Steve Cooper
CS 106A

Handout #15A
January 19, 2011

Solutions for Section #2

Portions of this handout by Eric Roberts

1. The Fibonacci sequence

```
/*
 * File: Fibonacci.java
 * --------------------
 * This program lists the terms in the Fibonacci sequence up to
 * a constant MAX_TERM_VALUE, which is the largest Fibonacci term
 * the program will display.
 */

import acm.program.*;

public class Fibonacci extends ConsoleProgram {

   public void run() {
      println("This program lists the Fibonacci sequence.");
      int t1 = 0;
      int t2 = 1;
      while (t1 <= MAX_TERM_VALUE) {
         println(t1);
         int t3 = t1 + t2;
         t1 = t2;
         t2 = t3;
      }
   }

/* Defines the largest term to be displayed */
   private static final int MAX_TERM_VALUE = 10000;

}
```

2. Birthday Cake

```
/*
 * File: BirthdatCake.java
 * -----------------------
 * This class creates a birthday cake card. It uses constants for the
 * size of each of the layers of the cake.
 */


import java.awt.*;        // for colors
import acm.graphics.*;
import acm.program.*;


public class BirthdayCake extends GraphicsProgram {

   // constants for the layers of the cake
   private static final int INNER_LAYER_HEIGHT = 50;
   private static final int OUTER_LAYER_HEIGHT = 100;
   private static final int CAKE_WIDTH = 200;


   public void run() {

       // determine the upper left-hand coordinates for each
       // of the layers. Note that all use the same x-coordinate
       int xcoord = getWidth()/2 - CAKE_WIDTH/2;

       int ycoordTop = getHeight()/2 -
                INNER_LAYER_HEIGHT/2 - OUTER_LAYER_HEIGHT;

       int ycoordMid = getHeight()/2 - INNER_LAYER_HEIGHT/2;

       int ycoordLow = getHeight()/2 + INNER_LAYER_HEIGHT/2;

       // make the upper layer
       addLayer (xcoord, ycoordTop, CAKE_WIDTH,
                 OUTER_LAYER_HEIGHT, Color.RED);

       // make the middle layer
       addLayer (xcoord, ycoordMid, CAKE_WIDTH,
                 INNER_LAYER_HEIGHT, Color.YELLOW);

       // make the lower layer
       addLayer (xcoord, ycoordLow, CAKE_WIDTH,
                 OUTER_LAYER_HEIGHT, Color.RED);

    }

   // adds a layer based on the coordinates, color, and size
   // passed
   private void addLayer (int x, int y, int width,
                          int height, Color color) {

       GRect layer = new GRect (x,y, width, height);
       layer.setColor (color);
```

```
        layer.setFilled (true);
        add (layer);
    }

}
```

3. Drawing a robot face

```
/* File: RobotFace.java                    */
/* This program draws a robot face.        */

import acm.graphics.*;
import acm.program.*;
import java.awt.*;

public class RobotFace extends GraphicsProgram {

/* Parameters for the drawing */
   private static final int HEAD_WIDTH = 100;
   private static final int HEAD_HEIGHT = 150;
   private static final int EYE_RADIUS = 10;
   private static final int MOUTH_WIDTH = 60;
   private static final int MOUTH_HEIGHT = 20;

   public void run() {
      addFace(getWidth() / 2, getHeight() / 2);
   }

/* Adds the entire face centered at (cx, cy) */
   private void addFace(double cx, double cy) {
      addHead(cx, cy);
      addEye(cx - HEAD_WIDTH / 4, cy - HEAD_HEIGHT / 4);
      addEye(cx + HEAD_WIDTH / 4, cy - HEAD_HEIGHT / 4);
      addMouth(cx, cy + HEAD_HEIGHT / 4);
   }

/* Adds the head centered at (cx, cy) */
   private void addHead(double cx, double cy) {
      double x = cx - HEAD_WIDTH / 2;
      double y = cy - HEAD_HEIGHT / 2;
      GRect head = new GRect(x, y, HEAD_WIDTH, HEAD_HEIGHT);
      head.setFilled(true);
      head.setFillColor(Color.GRAY);
      add(head);
   }

/* Adds an eye centered at (cx, cy) */
   private void addEye(double cx, double cy) {
      double x = cx - EYE_RADIUS;
      double y = cy - EYE_RADIUS;
      GOval eye = new GOval(x, y, 2 * EYE_RADIUS, 2 * EYE_RADIUS);
      eye.setFilled(true);
      eye.setColor(Color.YELLOW);
      add(eye);
   }

/* Adds a mouth centered at (cx, cy) */
   private void addMouth(double cx, double cy) {
      double x = cx - MOUTH_WIDTH / 2;
      double y = cy - MOUTH_HEIGHT / 2;
      GRect mouth = new GRect(x, y, MOUTH_WIDTH, MOUTH_HEIGHT);
      mouth.setFilled(true);
      mouth.setColor(Color.WHITE);
      add(mouth);
   }
}
```