

Solutions for Section #2

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 {

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

    /* Runs the program */
    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;
        }
    }
}
```

2. Centering text

```
/*
 * File: CenteredText.java
 * -----
 * This programs displays a message centered in the graphics window.
 */

import acm.graphics.*;
import acm.program.*;

public class CenteredText extends GraphicsProgram {

    public void run() {
        GLabel label = new GLabel("CS106A rocks my socks!");
        label.setFont("SansSerif-28");
        double x = (getWidth() - label.getWidth()) / 2;
        double y = (getHeight() + label.getAscent()) / 2;
        label.setLocation(x, y);
        add(label);
    }
}
```

3. Drawing a robot face

```

/*
 * File: RobotFace.java
 * -----
 * This program draws a robot face for Section #2.
 */

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.setFill(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);
    }
}

```