```
/* Chapter No. NA. - Project No.  NA.
    File Name:          Problem11.java
    Programmer:         Andrew Caldwell
    Date Last Modified: Jan. 24, 2014

    Problem Statement:
    Draw a circle within a square.

    Overall Plan
    * Draw square
    * Draw Circle

    Classes needed and Purpose
    main class - Problem11
    Graphics - context
*/

import javax.swing.JApplet;
import java.awt.Graphics;
import java.awt.Color;

public class Problem11 extends JApplet {
    public static final int START_X = 100;
    public static final int START_Y = 100;
    public static final int RING_DIAMETER = 100;
    public static final int RING_RADIUS = RING_DIAMETER/2;

    public static final int SQUARE_EDGE = RING_DIAMETER;

    public static final Color[] COLORS_ARRAY = {Color.PINK, Color.BLUE, Color.YELLOW,
 Color.BLACK, Color.GREEN, Color.RED};

    public void paint(Graphics context) {
        context.setColor(Color.PINK);
        context.drawRect(START_X,START_Y,SQUARE_EDGE,SQUARE_EDGE);
        context.setColor(Color.RED);
        context.drawOval(START_X,START_Y,RING_DIAMETER,RING_DIAMETER);
    }
}
```

```
/* Chapter No. NA. - Project No.  NA.
    File Name:          Problem12.java
    Programmer:         Andrew Caldwell
    Date Last Modified: Jan. 24, 2014

    Problem Statement:
    Draw a crescent moon, outline and inside in different colors.

    Overall Plan
    * Draw square
    * Draw Circle

    Classes needed and Purpose
    main class - Problem12
    Graphics - context
*/

import javax.swing.JApplet;
import java.awt.Graphics;
import java.awt.Color;

public class Problem12 extends JApplet {
    public static final int START_X = 100;
    public static final int START_Y = 100;
    public static final int RING_DIAMETER = 100;

    public static final int OVAL_DIAMETER_X = RING_DIAMETER;
    public static final int OVAL_DIAMETER_Y = (RING_DIAMETER * 3) / 4;

    public static final Color[] COLORS_ARRAY = {Color.PINK, Color.BLUE, Color.YELLOW,
 Color.BLACK, Color.GREEN, Color.RED};
    public void init() {
        setBackground(Color.BLUE);
    }
    public void paint(Graphics context) {
        // int box_width = 30;
        // for (int i = 0, x = START_X, y = START_Y;i < COLORS_ARRAY.length;i++) {
        //  context.setColor(COLORS_ARRAY[i]);
        //  context.fillRect(x,y,box_width,100);
        //  x+= box_width;
        // }

        context.setColor(Color.RED);
        context.fillRect(0,0,1000,300);

        context.setColor(Color.PINK);
        context.fillOval(START_X,START_Y,100,100);
        context.setColor(Color.RED);
        context.fillOval(START_X,START_Y,100,75);
    }
}
```

```
/* Chapter No. NA. - Project No.  NA.
    File Name:          Problem14.java
    Programmer:         Andrew Caldwell
    Date Last Modified: Jan. 24, 2014

    Problem Statement:
    Draw a crescent moon, outline and inside in different colors.

    Overall Plan
    * Draw bull's eye pattern with alternating blue and green.

    Classes needed and Purpose
    main class - Problem14
    Graphics - context
*/

import javax.swing.JApplet;
import java.awt.Graphics;
import java.awt.Color;

public class Problem14 extends JApplet {
    public static final int START_X = 100;
    public static final int START_Y = 100;
    public static final int RING_DIAMETER = 100;
    public static final int RING_INSET = RING_DIAMETER/10;

    public static final Color[] COLORS_ARRAY = {Color.PINK, Color.BLUE, Color.YELLOW,
 Color.BLACK, Color.GREEN, Color.RED};
```

```
    public void paint(Graphics context) {

        for (int i = 0; i < 5;i++) {
            // Alternate colors between blue and green
            if (context.getColor() != Color.BLUE) {
                context.setColor(Color.BLUE);
            } else  {
                context.setColor(Color.GREEN);
            }


            context.fillOval(START_X+ RING_INSET*i/2, START_Y + RING_INSET*i/2, RING_
DIAMETER - RING_INSET*i, RING_DIAMETER - RING_INSET*i);
        }

    }
}
```

```
/* Chapter No. NA. - Project No.  NA.
    File Name:          Problem16.java
    Programmer:         Andrew Caldwell
    Date Last Modified: Jan. 24, 2014

    Problem Statement:
    Use you imagination to draw things

    Overall Plan:
    loop i from  0 to 10
        Set context from color array
        if i = 5
            set x to 0
            set y to diameter
        draw face
        incement happiness
        increment x

    Classes needed and Purpose
    main class - Problem16
    Graphics - context
    AJTTDrawing - drawing faces with happiness
```

```
*/

import javax.swing.JApplet;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Color;
import java.awt.Point;

public class Problem16 extends JApplet {
    public static final int FACE_DIAMETER = 100;
    public static final int FACE_RADIUS = FACE_DIAMETER / 2;
    public static final int HAPPINESS_OFFSET = 10;
    public static final Color[] COLORS = {Color.BLACK, Color.BLUE, Color.CYAN, Color.
GREEN, Color.MAGENTA, Color.ORANGE,Color.PINK, Color.RED, Color.WHITE, Color.YELLOW};
    public static final int START_X = 100;
    public static final int START_Y = 100;

    public void paint(Graphics context) {
        Graphics2D context2D = (Graphics2D)context;
        int happiness = 0;

        for (int i = 0, x = START_X, y = START_Y;i < 10;i++) {
            context.setColor(COLORS[i]);
            if (i == 5) {
                // change path midway
                x = START_X;
                y = START_Y + FACE_DIAMETER;
            }
            AJTTDrawing.drawFace(context2D,new Point(x,y),happiness,FACE_DIAMETER);
            happiness += HAPPINESS_OFFSET;
            x += FACE_DIAMETER;
        }
    }
}
```

```
import javax.swing.JApplet;
import java.awt.*;
import java.awt.Point;
import java.awt.Graphics2D;
import java.awt.geom.Path2D.Double;
import java.awt.BasicStroke;

public class AJTTDrawing {
    // happiness is an integer representing the smile on the face, from 0 - 100
    // 0 happiness is absolue unhappy and 100 happiness is absolutly happy
    public static void drawFace(Graphics2D context, Point faceCoords, int happiness,
int DIAMETER) {
        // Check for bad happiness value
        if (happiness > 100)
            happiness = 100;
        else if (happiness < 0)
            happiness = 0;


        // Get int face coords
        int FACE_X = (int)faceCoords.getX();
        int FACE_Y = (int)faceCoords.getY();

        // draw face outline
        context.drawOval(FACE_X,FACE_Y,DIAMETER,DIAMETER);

        // draw eyes
        int EYE_LEVEL_Y = FACE_Y + DIAMETER / 4;
        int EYE_OFFSET_X = DIAMETER / 8;
        int RADIUS = DIAMETER / 2;
        int EYE_DIAMETER = DIAMETER / 8;

        int LEFT_EYE_X = FACE_X + RADIUS - EYE_OFFSET_X - EYE_DIAMETER;
        int RIGHT_EYE_X = FACE_X + RADIUS + EYE_OFFSET_X;

        context.drawOval(LEFT_EYE_X,EYE_LEVEL_Y,EYE_DIAMETER,EYE_DIAMETER);
        context.drawOval(RIGHT_EYE_X,EYE_LEVEL_Y,EYE_DIAMETER,EYE_DIAMETER);

        // Draw Smile
        drawSmile(context,faceCoords,happiness,DIAMETER);
    }
    private static void drawSmile(Graphics2D context, Point faceCoords, int happiness
, int DIAMETER) {
        int smilePolyLineSize = 5;
        int[] smilePolyLineX = new int[smilePolyLineSize];
        int[] smilePolyLineY = new int[smilePolyLineSize];

        int radius = DIAMETER / 2;
        int eigth = DIAMETER / 8;
        int FORTH_DIAMETER = DIAMETER / 4;
        int fiveEighthsDown = radius + eigth;



        // Draw Smile
        // context.drawPolyline(smilePolyLineX, smilePolyLineY, smilePolyLineSize);
        java.awt.geom.Path2D.Double arc = new java.awt.geom.Path2D.Double();
        arc.moveTo(faceCoords.getX() + FORTH_DIAMETER,faceCoords.getY() + radius + ei
gth);

        // Happiness level
        boolean isHappy = (happiness >= 50)?true:false;
        double HAPPINESS_OFFSET_PERCENTAGE = Math.abs(((double)happiness-50)/100.0);
        double HAPPINESS_OFFSET = HAPPINESS_OFFSET_PERCENTAGE * FORTH_DIAMETER;



        // Initial Values
        double SMILE_START_X = faceCoords.getX() + FORTH_DIAMETER;
        double SMILE_START_Y = faceCoords.getY() + radius + eigth;
        double SMILE_MIDDLE_X = faceCoords.getX() + radius;
        double SMILE_MIDDLE_Y = SMILE_START_Y;
        double SMILE_END_X = faceCoords.getX() + radius + FORTH_DIAMETER;
        double SMILE_END_Y = SMILE_START_Y;

        SMILE_MIDDLE_Y += (isHappy)?HAPPINESS_OFFSET:-1*HAPPINESS_OFFSET;

        arc.curveTo(SMILE_START_X,SMILE_START_Y,SMILE_MIDDLE_X,SMILE_MIDDLE_Y,SMILE_E
ND_X, SMILE_END_Y);

        BasicStroke s = new BasicStroke(3.0f);
        context.draw(s.createStrokedShape(arc));
    }
}
```