Width and Height dimensions. Start indicates the degree location of the beginning of the arc and Degrees indicates the degrees traveled by the arc. **0 degrees** is at the **3:00 o'clock** position and increases counter clockwise to **360 degrees**.

setColor Method Class: Graphics

Color.constant Class: Color

setColor(Color.constant)

Sets the graphics display color of the following graphics output to the specified constant of the **Color** class. Color constants are combinations of Red, Green and Blue (RGB) values. The following constants are defined for the Color class:

red	255, 0, 0
green	0, 255, 0
blue	0, 0, 255
orange	255, 200, 0
cyan	0, 255, 255
magenta	255, 0, 255
yellow	255, 255, 0
gray	128, 128, 128
lightgray	192, 192, 192
darkgray	64, 64, 64
pink	255, 175, 175
black	0, 0, 0
white	255, 255, 255

drawString Method Class: Graphics

```
drawString(String s, int x, int y)
```

Draws a string s starting at the at coordinate (x,y).

drawPolygon & fill Polygon Methods Class: Graphics

```
drawPolygon(int[ ] xCoord, int[ ] yCoord, int points)
```

Draws a polygon by connecting the coordinates stored in the xCoord array and the yCoord array. The elements of the arrays must be equal to each other and equal to the number of points.

fillPolygon uses the same parameters, but fills the polygon completely.

drawPolyline Method Class: Graphics

```
drawPolyline(int[ ] xCoord, int[ ] yCoord, int points)
```

Draws a sequence of connected lines provided by the coordinates stored in the xCoord array and the yCoord array. The elements of the arrays must be equal to each other and equal to the number of points.

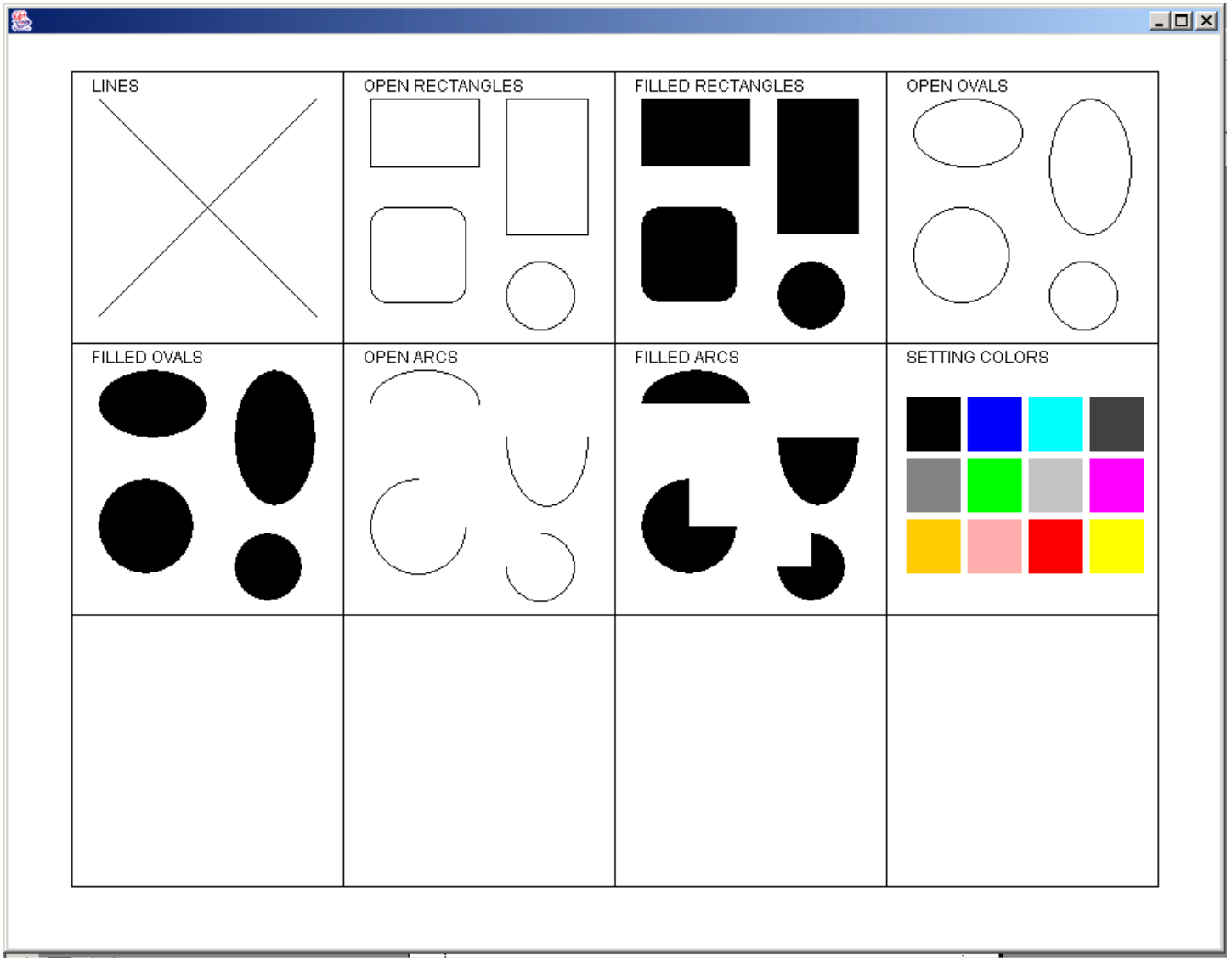
GFX-Unit01Lab Version-80 Point Required Execution

The image shows a Java Swing window with a blue title bar and standard window controls. The main content area is a 3x4 grid. The top row contains four columns of graphics examples, each with a label above them:

- LINES:** Two intersecting diagonal lines forming an 'X' shape.
- OPEN RECTANGLES:** A horizontal rectangle, a vertical rectangle, a rounded rectangle, and a circle.
- FILLED RECTANGLES:** A solid black horizontal rectangle, a solid black vertical rectangle, a solid black rounded rectangle, and a solid black circle.
- OPEN OVALS:** A horizontal open oval, a vertical open oval, a circle, and a smaller circle.

The bottom two rows of the grid are empty, providing space for the user to execute and observe the results of their code.

GFX-Unit01Lab Version-90 Point Required Execution



GFX-Unit01Lab Version-100 Point Required Execution

