SF212 Programming Skill Development 2

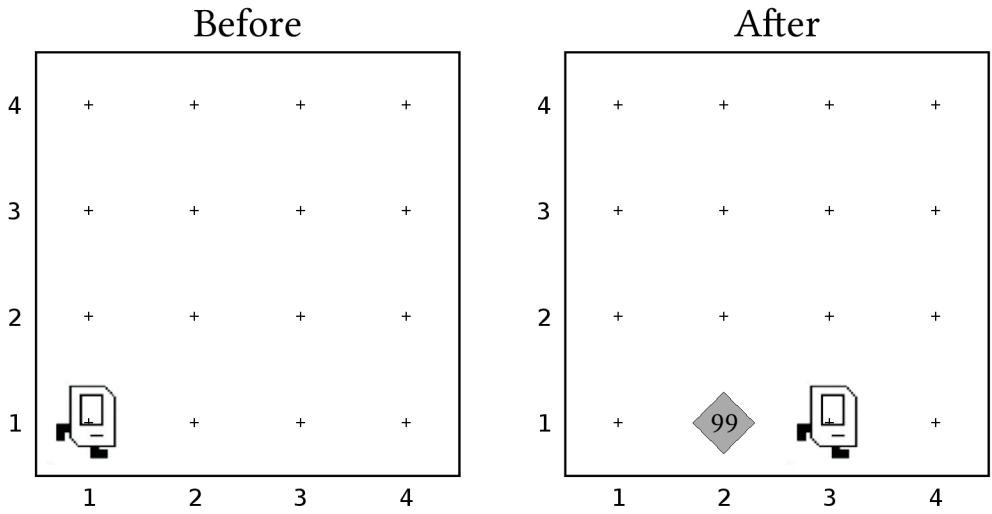
Lecture 2

SuperKarel

```
move();
turnLeft();
turnRight();
turnAround();
putBeeper();
pickBeeper();
paintCorner(color);
```

For Loops

Place 99 beepers?



Place 99 beepers?

```
public class Place99Beepers extends SuperKarel {
    public void run() {
        move();
        repeat(99) {
            putBeeper();
        }
        move();
    }
}
```

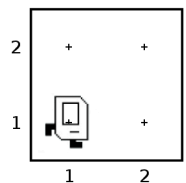
Place 99 beepers?

```
public class Place99Beepers extends SuperKarel {
    public void run() {
        move();
        for(int i = 0; i < 99; i++) {
            putBeeper();
        }
        move();
    }
}
```

This “for loop” repeats the code in its “body” 99 times

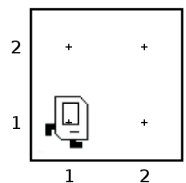
Place Beeper Square

```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



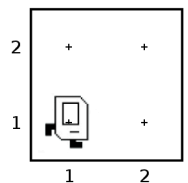
Place Beeper Square

```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Place Beeper Square

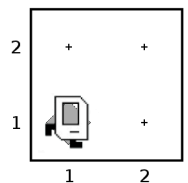
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



First time through the loop

Place Beeper Square

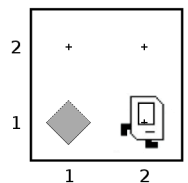
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



First time through the loop

Place Beeper Square

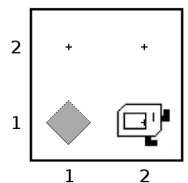
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



First time through the loop

Place Beeper Square

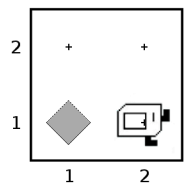
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



First time through the loop

Place Beeper Square

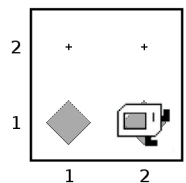
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Second time through the loop

Place Beeper Square

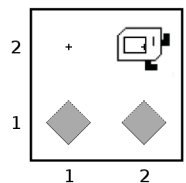
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Second time through the loop

Place Beeper Square

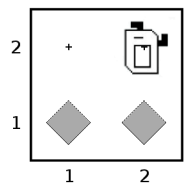
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Second time through the loop

Place Beeper Square

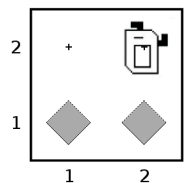
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Second time through the loop

Place Beeper Square

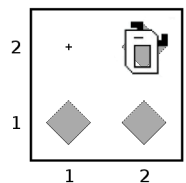
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Third time through the loop

Place Beeper Square

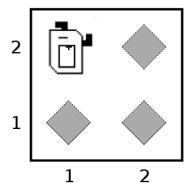
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Third time through the loop

Place Beeper Square

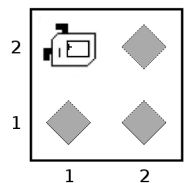
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Third time through the loop

Place Beeper Square

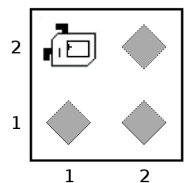
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Third time through the loop

Place Beeper Square

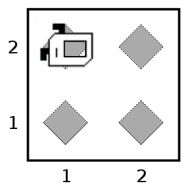
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Fourth time through the loop

Place Beeper Square

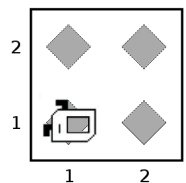
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Fourth time through the loop

Place Beeper Square

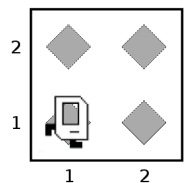
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



Fourth time through the loop

Place Beeper Square

```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
            turnLeft();
        }
    }
}
```



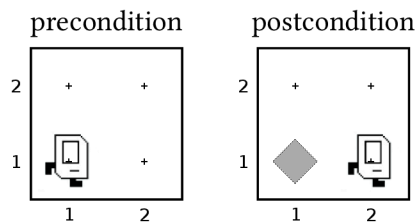
Fourth time through the loop

Why Doesn't This Make a Square?

```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }
    }
}
```

Why Doesn't This Make a Square?

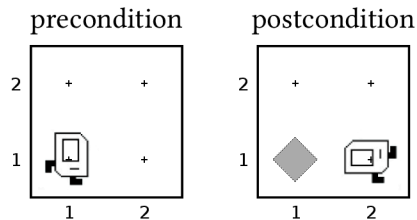
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            // precondition
            putBeeper();
            move();
            // postcondition
        }
    }
}
```



You need the postcondition of a loop to match the precondition

Why Doesn't This Make a Square?

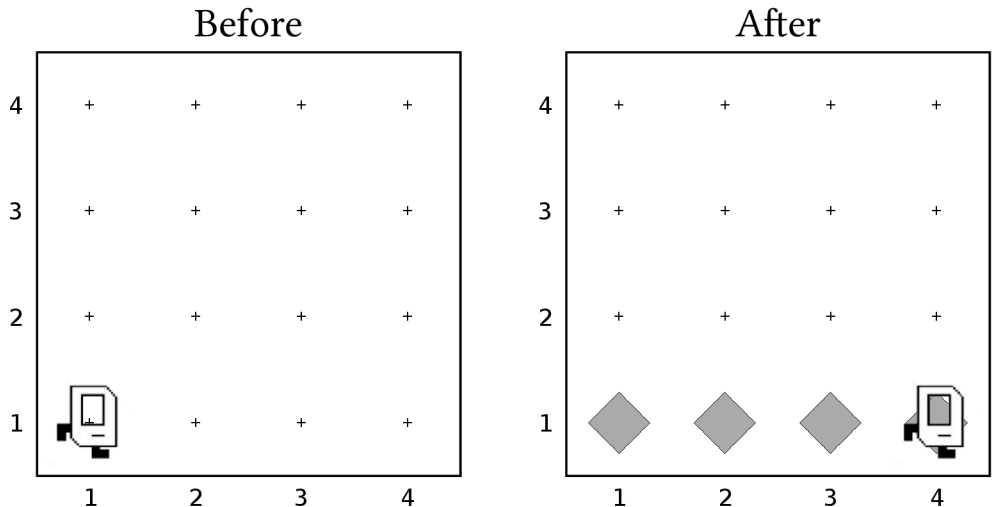
```
public class BeeperSquare extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            // precondition
            putBeeper();
            move();
            turnLeft();
            // postcondition
        }
    }
}
```



You need the postcondition of a loop to match the precondition

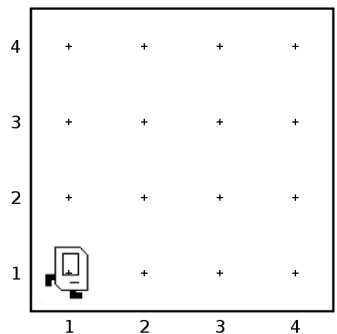
Next Task

Place Beeper Line



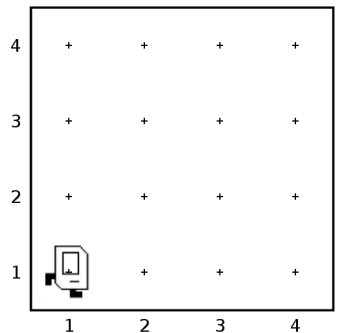
Place Beeper Line

```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



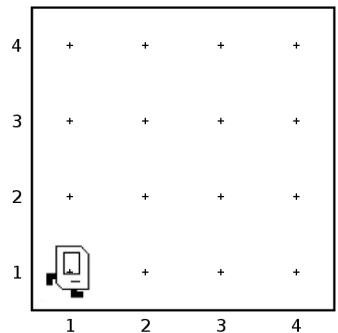
Place Beeper Line

```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Place Beeper Line

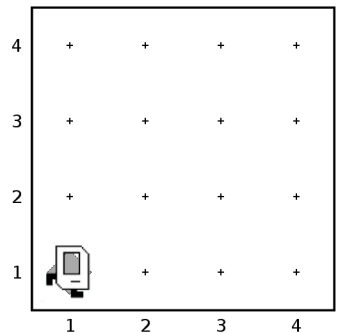
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



First time through the loop

Place Beeper Line

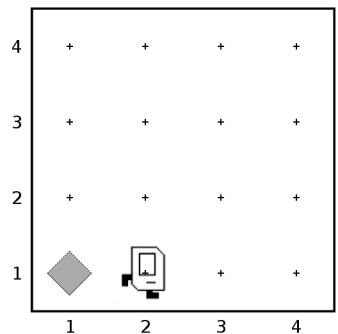
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



First time through the loop

Place Beeper Line

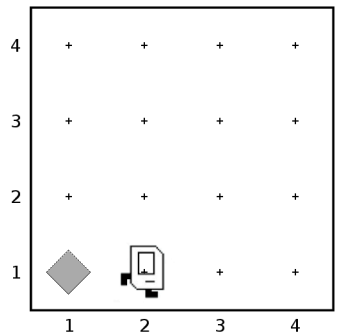
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



First time through the loop

Place Beeper Line

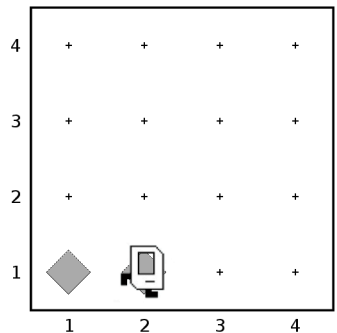
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Second time through the loop

Place Beeper Line

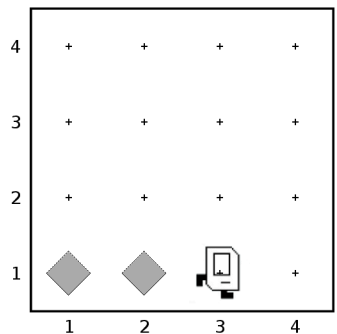
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Second time through the loop

Place Beeper Line

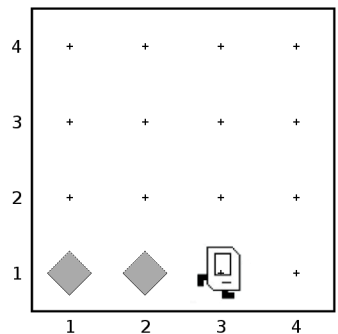
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Second time through the loop

Place Beeper Line

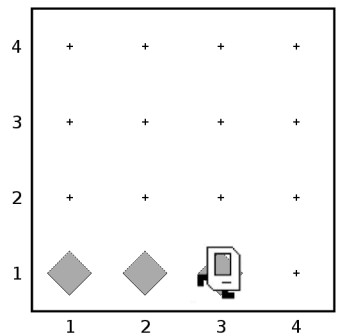
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Third time through the loop

Place Beeper Line

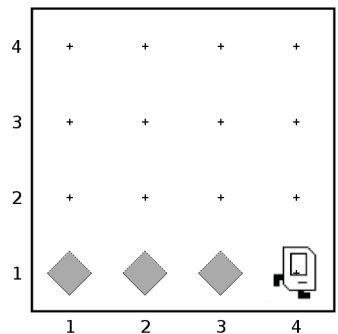
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Third time through the loop

Place Beeper Line

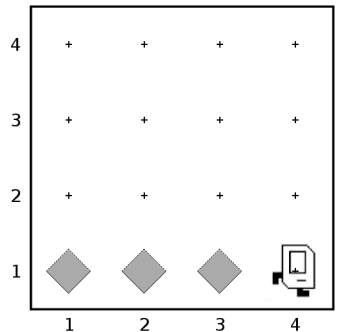
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Third time through the loop

Place Beeper Line

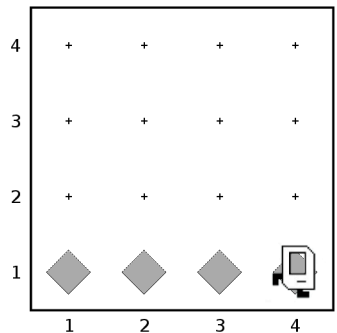
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Fourth time through the loop

Place Beeper Line

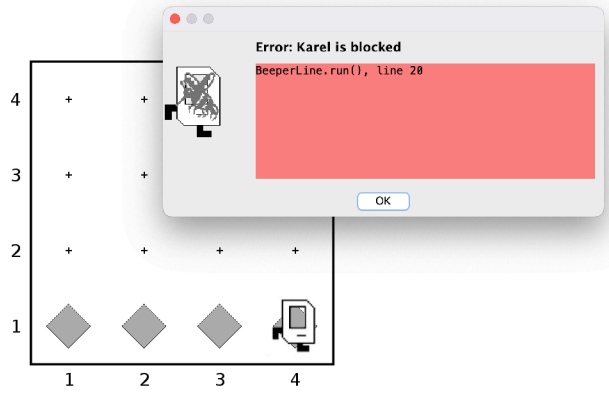
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Fourth time through the loop

Place Beeper Line

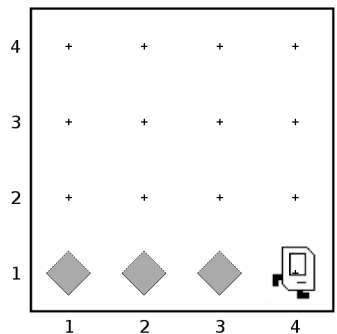
```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 4; i++) {
            putBeeper();
            move();
        }

    }
}
```



Fourth time through the loop

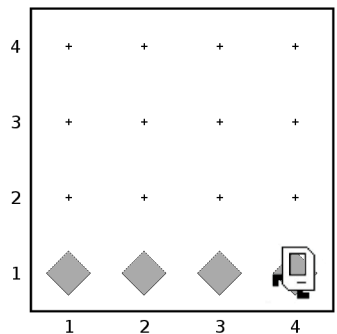
Place Beeper Line

```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 3; i++) {
            putBeeper();
            move();
        }

    }
}
```



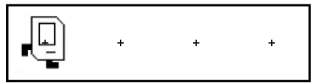
Place Beeper Line

```
public class BeeperLine extends SuperKarel {
    public void run() {
        for(int i = 0; i < 3; i++) {
            putBeeper();
            move();
        }
        putBeeper();
    }
}
```

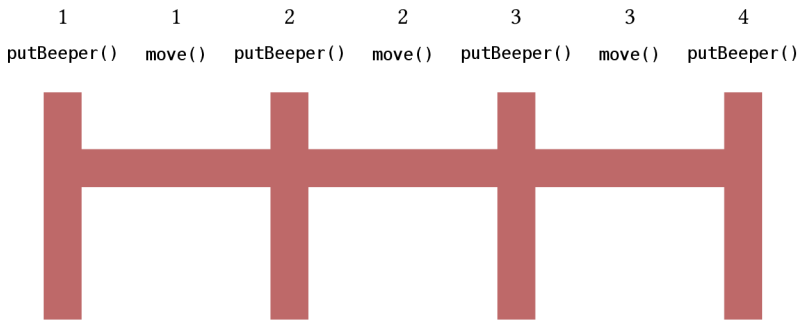


Fence Post Problem

Notice:

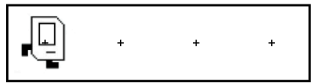


We put a beeper 4 times.
We moved 3 times.

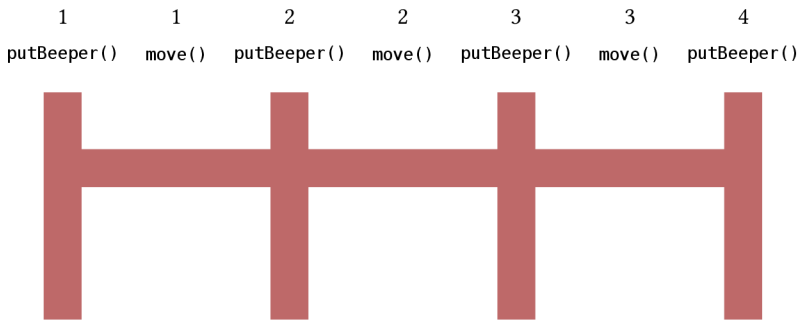


Fence Post Problem

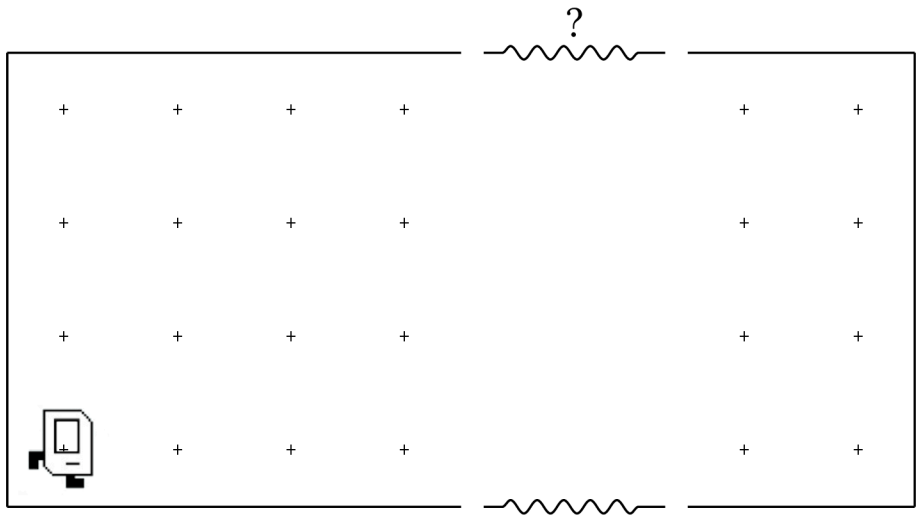
Notice:



We put a beeper $N + 1$ times.
We moved N times.



Don’t Know World Size



While Loops

While Loop

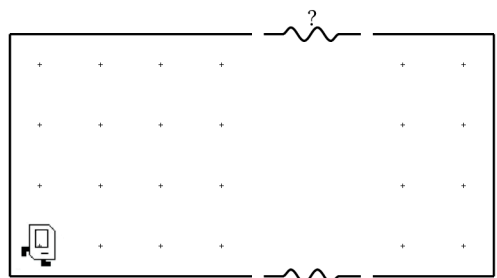
```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(condition) {
            code to repeat
        }

    }

}
```

Possible Conditions

Test	Opposite	What it checks
`frontIsClear()`	`frontIsBlocked()`	Is there a wall in front of Karel?
`leftIsClear()`	`leftIsBlocked()`	Is there a wall to Karel's left?
`rightIsClear()`	`rightIsBlocked()`	Is there a wall to Karel's right?
`beepersPresent()`	`noBeepersPresent()`	Are there beepers on this corner?
`beepersInBag()`	`noBeepersInBag()`	Any there beepers in Karel's bag?
`facingNorth()`	`notFacingNorth()`	Is Karel facing north?
`facingEast()`	`notFacingEast()`	Is Karel facing east?
`facingSouth()`	`notFacingSouth()`	Is Karel facing south?
`facingWest()`	`notFacingWest()`	Is Karel facing west?

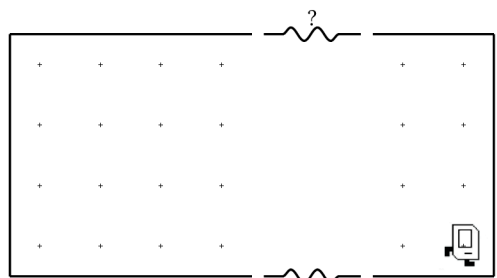
Move to Wall

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(condition) {
            code to repeat
        }

    }

}
```

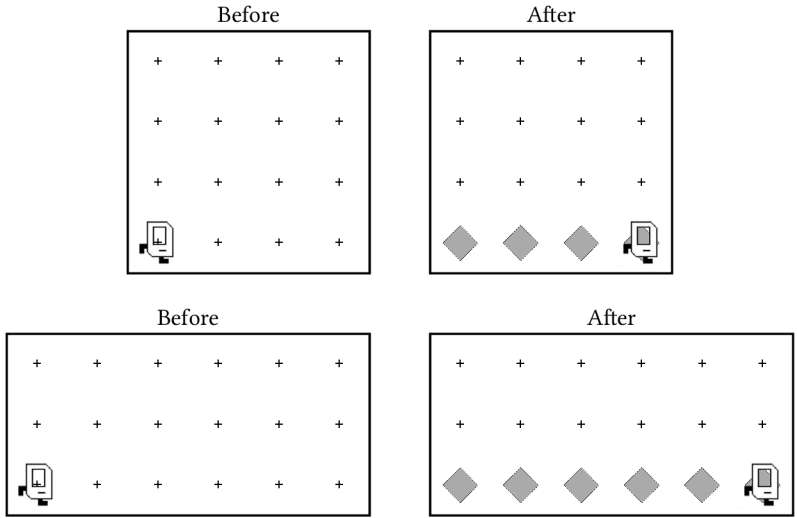


Move to Wall

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            move();
        }

    }

}
```



Code that Works in Any World



Place Beeper Line

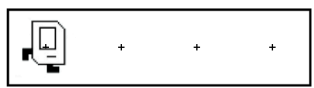
```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            move();
        }

    }

}
```

Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```

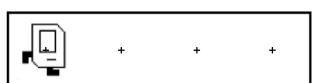
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



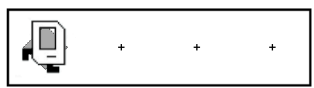
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



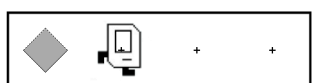
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



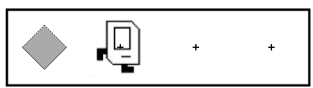
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



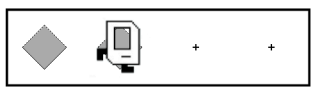
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



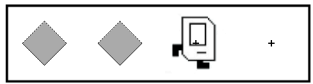
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



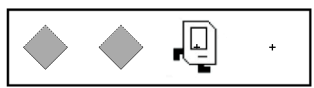
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



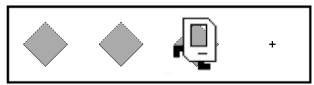
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



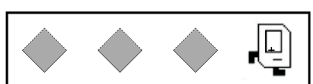
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



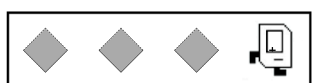
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



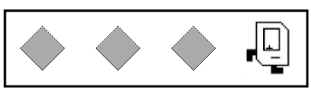
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```



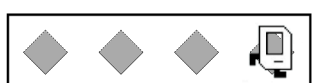
Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }

    }

}
```

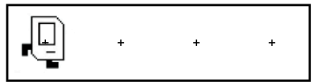


Place Beeper Line

```
public class WhileLoop extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        putBeeper();
    }

}
```



Fence Post Problem

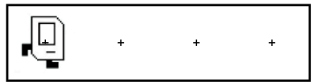


Inside loop

After loop

1	1	2	2	3	3	4
`putBeeper()`	`move()`	`putBeeper()`	`move()`	`putBeeper()`	`move()`	`putBeeper()`

Fence Post Problem

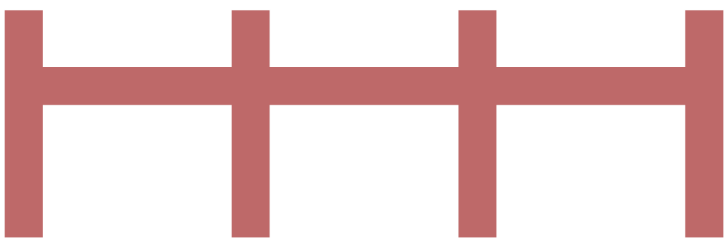


Before loop

1
putBeeper()

Inside loop

1	2	2	3	3	4
move()	putBeeper()	move()	putBeeper()	move()	putBeeper()

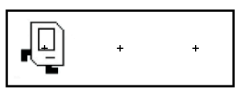


Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```

Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```



Place Beeper Line: Redux

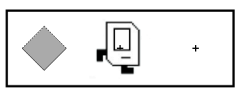
```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```

Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```

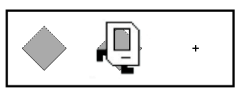
Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```



Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```

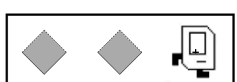


Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```

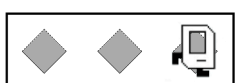
Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```



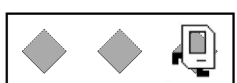
Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```



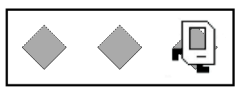
Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```

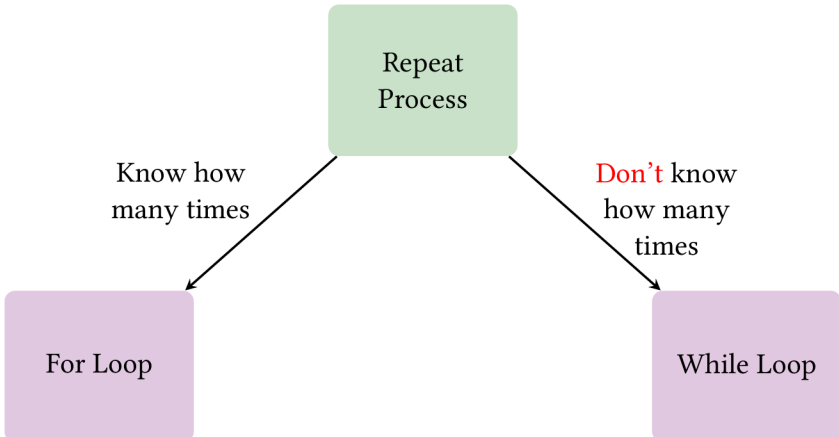


Place Beeper Line: Redux

```
public class WhileLoop extends SuperKarel {

    public void run() {
        // place a first beeper
        putBeeper();

        // example while loop
        while(frontIsClear()) {
            move();
            putBeeper();
        }
    }

}
```



Which Loop?


Repeat Process
Know how many times
Don't know how many times
For Loop
While Loop

If Statements

If Statement

```
public class IfExample extends SuperKarel {

    public void run() {

        // example of an if statement
        if(condition) {
            // code to run if condition is true
        }

    }

}
```

If Statement

```
public class IfExample extends SuperKarel{

    public void run() {
        safeMove();
    }

    private void safeMove() {
        if(frontIsClear()) {
            move();
        }
    }

}
```

If / Else Statement

```
public class IfExample extends SuperKarel{

    public void run() {
        invertBeeper();
    }

    private void invertBeeper() {
        if(beepersPresent()) {
            pickBeeper();
        } else {
            putBeeper();
        }
    }

}
```

The Full Karel

Built-in Karel commands:

```
move();
turnLeft();
putBeeper();
pickBeeper();
```

Karel program structure:

```
/*
 * Comments may be included anywhere in
 * the program between a slash-star and
 * the corresponding star-slash characters.
 */

import stanford.karel.*;

/* Definition of the new class */

public class name extends Karel {
   public void run() {
      statements in the body of the method
   }

   definitions of private methods

}
```

Conditional statements:

```
if (condition) {
   statements executed if condition is true
}

if (condition) {
   statements executed if condition is true
} else {
   statements executed if condition is false
}
```

Iterative statements:

```
for (int i = 0; i < count; i++) {
   statements to be repeated
}

while (condition) {
   statements to be repeated
}
```

Method definition:

```
private void name () {
   statements in the method body
}
```

Karel condition names:

`frontIsClear()`	`frontIsBlocked()`
`leftIsClear()`	`leftIsBlocked()`
`rightIsClear()`	`rightIsBlocked()`
`beepersPresent()`	`noBeepersPresent()`
`beepersInBag()`	`noBeepersInBag()`
`facingNorth()`	`notFacingNorth()`
`facingEast()`	`notFacingEast()`
`facingSouth()`	`notFacingSouth()`
`facingWest()`	`notFacingWest()`

New commands in the `SuperKarel` class:

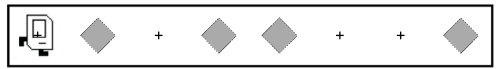
```
turnRight();
turnAround();
paintCorner(color);
```

New conditions in the `SuperKarel` class:

```
random()
random(p)
cornerColorIs(color)
```

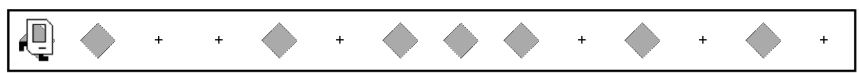
Paint By Beeper

Before



After

Before



After