

Problem 1: Karel the Robot

```
Public class BreakoutKarel extends SuperKarel {

    Public void run() {
        While (beepersInBag()) {
            If (beeperspresent()) {
                pickBeeper();
                bounce();
            }
            While (frontIsBlocked()) {
                bounce();
            }
        }

        private void bounce() {
            turnLeft();
        }

        private void stepDiagonally() {
            move();
            If (leftIsClear() && noBeepersPresent()) {
                turnLeft();
                move();
                turnRight();
            }
        }
    }
}
```

Problem 2:

a)

5.0 / 4 - 4 / 5	1.25
7 < 9 - 5 && 3 % 0 == 3	false
"B" + 8 + 4	"B84"

b)

To care is human!

Problem 3:

```
public class SecoundLargest extends ConsoleProgram {
```

```

private static final int SENTINEL = 0;

public void run() {
    println("this program finds the largest integers in a");
    println("list. Enter values, one per line, using a "
        + SENTINEL + "to");
    println("signal the end of the list.");
    int largest = - 1;
    int secondLargest = -1;
    while (true) {
        int input = readInt("?");
        if (input==SENTINEL) break;
        if (input > largest) {
            secondLargest = largest;
            largest = input;
        } else if (input > secondLargest) {
            secondLargest = input;
        }
    }
    println("The largest value is " + largest);
    println("The second largest is " + secondLargest);

}
}

```

Problem 4

```

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(); }
        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);
        } } }
        private void moveFrog(double dx, double dy) {
        if (insideFroggerWorld(fx + dx, fy + dy)) {
        fx += dx; fy += dy; frog.move(dx, dy);
        private boolean insideFroggerWorld(double x, double y) {

```

```
return (x >= 0 && x <= NCOLUMNS * SQUARE_SIZE && y >= 0 && y <= NROWS *
SQUARE_SIZE); }
```

```
private static final int SQUARE_SIZE = 75;
private static final int NROWS = 4;
private static final int NCOLUMNS = 7;
private GImage frog;
private double fx;
private double fy;
public static final int APPLICATION_WIDTH = NCOLUMNS * SQUARE_SIZE;
public static final int APPLICATION_HEIGHT = NROWS * SQUARE_SIZE;
}
```

Problem 5

```
private String removeDoubledLetters(String str) {
String result = "";
for (int i = 0; i < str.length(); i++) {
char ch = str.charAt(i);
if (i == 0 || ch != str.charAt(i - 1)) {
result += ch; } }
return result;
```