

Solutions for Section #1

Based on a handout by Eric Roberts

```
/*
 * File: UnitedNationsKarel.java
 * -----
 * The UnitedNationsKarel subclass builds houses at corners
 * marked by rubble.
 */

import stanford.karel.*;

public class UnitedNationsKarel extends SuperKarel {

    public void run() {
        while (frontIsClear()) {
            if (beepersPresent()) {
                pickBeeper();
                backup();
                buildHouse();
            }
            if (frontIsClear()) {
                move();
            }
        }
    }

    /**
     * Builds a beeper house on stilts.
     * Precondition: Karel facing East at bottom of left stilt
     * Postcondition: Karel facing East at bottom of right stilt
     */
    private void buildHouse() {
        turnLeft();
        putThreeBeepers();
        move();
        turnRight();
        move();
        turnRight();
        putThreeBeepers();
        turnAround();
        move();
        turnRight();
        move();
        turnRight();
        putThreeBeepers();
        turnLeft();
    }
}
```

```
/*
 * Creates a line of three beepers.
 * Precondition: Karel is in the first square in the line
 * Postcondition: Karel is in the last square in the line
 */
private void putThreeBeepers() {
    for (int i = 0; i < 2; i++) {
        putBeeper();
        move();
    }
    putBeeper();
}

/**
 * Backs up one corner, leaving Karel facing in the same direction.
 * If there is no space behind Karel, it will run into a wall.
 */
private void backup() {
    turnAround();
    move();
    turnAround();
}
```