



# Control Flow with Karel

## Lecture 2

---

CS106A, Summer 2019

Sarai Gould & Laura Cruz-Albrecht



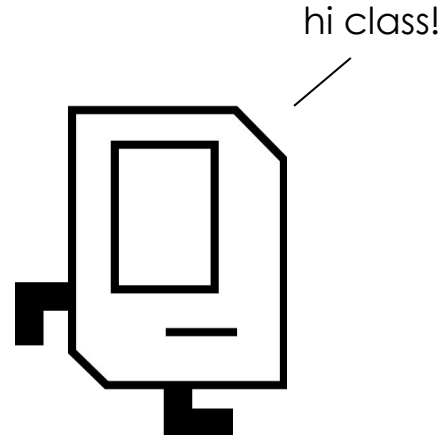
# Quick Announcements

Sign up for sections **by today at 5pm.**

Don't forget to fill out the form to tell us about yourself! It's on Piazza, the lecture slides, and the CS106a website.

# Plan for Today

- Karel's (Code) World
- Review: Our First Method!
- For Loops and Fence Posts
- While Loops
- If Statements



# Karel's (Code) World

```
import stanford.karel.*;

public class CollectNewspaperKarel extends SuperKarel {

    public void run(){

        // Your main program here!

    }

    // Create new methods for Karel down here!

}
```

# Karel's (Code) World

```
import stanford.karel.*;
```

**Programming Library We're Utilizing**

```
public class CollectNewspaperKarel extends SuperKarel {
```

```
    public void run(){
```

```
        // Your main program here!
```

```
    }
```

```
    // Create new methods for Karel down here!
```

```
}
```

# Karel's (Code) World

```
import stanford.karel.*; Name of Program
public class CollectNewspaperKarel extends SuperKarel {
    public void run(){
        // Your main program here!
    }
    // Create new methods for Karel down here!
}
```

# Karel's (Code) World

```
import stanford.karel.*;
```

This is the Type of Program We're Creating

```
public class CollectNewspaperKarel extends SuperKarel {
```

```
    public void run(){
```

```
        // Your main program here!
```

```
    }
```

```
    // Create new methods for Karel down here!
```

```
}
```

# Karel's (Code) World

```
import stanford.karel.*;

public class CollectNewspaperKarel extends SuperKarel {

    public void run(){

        // Your main program here!

    }

    // Create new methods for Karel down here!

}
```

**“run” Method - All of Our Programs Will Have This!**

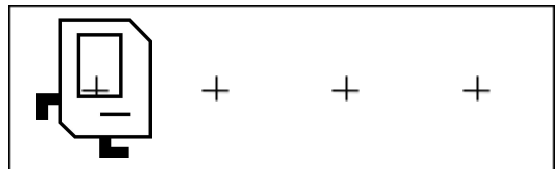


# Karel's (Code) World

```
import stanford.karel.*;
public class CollectNewspaperKarel extends SuperKarel {
    public void run(){
        // Your main program here!
    }
    // Create new methods for Karel down here!
}
```

**Brackets Noting the Beginning and End of a Code Block**

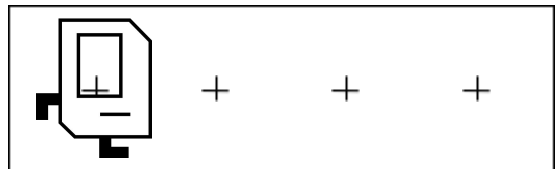
# A Method



A **method** is a set of new instructions we've created!

```
/* Comment describing method */  
private void nameOfMethod(){  
  
    // command 1  
    // command 2  
  
}
```

# A Method

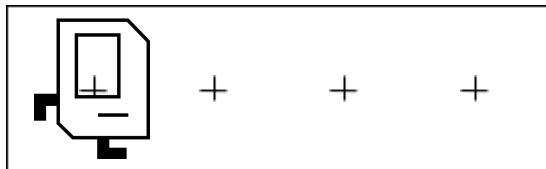


A **method** is a set of new instructions we've created!

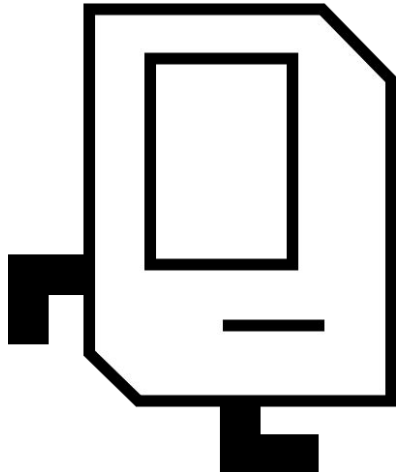
```
/* Allows Karel to turn right */  
private void turnRight(){  
  
    turnLeft();  
    turnLeft();  
    turnLeft();  
  
}
```

# Do It Again, Karel!

Karel recently got a job tiling floors. Can we help Karel fill this WHOLE world with tiles (aka beepers)?



# Reminder: What Can Karel Do?



Karel Can:

`move () ;`

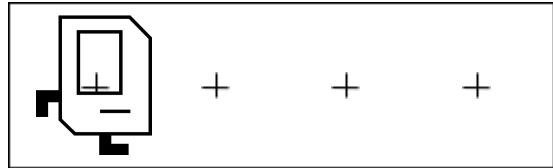
`turnLeft () ;`

`putBeeper () ;`

`pickBeeper () ;`

And now: `turnRight () ;`

# What's The Plan?

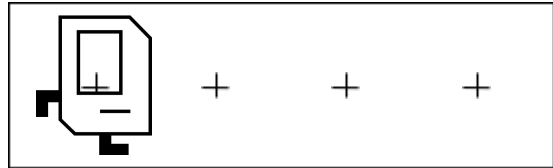


What's the Plan?

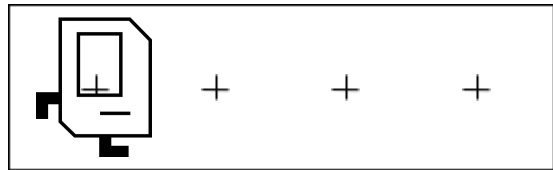
*Pseudocode or “fake code” is common term used to describe a programming plan.*

# What's The Pseudocode?

What's the Pseudocode?



# What's The Pseudocode?



What's the Pseudocode?

*place a beeper*

*move*

*place a beeper*

*move*

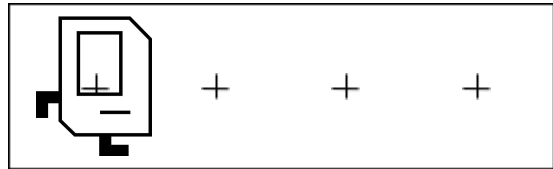
*place a beeper*

*move*

*place a beeper*



# What's The Pseudocode?



What's the Pseudocode?

*place a beeper*

*move*

*place a beeper*

*move*

*place a beeper*

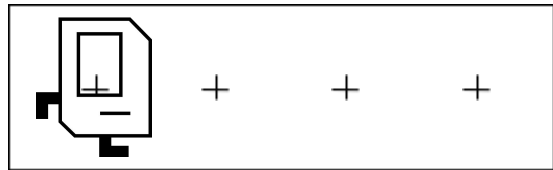
*move*

*place a beeper*

**Seems repetitive...**

**How can we  
simplify it?**

# What's The Pseudocode?



What's the Pseudocode?

*repeat three times:*

*place a beeper*

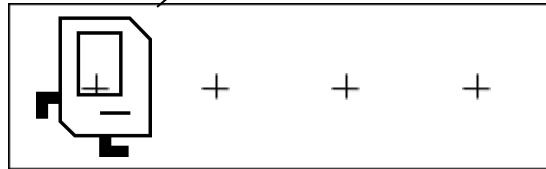
*move*

*place a beeper*

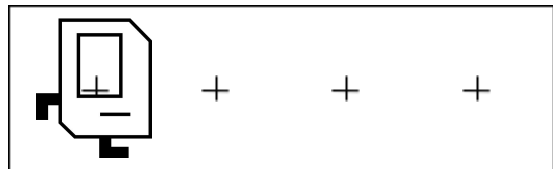
**Much better!**

# Wait! How Do We Repeat?

Ummmm, what do you mean by repeat? What does that code look like?



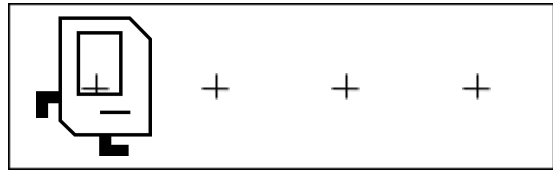
# For Loops



**For Loops** repeat something  
a *specific number of times!*

```
for(int i = 0; i < num_times; i++){  
    // command 1 to repeat!  
    // command 2 to repeat!  
}  
  
// code out here is NOT repeated!
```

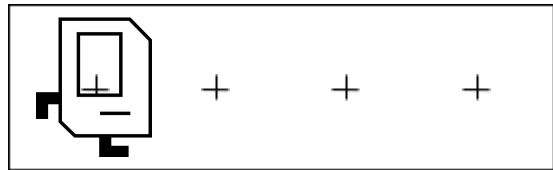
# For Loops



Our pseudocode:

*repeat three times:*  
    *place a beeper*  
    *move*  
*place a beeper*

# For Loops



Our pseudocode:

**repeat three times:**  
    *place a beeper*  
    *move*  
    *place a beeper*

Our Code!

```
for(int i = 0; i < 3; i++){  
  
}
```

# For Loops



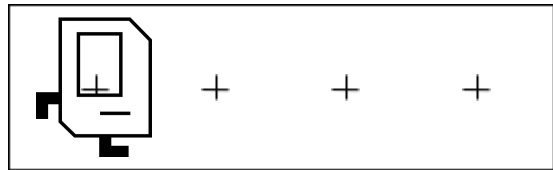
Our pseudocode:

*repeat three times:*  
*place a beeper*  
*move*  
*place a beeper*

Our Code!

```
for(int i = 0; i < 3; i++){  
    putBeeper();  
    move();  
}  
putBeeper();
```

# The Fencepost Pattern



## Notice:

We put a beeper **4 times**.

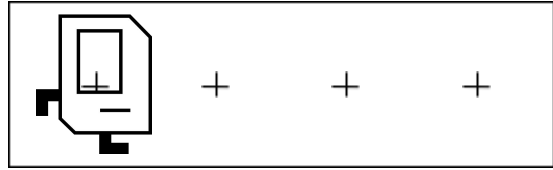
We moved **3 times**.

## Our Code!

```
for(int i = 0; i < 3; i++){  
    putBeeper();  
    move();  
}  
putBeeper();
```



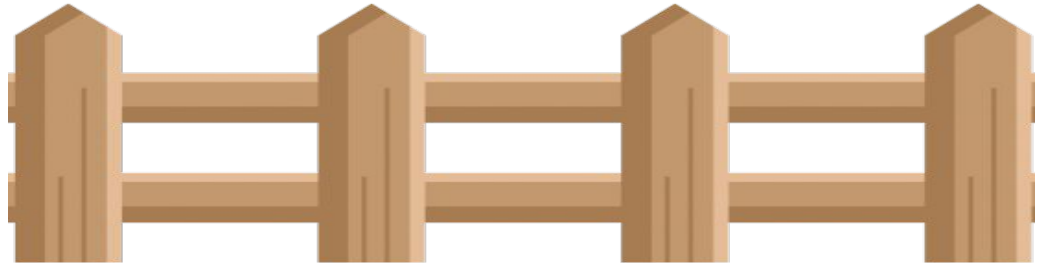
# The Fencepost Pattern



## Notice:

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

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



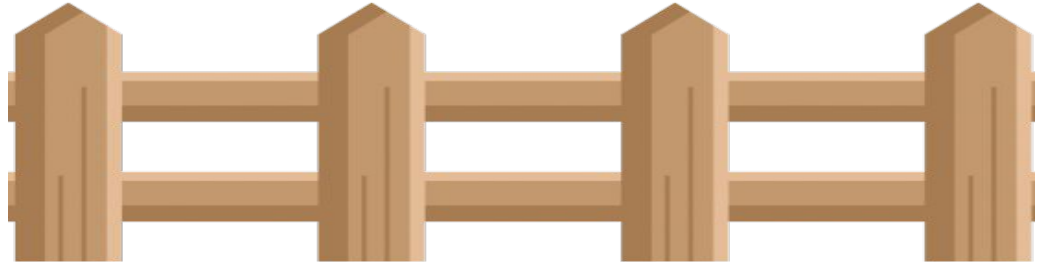
# The Fencepost Pattern



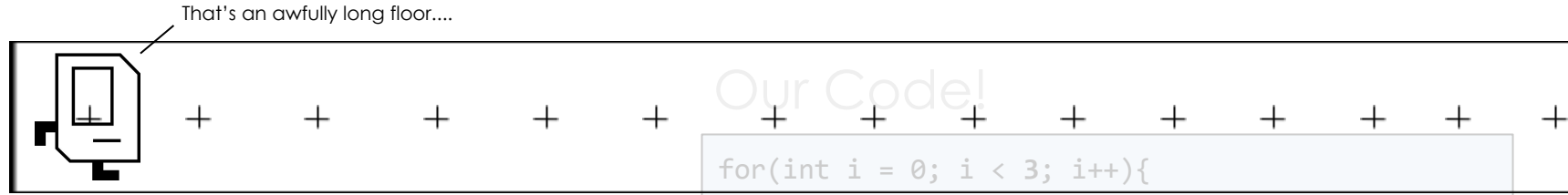
## Notice:

We put a beeper  **$n+1$  times**.  
We moved  **$n$  times**.

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



# What If...



Our Code!

```
for(int i = 0; i < 3; i++){  
  
    putBeeper();  
    move();
```

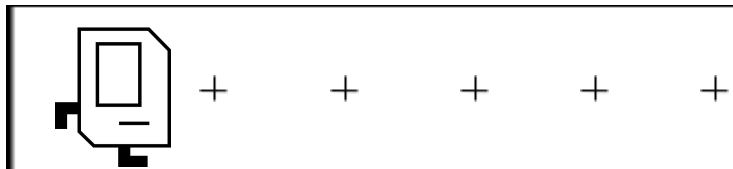
Our pseudocode:

*repeat three times:  
 place a beeper  
 move  
 place a beeper*

**What if we don't know how many  
tiles Karel will have to lay down?**

# While Loops

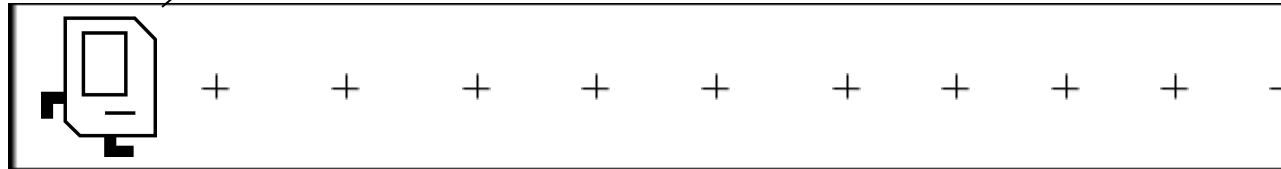
**While Loops** repeat *until* a condition is no longer met!



```
while(conditionIsTrue()){  
    // command 1 to repeat!  
    // command 2 to repeat!  
}  
  
// the condition isn't true anymore,  
// so our code exited the while loop.  
// code out here is NOT repeated!
```

# What Does Karel Know?

What conditions can I check? Can I check if *anything* is true?

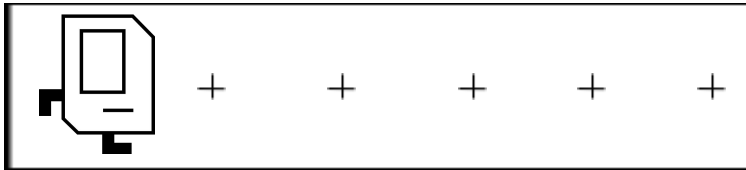


# What Conditions Can Karel Check?

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

# Tiling the Floor with While Loops

What's the Pseudocode?



# Tiling the Floor with While Loops

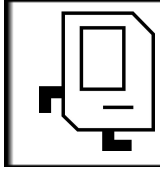
What's the Pseudocode?

while nothing is in front of Karel:  
    place a beeper  
    move





# Tiling the Floor with While Loops



+ + + + +

Our Code!

Our pseudocode:

```
while nothing is in front of Karel:  
  place a beeper  
  move
```

# Tiling the Floor with While Loops



Our pseudocode:

**while nothing is in front of Karel:**

place a beeper

move

Our Code!

```
while(frontIsClear()){  
  
}
```

# Tiling the Floor with While Loops



Our pseudocode:

while nothing is in front of Karel:  
  place a beeper  
  move

Our Code!

```
while(frontIsClear()){  
    putBeeper();  
    move();  
}
```

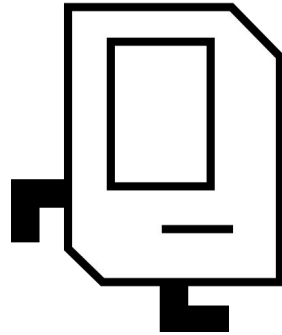
# Let's Code It!

---

# Don't Forget to Test!

Remember to test early and test often.

Test parts of your program as you go along to catch bugs along the way!



# What Are We Missing?

What's the Pseudocode?

while nothing is in front of Karel:

    place a beeper  
    move



# Tiling the Floor with While Loops



What's the Pseudocode?

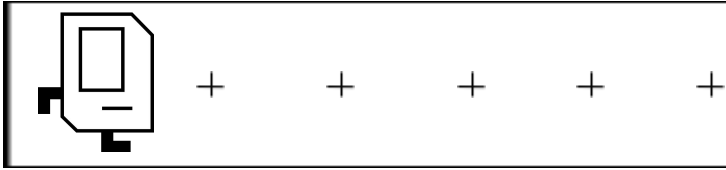
while nothing is in front of Karel:

    place a beeper

    move

place a beeper

# Tiling the Floor with While Loops



Our pseudocode:

while nothing is in front of Karel:

    place a beeper

    move

place a beeper

Our Code!

```
while(frontIsClear()){  
    putBeeper();  
    move();  
}  
putBeeper();
```



# Let's Code It!

---

# Tiling the Floor with While Loops

Our pseudocode:

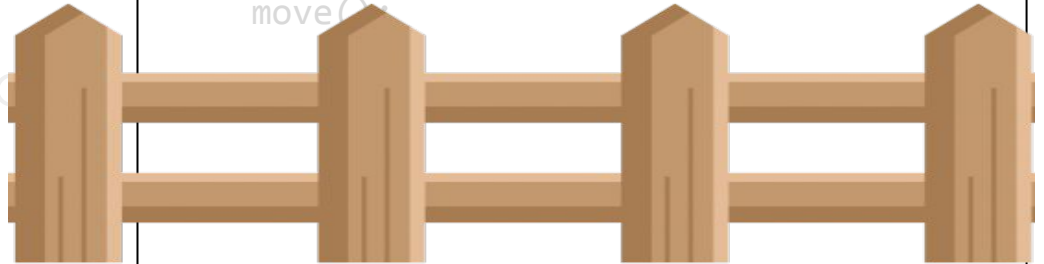
**Looks like another fencepost program!**

place a beeper  
move

place a beeper

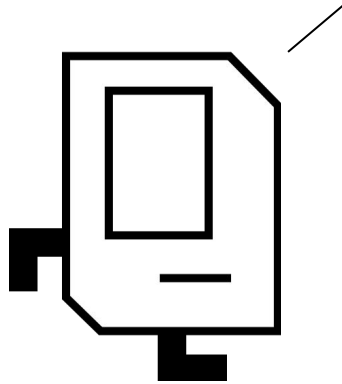
Our Code!

```
while(frontIsClear()){  
    putBeeper();  
    move();  
}
```

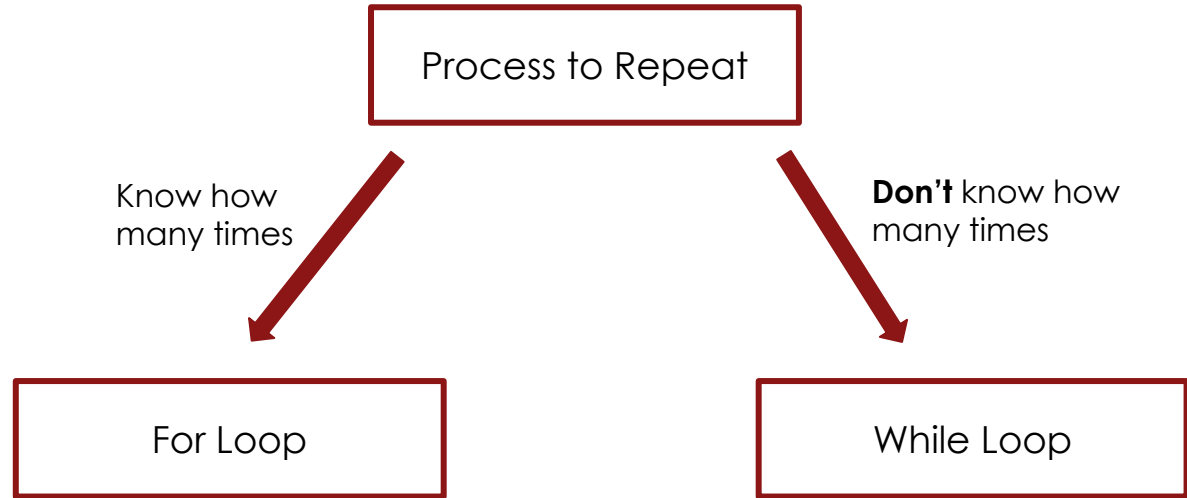
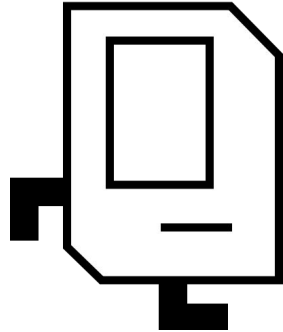


# While Loop or For Loop?

How do I know when I should use a while loop or a for loop?

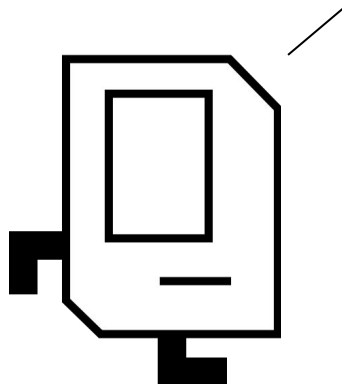


# While Loop or For Loop?



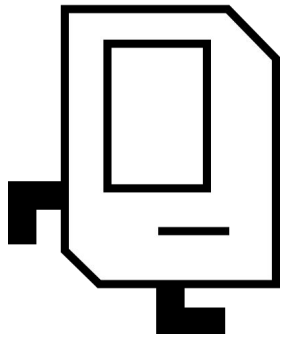
# What Is It Now, Karel?

All of this repeating is  
making me *tired*!  
What if I only want to check  
for something only **once**?



# If Statements

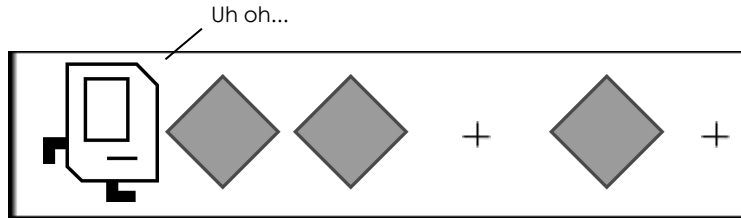
**If statements** check a condition *once*!



Phew, that looks  
much better...

```
if(conditionIsTrue()){  
  
    // command 1!  
    // command 2!  
  
}  
  
// we have left the if statement.  
// code out here happens no matter what!
```

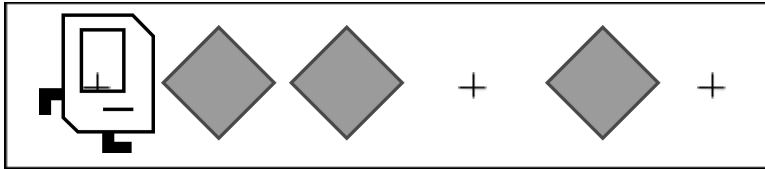
# Karel, Let's Pick Up These Old Tiles!



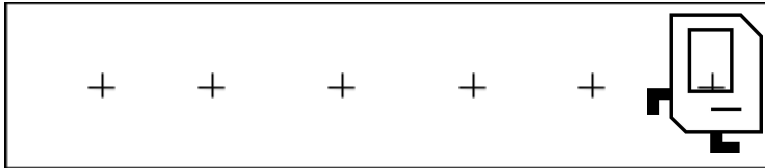
Karel must pick up all of the old tiles that were left behind!

# Tile Clean Up Example

Before:



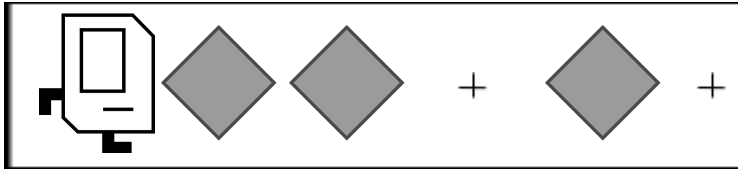
After:





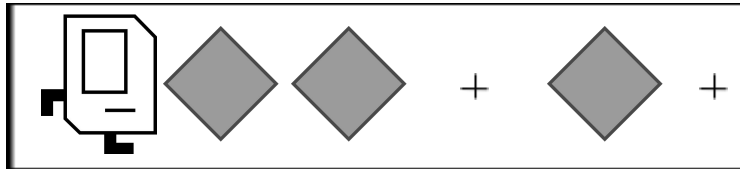
# Karel, Let's Clean Up!

What's the Pseudocode?



# Karel, Let's Clean Up!

What's the Pseudocode?



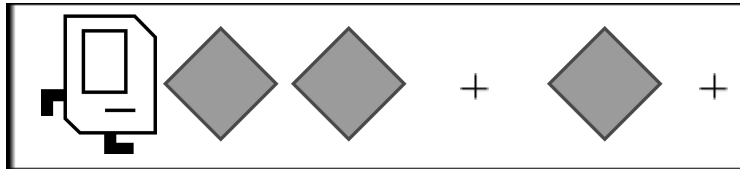
while nothing is in front of Karel:

pick up beeper

move

# Karel, Let's Clean Up!

What's the Pseudocode?



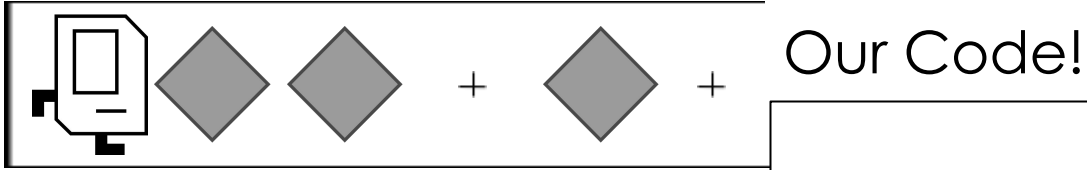
while nothing is in front of Karel:

pick up beeper

move

pick up beeper

# Cleaning Up

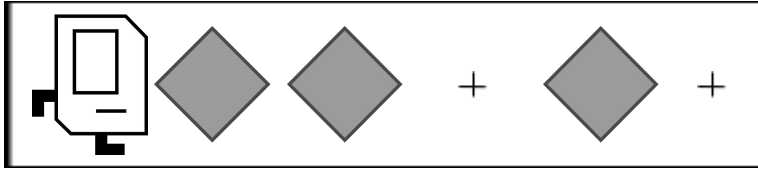


Our pseudocode:

```
while nothing is in front of Karel:  
    pick up beeper  
    move  
    pick up beeper
```



# Cleaning Up



Our pseudocode:

```
while nothing is in front of Karel:  
    pick up beeper  
    move  
    pick up beeper
```

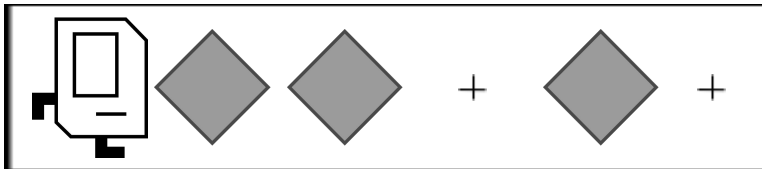
Our Code!

```
while(frontIsClear()){  
    pickBeeper();  
    move();  
}  
pickBeeper();
```

# Let's Code It!

---

# UH OH!



## Our Code!

### Our pseudocode:

while nothing is in front of Karel:

**pick up beeper**

move

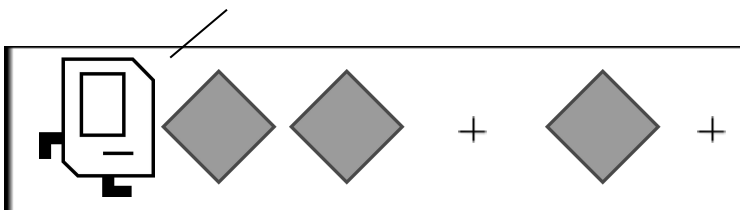
pick up beeper

```
while(frontIsClear()){  
    pickBeeper();  
    move();  
}  
pickBeeper();
```

**You can't pick  
up a beeper if  
there isn't one  
there!**

# UH OH!

Maybe we should add  
an if statement to check  
for a beeper!



What's the Pseudocode?

while nothing is in front of Karel:

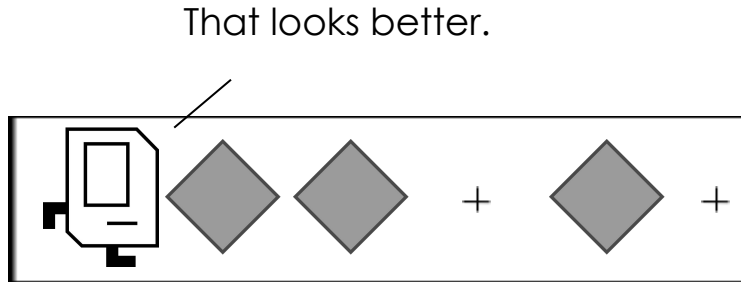
**pick up beeper**

move

pick up beeper



# Cleaning Up with If Statements



What's the Pseudocode?

while nothing is in front of Karel:

**if there is a beeper:**

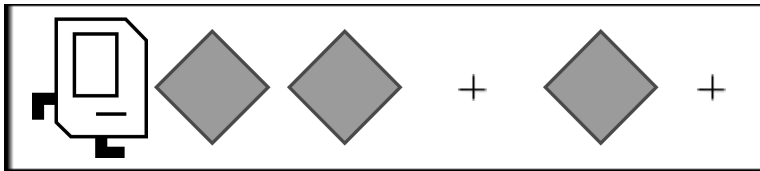
pick up beeper

move

**if there is a beeper:**

pick up beeper

# Old: Cleaning Up



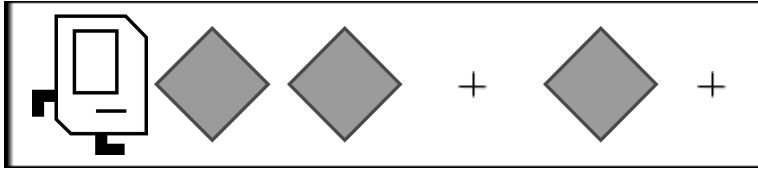
Our pseudocode:

```
while nothing is in front of Karel:  
    pick up beeper  
    move  
    pick up beeper
```

Our Code!

```
while(frontIsClear()){  
    pickBeeper();  
    move();  
}  
pickBeeper();
```

# Cleaning Up with If Statements



## Our pseudocode:

while nothing is in front of Karel:

**if there is a beeper:**

pick up beeper

move

**if there is a beeper:**

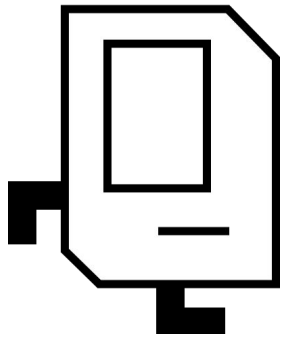
pick up beeper

## Our Code!

```
while(frontIsClear()){  
    if(beepersPresent()){  
        pickBeeper();  
    }  
    move();  
}  
  
if(beepersPresent()){  
    pickBeeper();  
}
```

# If Statements

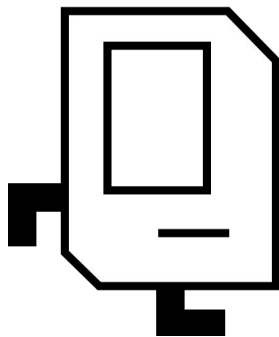
**If statements** check a condition *once!*



Phew, that looks  
much better...

```
if(conditionIsTrue()){  
  
    // command 1!  
    // command 2!  
  
}  
  
// we have left the if statement.  
// code out here happens no matter what!
```

# One Last Thing: If-Else



The **else** code only runs if the condition is *not* true!

```
if(conditionIsTrue()){  
  
    // command 1 if conditionIsTrue()!  
    // command 2 if conditionIsTrue()!  
  
} else {  
  
    // command 1 if conditionIsFalse()!  
    // command 2 if conditionIsFalse()!  
  
}  
// code out here happens no matter what!
```

# Plan for Today

- Karel's (Code) World
- Review: Our First Method!
- For Loops and Fence Posts
- While Loops
- If Statements

## Homework:

- Set up Eclipse (instructions on course website)
- **Due 5pm Today:** Sign up for section on course website
- **Assignment 0:** Tell us about yourself - <https://bit.ly/2X0Pmzz>