Eric Roberts
CS 106A

Handout #37A
February 5, 2010

Solutions to Practice Midterm #2

Problem 1: Karel the Robot (10 points)

```
/*
 * File: KarelCare
 * ---------------
 * Karel looks through the hospital ward for patients with
 * temperatures over 100 and paints the square under the
 * temperature red so that doctors can treat the patient.
 */

import stanford.karel.*;

public class KarelCare extends SuperKarel {

   public void run() {
      while (frontIsClear()) {
         if (beepersPresent()) {
            checkTemperature();
         }
         move();
      }
      if (beepersPresent()) {
         checkTemperature();
      }
   }

/* Flags temperatures greater than 100 */
   private void checkTemperature() {
      for (int i = 0; i < 100; i++) {
         if (beepersPresent()) {
            pickBeeper();
         }
      }
      if (beepersPresent()) {
         paintCorner(RED);
      }
      while (beepersInBag()) {
         putBeeper();
      }
   }

}
```

Problem 2: Simple C expressions, statements, and functions (10 points)

(2a)

`6 / 5 + 5 + 8 % 3 == 7`	`false`
`('6' - '2') + 'A'`	`69` *or* `'E'`
`"E" - "A"`	Error: Can't subtract strings

(2b) `1` (This program calculates the digital root as described in Chapter 4, exercise 7.)

(2c)

Problem 3: Simple Java programs (15 points)

```
/*
 * File: PythagoreanTriples.java
 * -----------------------------
 * This program finds all sets of numbers a, b, and c
 * such that a and b are both less than the constant MAX
 * and
 *        2      2       2
 *      a   +  b    =  c
 */

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

public class PythagoreanTriples extends ConsoleProgram {

   public void run() {
      for (int a = 1; a <= MAX; a++) {
         for (int b = a; b <= MAX; b++) {
            int csq = a * a + b * b;
            int c = GMath.round(Math.sqrt(csq));
            if (c * c == csq) {
               println(a + ", " + b + ", " + c);
            }
         }
      }
   }


/* Maximum value of a or b */
   private static final int MAX = 25;

}
```

Problem 4: Using the graphics and random number libraries (15 points)

```
/*
 * File: SimpleFrogger.java
 * ------------------------
 * This program solves the Frogger problem from the practice midterm.
 */

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

/*
 * This program gets a frog to jump one square in the closest
 * direction to a mouse click.
 */
public class SimpleFrogger extends GraphicsProgram {

   public void run() {
      frog = new GImage("frog.gif");
      fx = (NCOLUMNS / 2 + 0.5) * SQUARE_SIZE;
      fy = (NROWS - 0.5) * SQUARE_SIZE;
      add(frog, fx - frog.getWidth() / 2,
                fy - frog.getHeight() / 2);
      addMouseListeners();
   }

/* Responds to a mouse click */
   public void mouseClicked(MouseEvent e) {
      double mx = e.getX();
      double my = e.getY();
      if (Math.abs(mx - fx) > Math.abs(my - fy)) {
         if (mx > fx) {
            moveFrog(SQUARE_SIZE, 0);
         } else {
            moveFrog(-SQUARE_SIZE, 0);
         }
      } else {
         if (my > fy) {
            moveFrog(0, SQUARE_SIZE);
         } else {
            moveFrog(0, -SQUARE_SIZE);
         }
      }
   }

/* Moves the frog by dx/dy as long as it remains inside the world */
   private void moveFrog(double dx, double dy) {
      if (insideFroggerWorld(fx + dx, fy + dy)) {
         fx += dx;
         fy += dy;
         frog.move(dx, dy);
      }
   }
```

☞

```
/* Returns true if the point (x, y) is inside the frog's world */
   private boolean insideFroggerWorld(double x, double y) {
      return (x >= 0 && x <= NCOLUMNS * SQUARE_SIZE &&
              y >= 0 && y <= NROWS * SQUARE_SIZE);
   }

/* Private constants */
   private static final int SQUARE_SIZE = 75;
   private static final int NROWS = 4;
   private static final int NCOLUMNS = 7;

/* Private instance variables */
   private GImage frog;    /* The image of the frog */
   private double fx;      /* The x-coordinate of the frog's center */
   private double fy;      /* The y-coordinate of the frog's center */

/* Set the graphics window size */
   public static final int APPLICATION_WIDTH = NCOLUMNS * SQUARE_SIZE;
   public static final int APPLICATION_HEIGHT = NROWS * SQUARE_SIZE;

}
```

Problem 5: Using the String class (10 points)

```
/**
 * Counts the occurrences of the word "love"
 * in a string.
 */
   private int countLove(String str) {
      String lowerCaseString = str.toLowerCase();
      int count = 0;
      int start = lowerCaseString.indexOf("love");
      while (start != -1) {
         if (isSeparator(lowerCaseString, start - 1) &&
             isSeparator(lowerCaseString, start + 4)) {
            count++;
         }
         start = lowerCaseString.indexOf("love", start + 4);
      }
      return count;
   }

/**
 * Checks to see if the ith char of str is a separator.
 */
   private boolean isSeparator(String str, int i) {
      if (i < 0 || i >= str.length()) return true;
      return !Character.isLetter(str.charAt(i));
   }
```