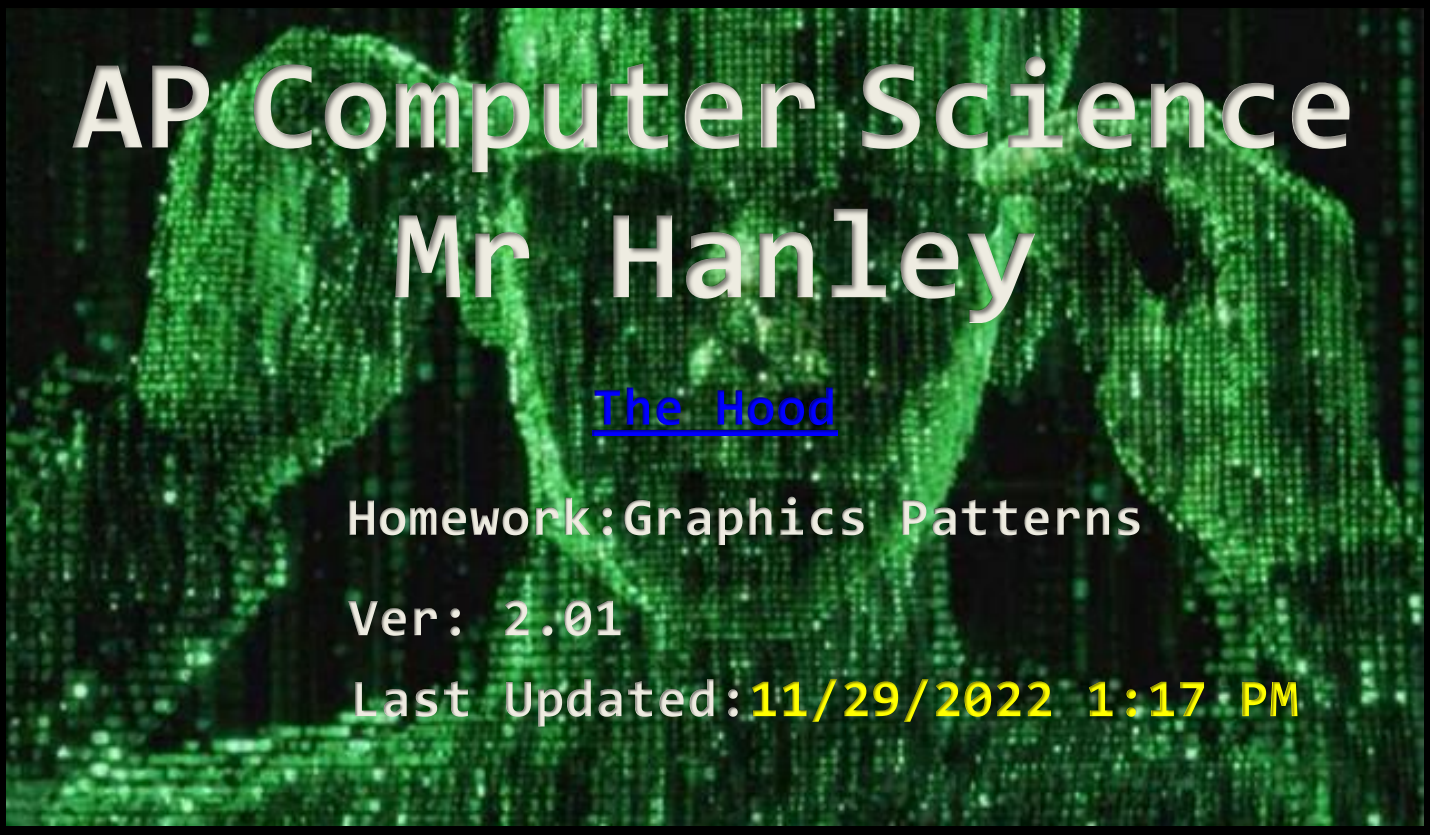


Homework: Graphics Patterns

On the next pages are nine different graphic outputs. Each was drawn using a for-loops. For this lab, you will add methods in the ForLoopPictures class so that your new pictures show up when running the main method in the ForLoopsRunner class.

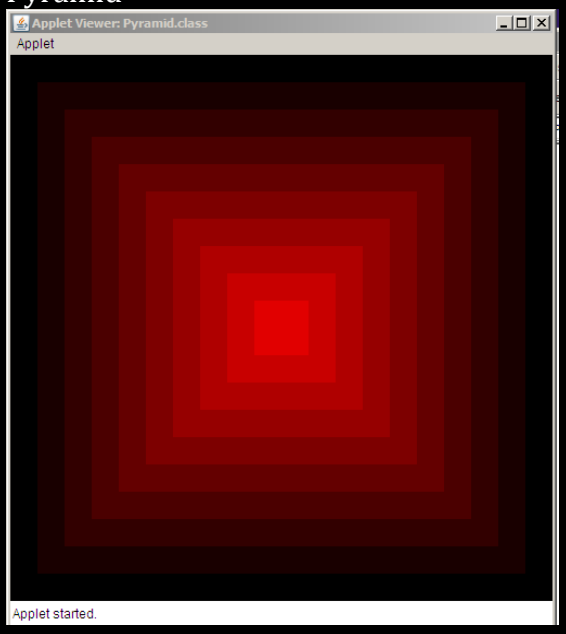
The general outline for the code is as follows:

Every applet should be drawn in a 500 by 500 window.

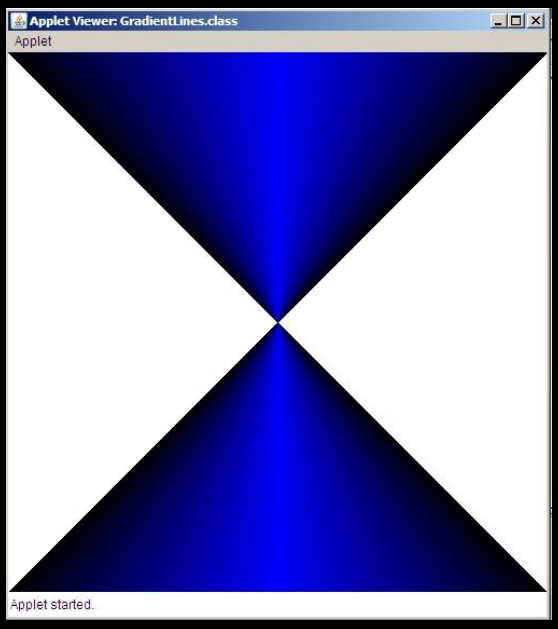
Pyramid consists of 10 squares. The rgb values for the largest square is (0, 0, 0) and the smallest square has rgb values of (250, 0, 0).

GradientLines consists of 500 lines. The first line goes from top left corner to the bottom right corner. For each subsequent line, the top endpoint moves to the right and the bottom endpoint moves to the left. For the first half of the lines, the rgb is (0, 0, blue) where blue equals the x-coordinate of the top of the line. After you hit halfway, the blue component reverses.

Pyramid



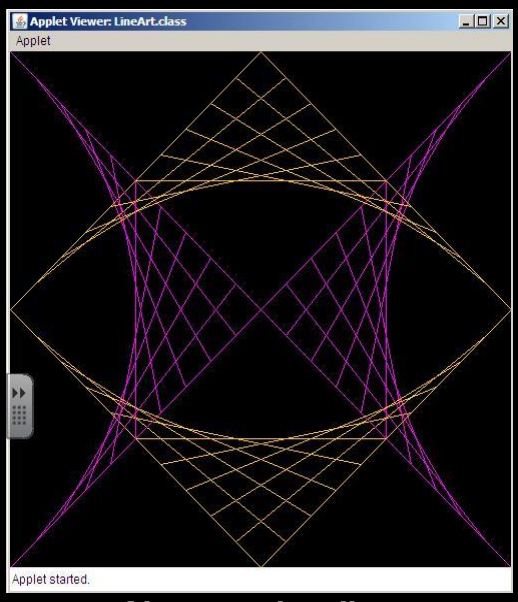
GradientLines



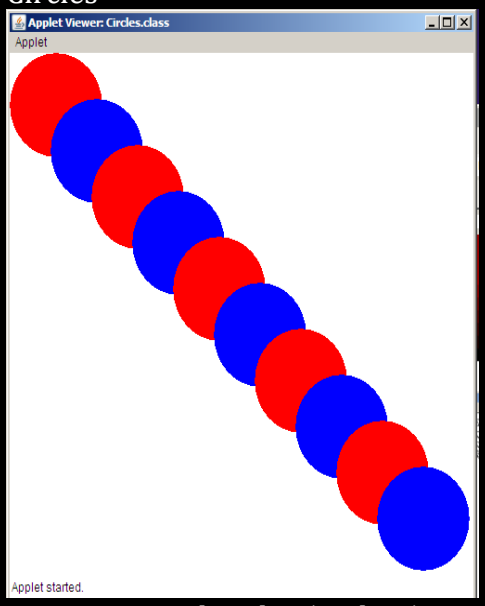
LineArt is made up of segments. Each of the four sections is made up of 10 line segments.

Circles is made up of 10 circles with diameter 100. The colors alternate. You'll need to find a way to alternate between colors.

LineArt



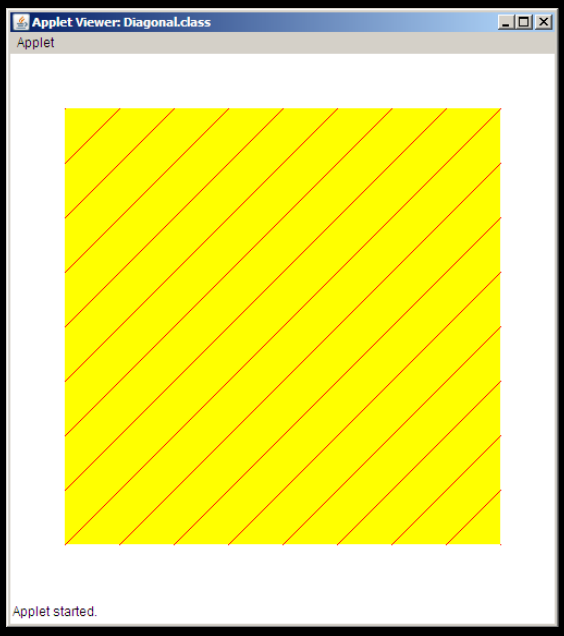
Circles



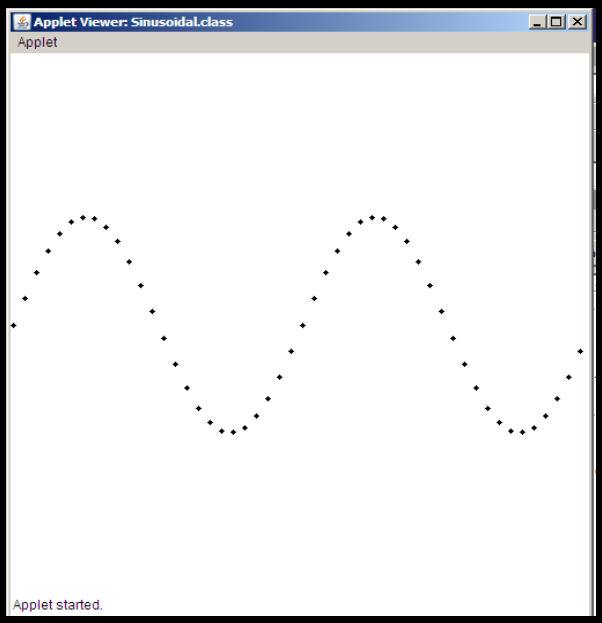
Diagonal has a single yellow square that is 400 by 400 centered in the 500 by 500 applet window. It features parallel red lines that have endpoints every 50 pixels along the border of the square.

Sinusoidal draws small circles (diameter = 5 pixels) along a sine curve. Use Math.sin(x) where x is in radians. Don't forget that on a typical xy axis, y incr as you go up, but here y incre as you go down.

Diagonal

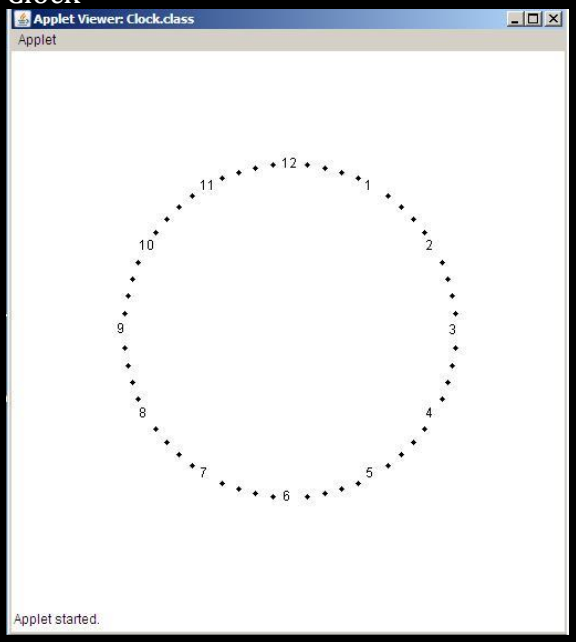


Sinusoidal

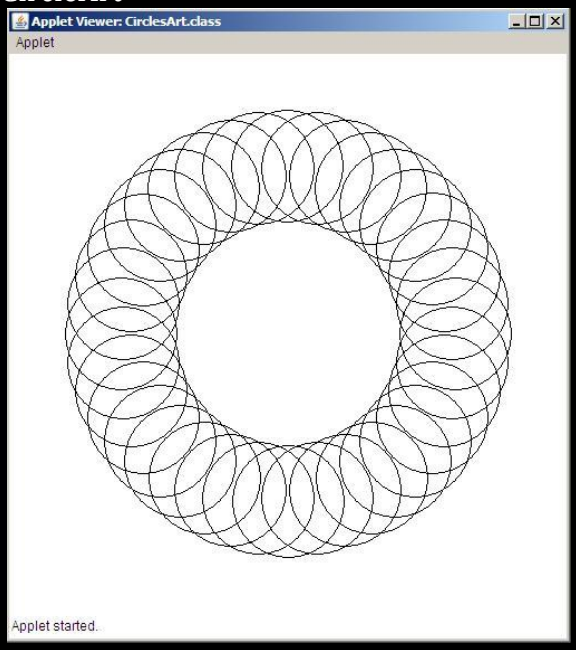


Clock draws a small 5 pixel circle or the numbers 1 through 12 like you would have on a clock face. The circle of the clock has a radius of 150 pixels. One way to do this is by using parametric equations for a circle x = h + r cos(a) and y = k + r sin(a). Your input for the sine or cosine of the angle will need to either be in radians or you can use Math.toRadians(int angle) to convert degrees to radians.

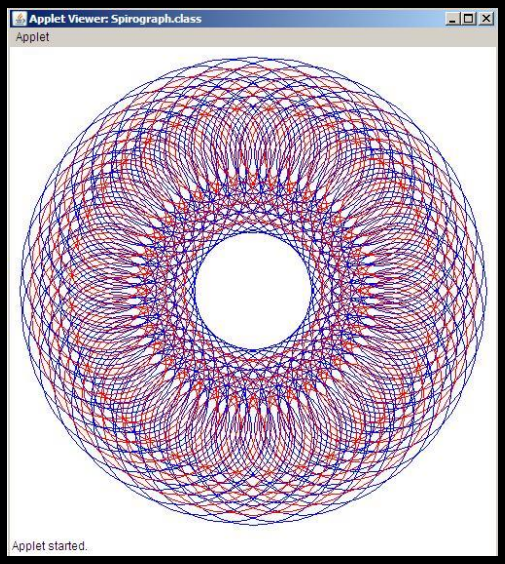
Clock



CircleArt



Spirograph



```
/**
 * ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 * S-h-e-n-e-n-d-e-h-o-w-a--H-i-g-h--S-c-h-o-o-l--T-e-c-h-n-o-l-o-g-y--D-e-p-t
 * ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 *      FILE:       GraphicsJPanel.java
 *      DATE:       Nov 5, 2019
 *      AUTHOR:     Someone from internet, I stole it!!!!!
 *      VERSION:    1.0
 *      PURPOSE:    Displays Graphical Patterns
 *
 * ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 *
 * m-r-h-a-n-l-e-y-c-.c-o-m~~~~~~~~~~t-e-a-m-2-0-.-c-o-m~~~~~~~~~~~~~~~~~~~~~
 */

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

public class GraphicsJPanel extends JPanel implements ActionListener {

    private int counter;        //# of Clicks
    private int pixNum;         //which picture to display

    public GraphicsJPanel(JButton button) {
        counter = 0;
        pixNum = 0;
        button.addActionListener(this);
    }

    public void actionPerformed(ActionEvent ae) {
        counter++;
        pixNum = counter % 2;
        repaint();

    }

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

        switch (pixNum) {
            case 0:
                redPyramid(g);
                break;

        }
    }

    public void redPyramid(Graphics g) {
        //Red Pyramid
    }
}
```

```
/**
 * ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 * S-h-e-n-e-n-d-e-h-o-w-a--H-i-g-h--S-c-h-o-o-l--T-e-c-h-n-o-l-o-g-y--D-e-p-t
 * ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 * FILE:        GraphicsRunner.java
 * DATE:        Nov 5, 2018
 * AUTHOR:      mr Hanley
 * PURPOSE:     Create 9 different Images to practice for loops and Graphics
 *
 * ~~~~~~~~~~~~~~~m-r-h-a-n-l-e-y.c-o-m~~~~~~~~~~t-e-a-m-2-0-.-c-o-m~~~~~~~~~~~
 */

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

public class GraphicsRunner
{
    public static void main(String args[])
    {
        JFrame jf = new JFrame("Graphics Patterns");
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jf.setResizable(false);
        jf.setSize(515, 575);

        Container c = jf.getContentPane();
        JButton nextPix = new JButton("New Picture");

        GraphicsJPanel flp = new GraphicsJPanel (nextPix);
        jf.add(flp);

        jf.add(nextPix, BorderLayout.SOUTH);

        jf.setVisible(true);
    }
}
```

Project 1 Name	**Graphics Pattern HW**
Class 1 Name	GraphicsRunner.java
Class 2 Name	GraphicsJPanel.java

RUBRIC

Red Pyramid	10
Gradient Lines	10
Line Art	10
Circles	10
Diagonal	15
Clock	15
Circle Art	15
Sinusoidal	15
Spirograph	15
TOTAL	115


Artificial_Intelligence
Dynamic_Memory
Waterfall_Model
Nested_Loops
Stacks
Short_Circuit_Evaluation
Insertion_Sort
Binary_Search
Inheritance
Files
Linear_nlogn
Extreme_Programming
Client_Server
Hong_Kong_Fooey
Video_Games
String
Recursion
Search
File_Processing
Arrays
Selection_Sort
Big_O_Notation
2D_Arrays
Hangman
ArrayList