

Solutions to Practice Midterm #2

Please remember that the midterm is open-book.
Tuesday, May 9, 3:30–5:30 P.M., 200-030
Tuesday, May 9, 7:00–9:00 P.M., 380-380Y

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 "turns";
import "extensions";

/* Main program */

function KarelCare() {
  while (frontIsClear()) {
    if (beepersPresent()) {
      checkTemperature();
    }
    move();
  }
  if (beepersPresent()) {
    checkTemperature();
  }
}

/* Flags temperatures greater than 100 */

function checkTemperature() {
  repeat (100) {
    if (beepersPresent()) {
      pickBeeper();
    }
  }
  if (beepersPresent()) {
    paintCorner("Red");
  }
  while (beepersInBag()) {
    putBeeper();
  }
}
```

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

(2a)	<code>10 * 9 + 8 * 7 * 6 * 5 + 4 * 3 / 2 / 1</code>	<u>1776</u>
	<code>var x = 7; (x !== 6) (x !== 7)</code>	<u>true</u>
	<code>"E".charCodeAt(0) - "A".charCodeAt(0)</code>	<u>4</u>

(2b) 1 (This program calculates the digital root as described in Chapter 3, exercise 8.)

(2c)

JavaScript Console

```
s1 = Heart
s2 = earth
```

Problem 3: Simple JavaScript programs (15 points)

```
/*
 * File: ConsecutiveHeads.js
 * -----
 * This file defines the function consecutiveHeads, which simulates tossing
 * a coin until the specified number of consecutive heads appears.
 */

import "RandomLib.js";

/*
 * Simulates the process of tossing a coin repeatedly until the number of
 * consecutive heads is equal to the value specified by numberNeeded.
 */

function consecutiveHeads(numberNeeded) {
  var nTosses = 0;
  var nHeads = 0;
  while (nHeads < numberNeeded) {
    if (randomChance()) {
      console.log("Heads");
      nHeads++;
    } else {
      console.log("Tails");
      nHeads = 0;
    }
    nTosses++;
  }
  console.log("It took " + nTosses + " tosses to get " +
    numberNeeded + " consecutive heads.");
}
```

Problem 4: Using graphics and animation (20 points)

```
/*
 * File: Fireworks.js
 * -----
 * This program simulates a firework launch.
 */

import "graphics";
import "RandomLib.js";

/* Constants */

const GWINDOW_WIDTH = 500;    /* pixels */
const GWINDOW_HEIGHT = 300;  /* pixels */
const DELTA_RADIUS = 2;      /* pixels */
const TIME_STEP = 20;        /* milliseconds */
const FLIGHT_TIME = 1200;    /* milliseconds */
const EXPANSION_TIME = 500;  /* milliseconds */

function Fireworks() {
    var gw = GWindow(GWINDOW_WIDTH, GWINDOW_HEIGHT);
    var r = 1;
    var firework = GOval(gw.getWidth() / 2, gw.getHeight(), r, r);
    firework.setColor(randomColor());
    gw.add(firework);
    var targetX = randomReal(0, gw.getWidth());
    var targetY = randomReal(0, gw.getHeight() / 2);
    var nSteps = FLIGHT_TIME / TIME_STEP;
    var dx = (targetX - firework.getX()) / nSteps;
    var dy = (targetY - firework.getY()) / nSteps;
    var t = 0;
    var step = function() {
        if (t > FLIGHT_TIME + EXPANSION_TIME) {
            clearTimeout(timer);
        } else if (t > FLIGHT_TIME) {
            r += DELTA_RADIUS;
            firework.setBounds(firework.getX() - DELTA_RADIUS,
                               firework.getY() - DELTA_RADIUS, 2 * r, 2 * r);
        } else {
            firework.move(dx, dy);
        }
        t += TIME_STEP;
    };
    var timer = setInterval(step, TIME_STEP);
}
```

Problem 5: Strings (15 points)

```
/*
 * File: Spoonerize.js
 * -----
 * This file defines a function that creates spoonerisms.
 */

/*
 * Creates a spoonerism by swapping the leading consonant strings from the
 * first and last words in the phrase.
 */

function spoonerize(phrase) {
    var firstSpace = phrase.indexOf(' ');
    var lastSpace = phrase.lastIndexOf(' ');
    var firstWord = phrase.substring(0, firstSpace);
    var lastWord = phrase.substring(lastSpace + 1);
    var middle = phrase.substring(firstSpace, lastSpace + 1);
    var newFirstWord = getHead(lastWord) + getTail(firstWord);
    var newLastWord = getHead(firstWord) + getTail(lastWord);
    return newFirstWord + middle + newLastWord;
}

/*
 * Returns the head of the string up to but not including the first vowel.
 */

function getHead(word) {
    return word.substring(0, findFirstVowel(word));
}

/*
 * Returns the tail of the string from the first vowel onward.
 */

function getTail(word) {
    return word.substring(findFirstVowel(word));
}

/*
 * Returns the index of the first vowel in the word, or -1 if none.
 */

function findFirstVowel(word) {
    for (var i = 0; i < word.length; i++) {
        if (isEnglishVowel(word.charAt(i))) return i;
    }
    return -1;
}

/*
 * Returns true if the character ch is a vowel (A, E, I, O, or U, in
 * either upper or lower case).
 */

function isEnglishVowel(ch) {
    return ch.length === 1 && "AEIOUaeiou".indexOf(ch) !== -1;
}
```