



# CS106A: Programming Methodologies



# Chris Piech

My parents are interesting folks

I originally concentrated in graphics and worked at Pixar

- Childhood: Nairobi, Kenya
- High School: Kuala Lumpur, Malaysia
- Stanford University Ph.D. in Neural Networks
- Research lab on AI for Social Good

The problem I really want to solve is to make high quality more education accessible



# Chris Piech



11 years ago to this day, I was sitting in your seats

Piech, CS106A, Stanford University



# Head TA: Julia Daniel



Piech, CS106A, Stanford University



# Section Leaders



**Luciano  
Gonzalez** ✉



**Maggie  
Davis** ✉



**Marilyn  
Zhang** ✉



**Meng  
Zhang** ✉



**Nidhi  
Manoj** ✉



**Niki  
Agrawal** ✉



**Peter  
Maldonado** ✉



**Rachel  
Gardner** ✉



**Rhea  
Karuturi** ✉



**Robbie  
Jones** ✉



**Ruiqi  
Chen** ✉



**Semir  
Shafi** ✉



**Shanon  
Reckinger** ✉



**Tessera  
Chin** ✉



**Thariq  
Ridha** ✉



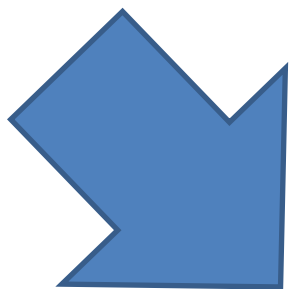
**Vineet  
Kosaraju** ✉



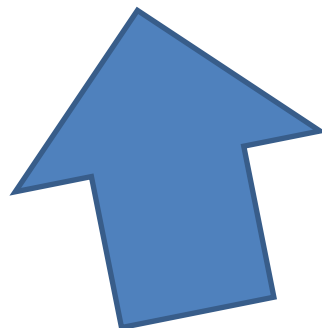
# Course mechanics

(this is a light version. Please read the handout for details).

# Course Website



<http://cs106a.stanford.edu>



# Prerequisite Test





# Lectures and Sections

- Weekly 50-min section led by awesome section leaders (the backbone of the class!)
- Signups begin on Thursday at 5:00pm and close Sunday at 5:00pm



Piech, CS106A, Stanford University



# Office Hours



LaIR: 7pm until 11pm Sunday through Thursday  
(starting next Sunday)

Piech, CS106A, Stanford University



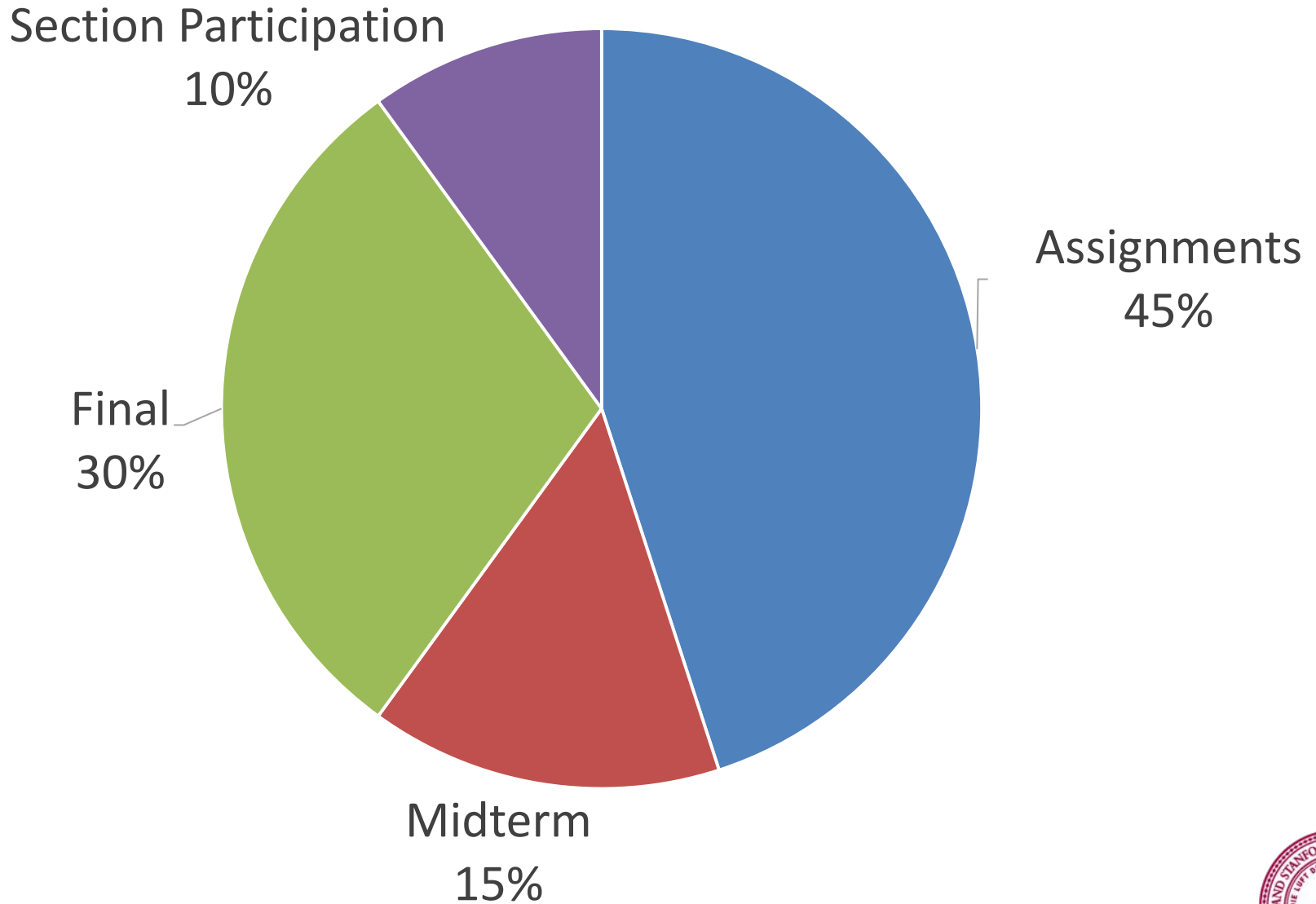
# Grading Scale

Functionality and style grades for the assignments use the following scale:

- ++** A submission so good it “makes you weep.”
- +** Exceeds requirements.
- ✓+** Satisfies all requirements of the assignment.
- ✓** Meets most requirements, but with some problems.
- ✓-** Has more serious problems.
- Is even worse than that.
- Better than nothing.



# What we will ask you to do



Piech, CS106A, Sta \*Two free late days



# Optional Contest



Piech, CS106A, Stanford University



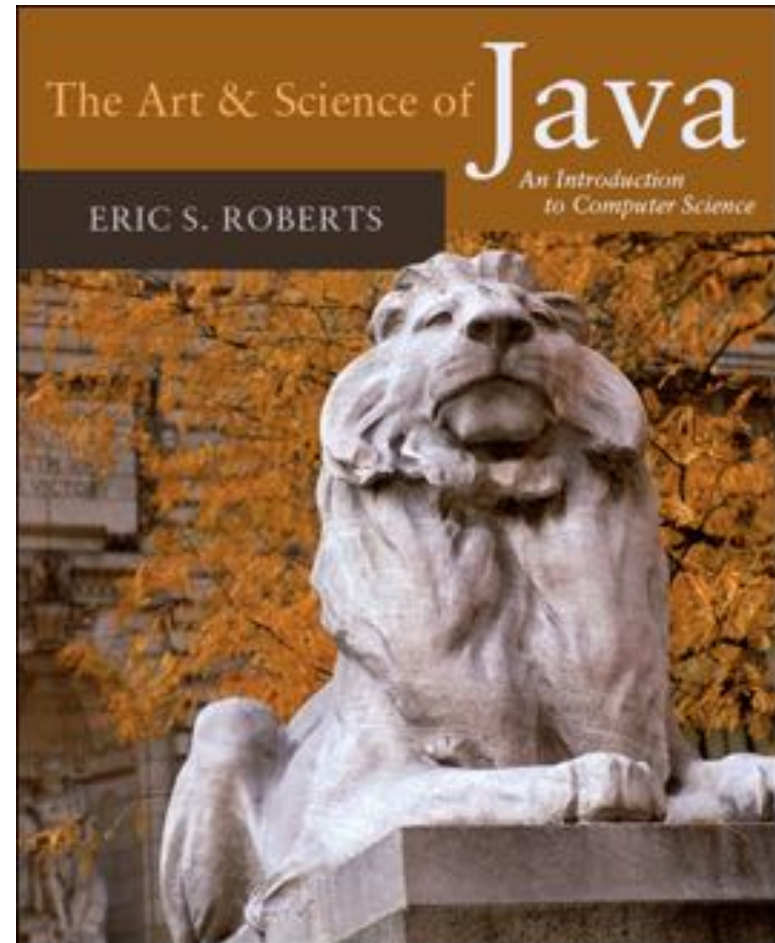
# Textbook

## *The Art & Science of Java* by Eric Roberts

- written here at Stanford
- tailored to this course
- a valuable reference
- usable on open-book exams

## Karel the Robot Learns Java

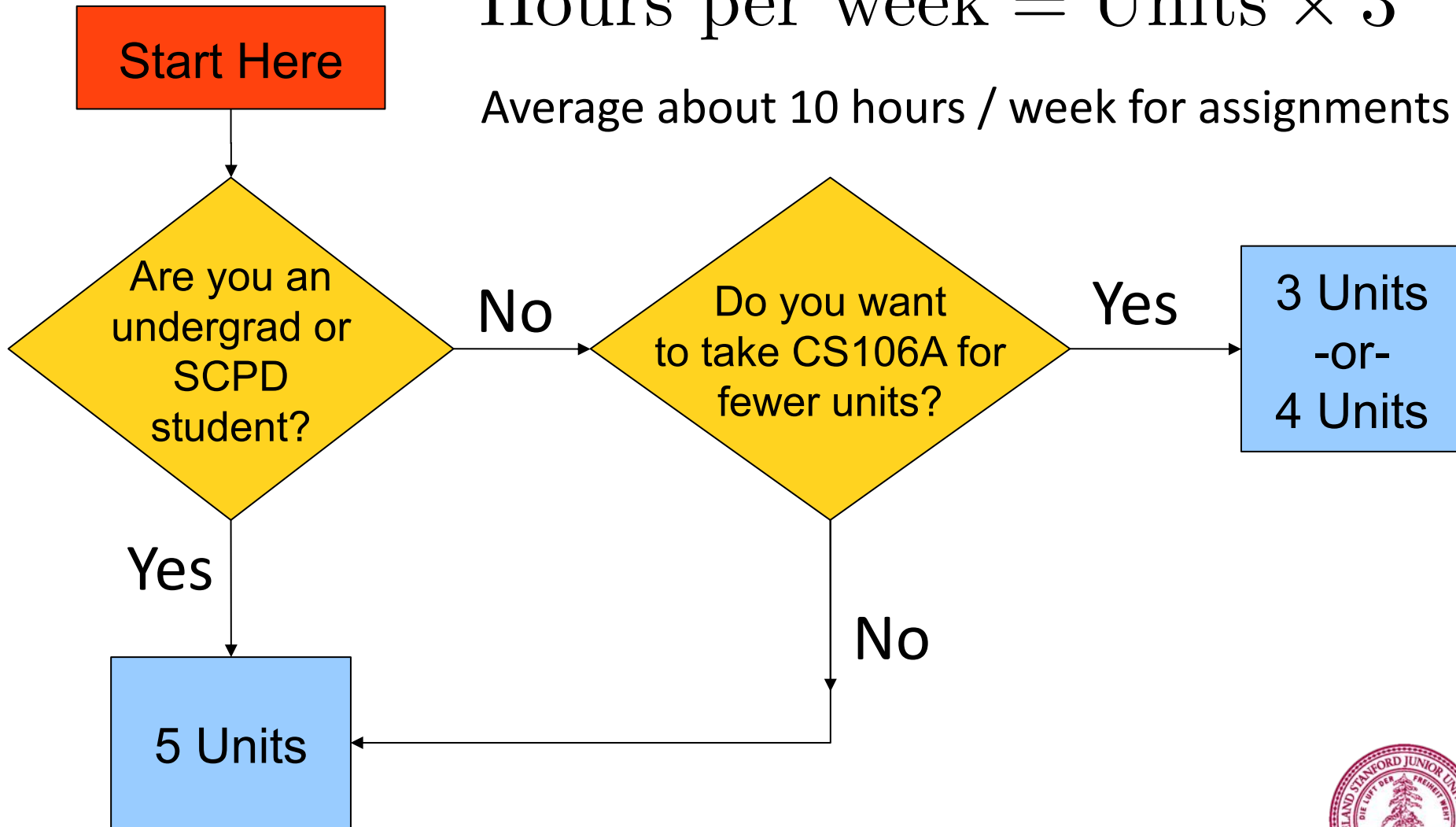
- First week of material



# CS106A Units

Hours per week = Units  $\times$  3

Average about 10 hours / week for assignments



Are you in the right place?



What is CS106A?

# Computer Science

Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do.

- Michael Fellows and Ian Parberry



# Learning Goals

- *Learn how to harness computing power to solve problems.*
- To that end:
  - Explore fundamental techniques in computer programming.
  - Develop good software engineering style.
  - Gain familiarity with the Java programming language.



There are a lot of cool  
programs you may one day  
write

# Computer Graphics



Pat Hanrahan, one of the founders of Pixar is a professor here

Piech, CS106A, Stanford University



# Consumer Applications



# Autonomous Surgery



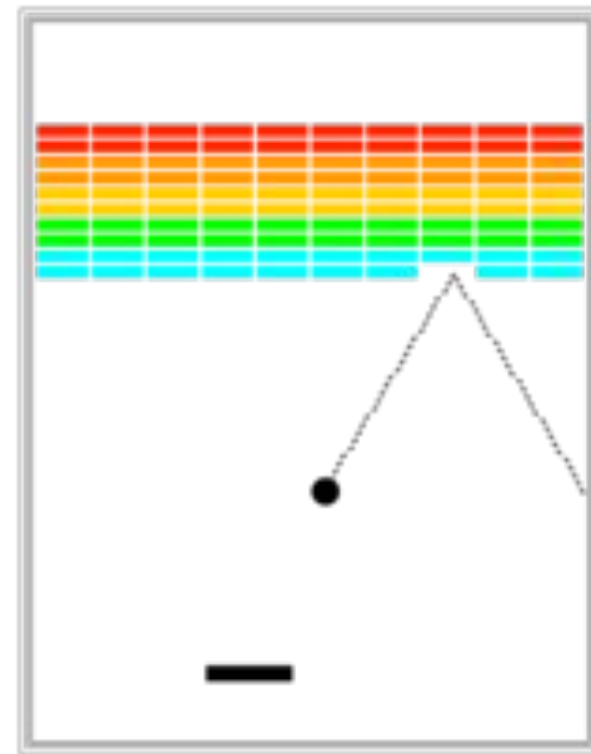
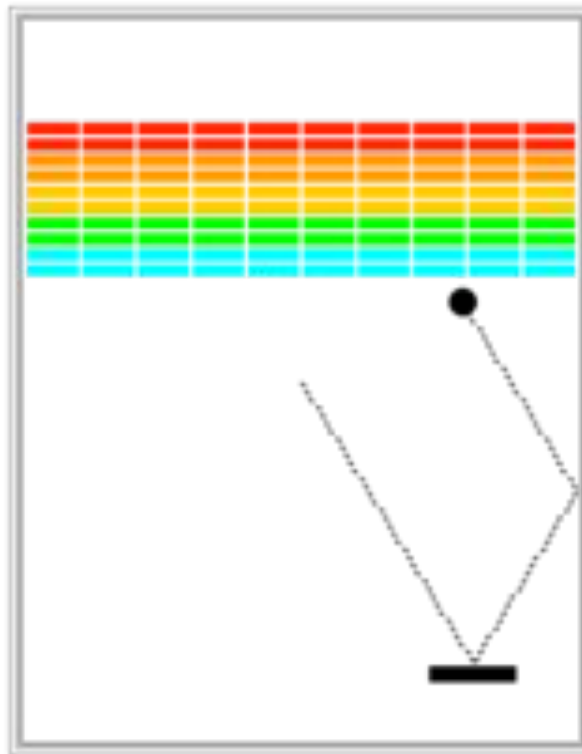
# Self Driving Car



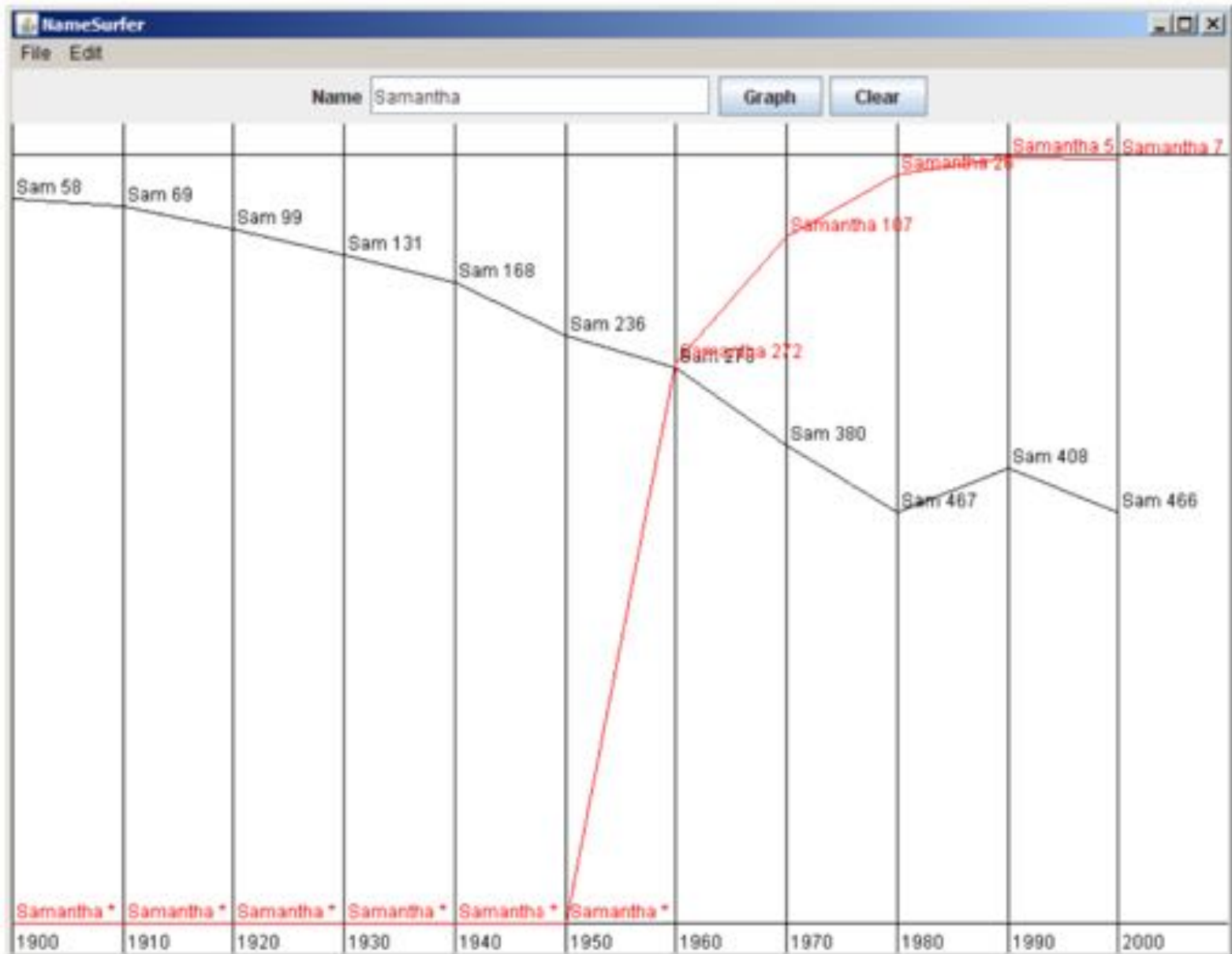


If only we could program self  
driving cars...

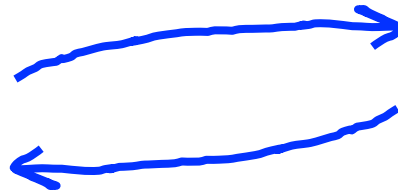
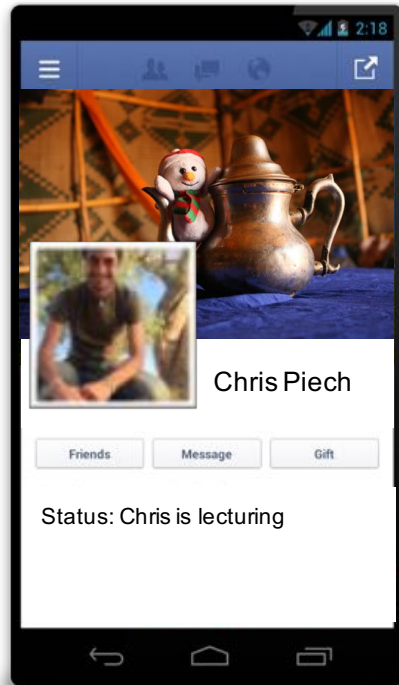
# Graphical Games



# Data Visualization



# Internet Based Program



```
FacePamphletServer
Starting server on port 8000...
addProfile (name=Mehran)
=> success
addProfile (name=Chris)
=> success
addProfile (name=Chris)
=> Error: Database already contains Chris.
getStatus (name=Chris)
=> none
setStatus (name=Chris, status=teaching)
=> success
getStatus (name=Chris)
=> teaching
addFriend (name2=Mehran, name1=Chris)
=> success
getFriends (name=Chris)
=> [Mehran]
addProfile (name=Julie)
=> success
getImg (name=Julie)
=> none
getStatus (name=Julie)
=> none
getFriends (name=Julie)
=> []
setImg (img=JulieZ.jpg, name=Julie)
=> success
getImg (name=Julie)
=> JulieZ.jpg
getStatus (name=Julie)
=> none
getFriends (name=Julie)
=> []
addFriend (name2=Chris, name1=Julie)
=> success
getImg (name=Julie)
=> JulieZ.jpg
getStatus (name=Julie)
=> none
```



# Art of Computer *Science*



# Strive for Everyone to Succeed



# Lets Get Started



# Meet Karel the Robot

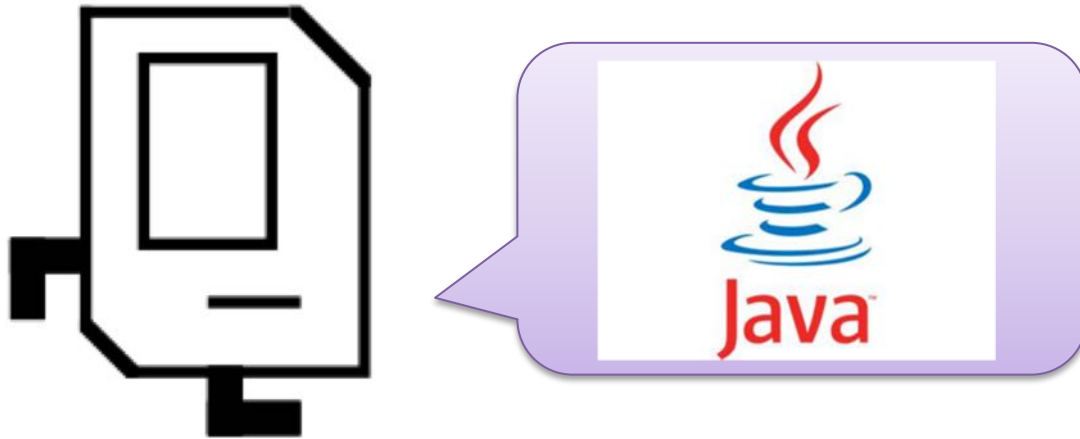


Good morning





# Karel Speaks Java

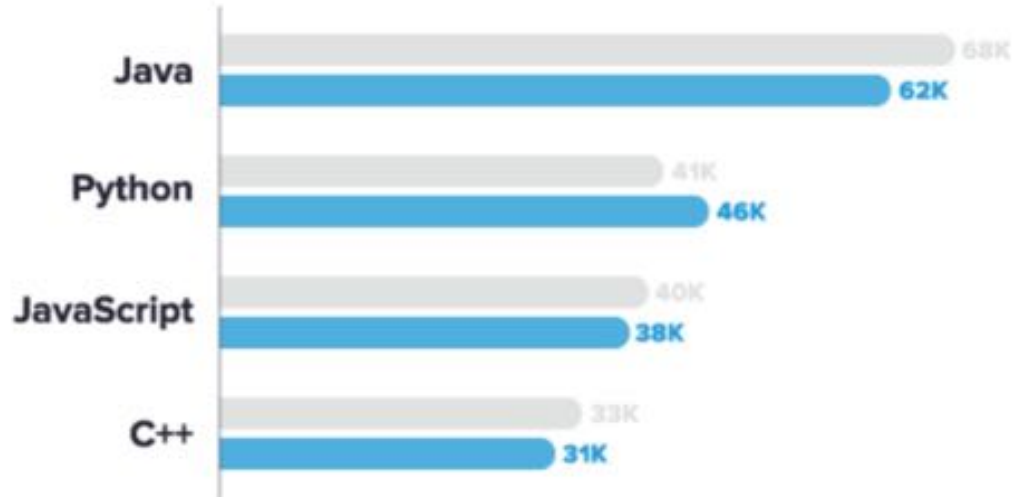


# Why Java?

1

## Job postings containing top languages

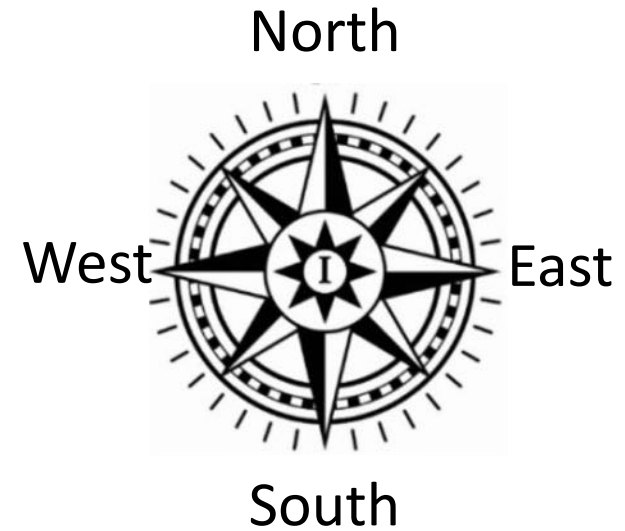
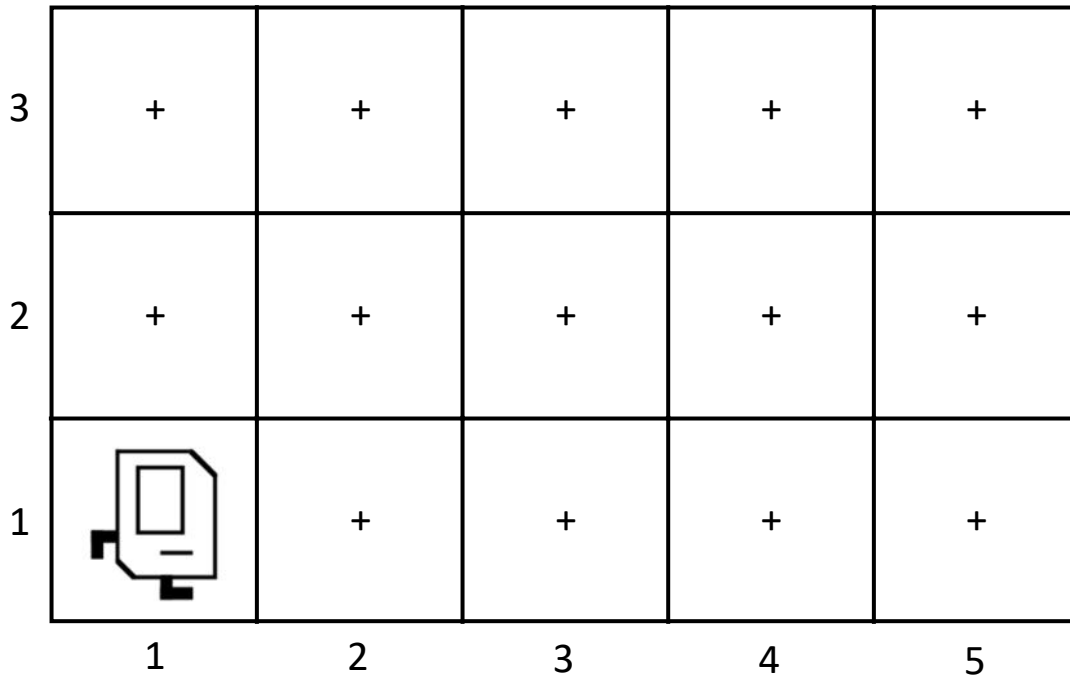
Indeed.com - November, 17th 2017



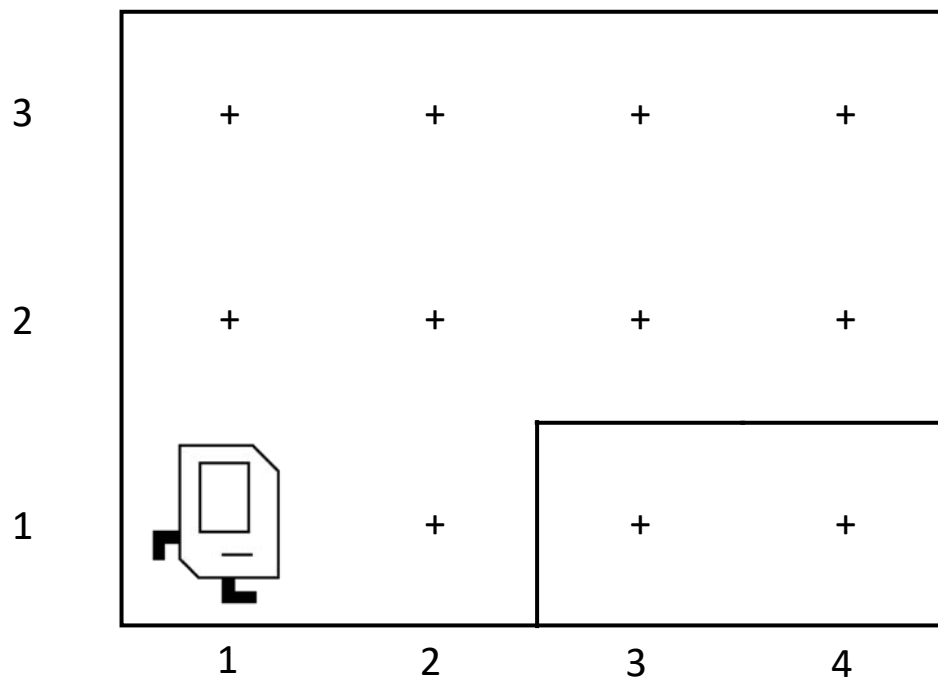
2



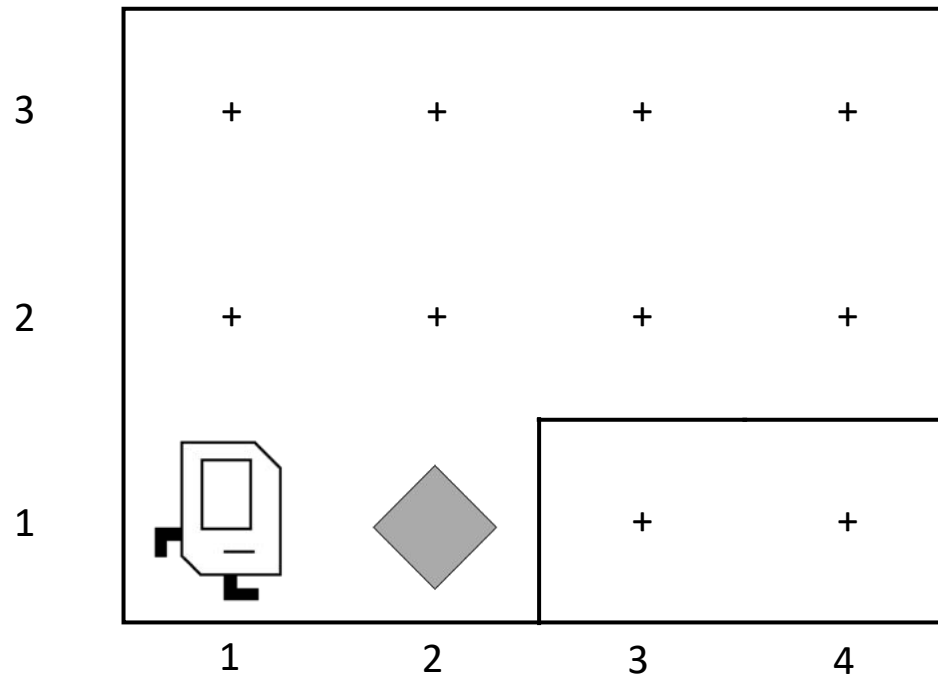
# Karel's World



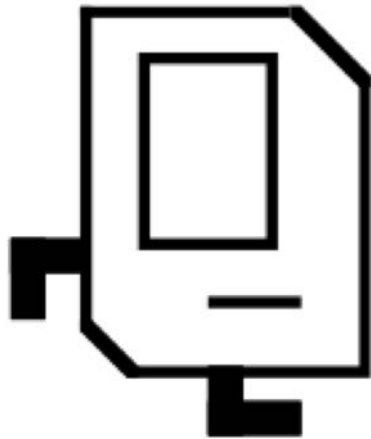
# Walls



# Beepers



# Knows Four Commands



```
move();
```

```
turnLeft();
```

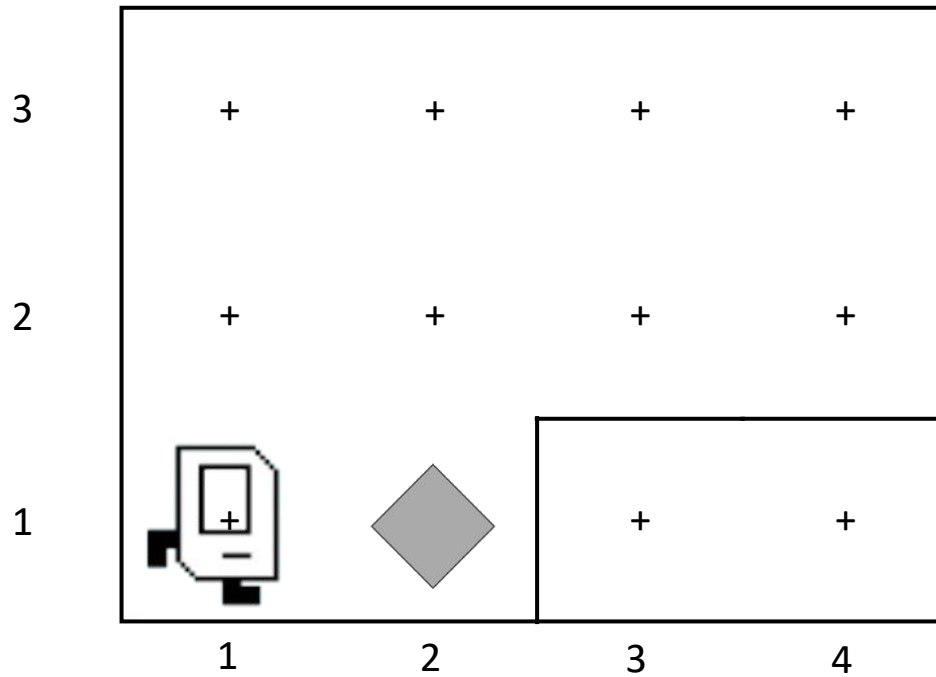
```
putBeeper();
```

```
pickBeeper();
```



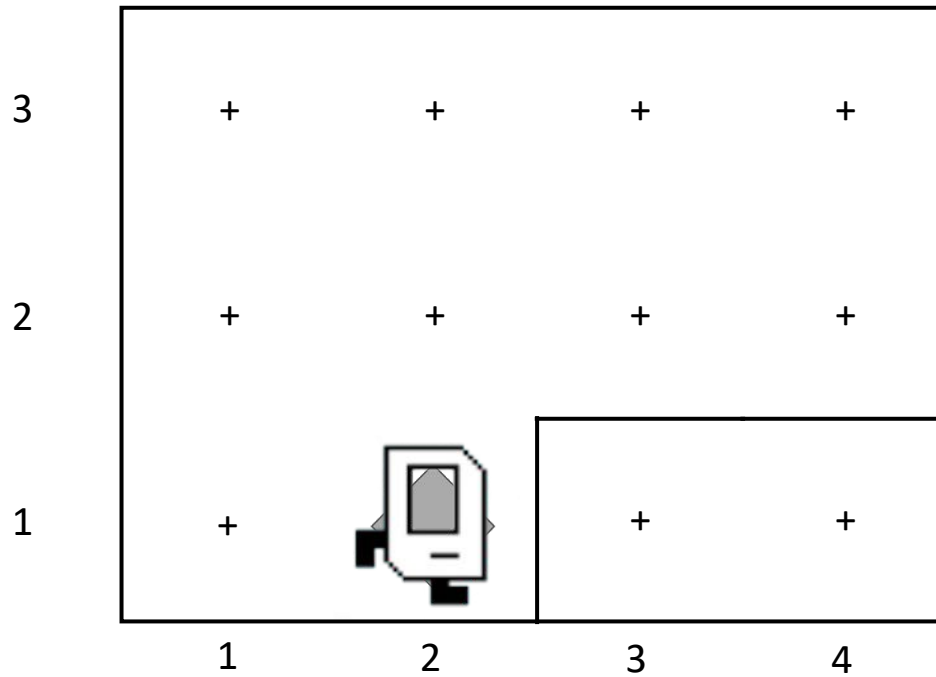
```
move ( ) ;
```

# move ( ) ;



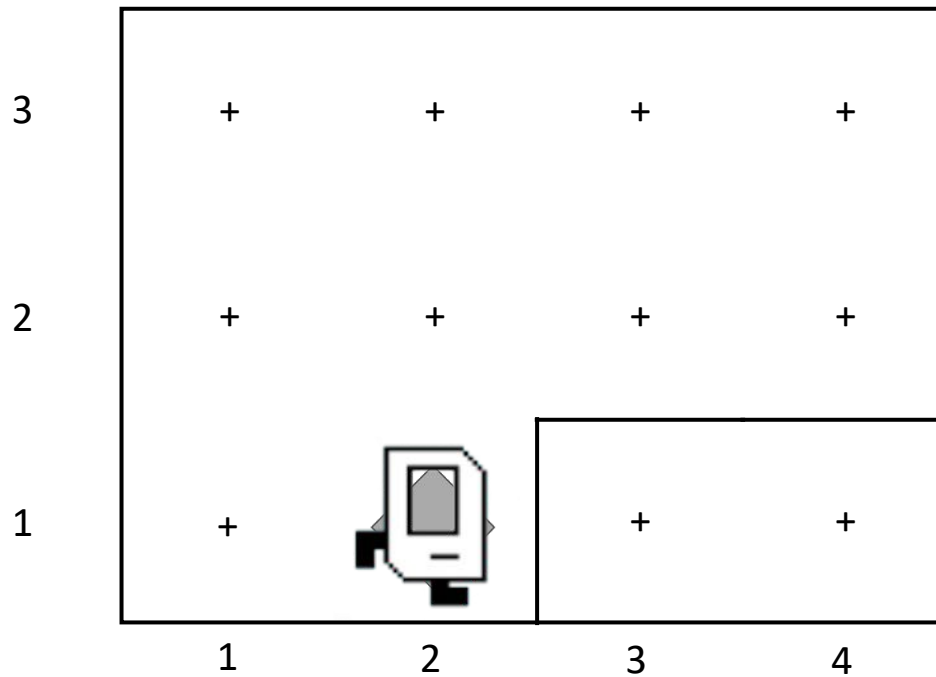


# move ( ) ;

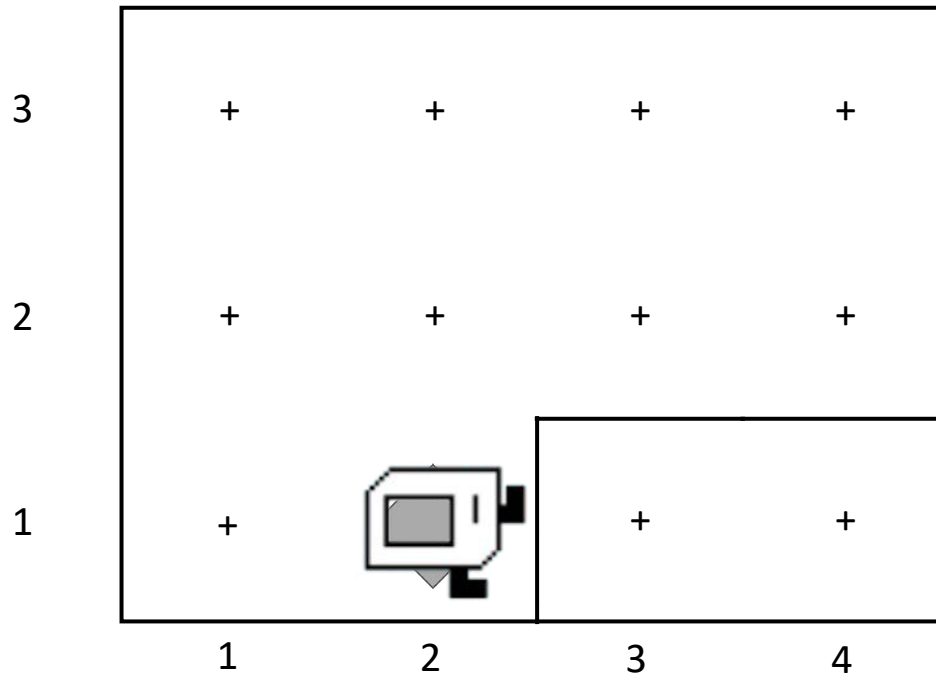


```
turnLeft();
```

# turnLeft();

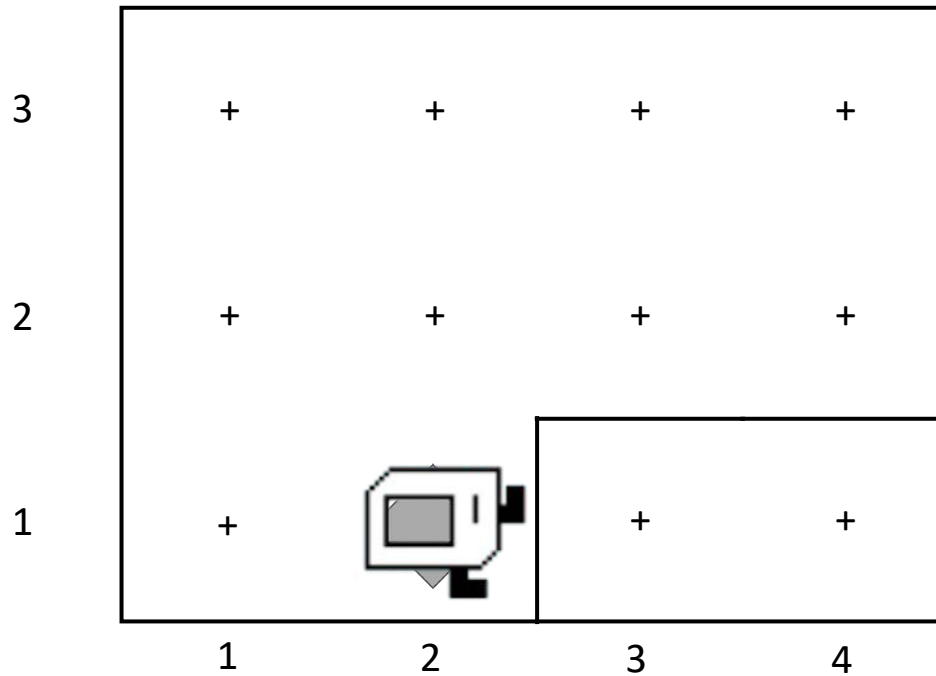


# turnLeft();

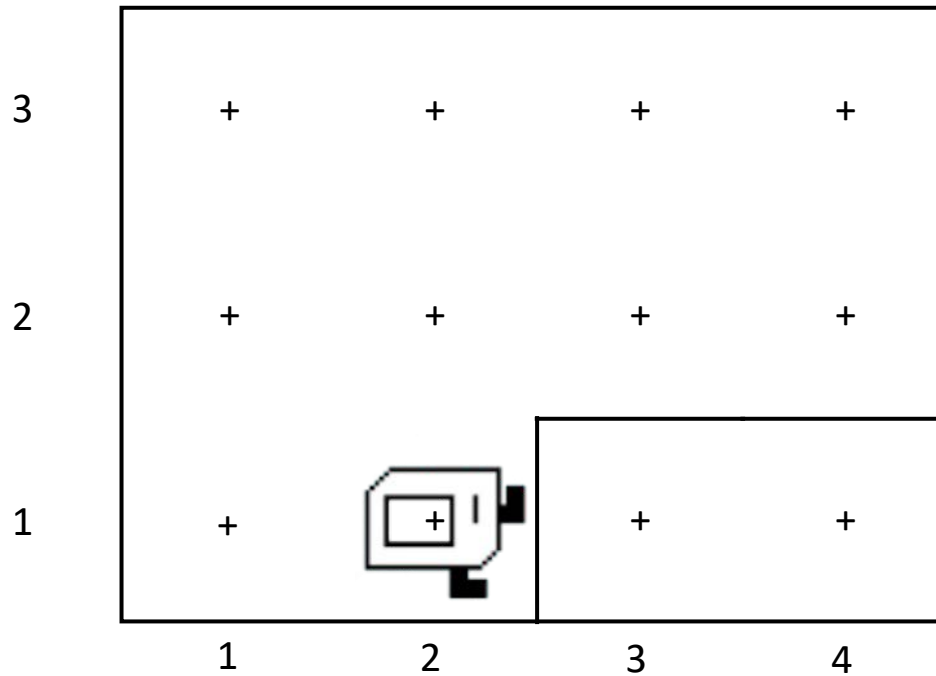


```
pickBeeper ( ) ;
```

# pickBeeper();



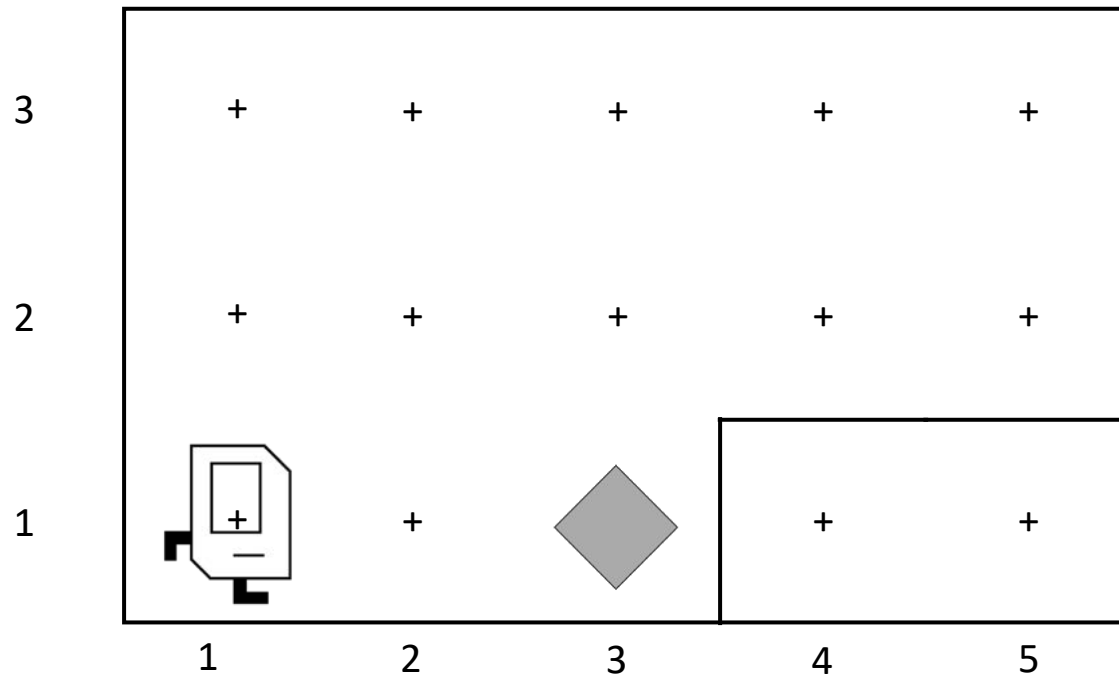
# pickBeeper( ) ;



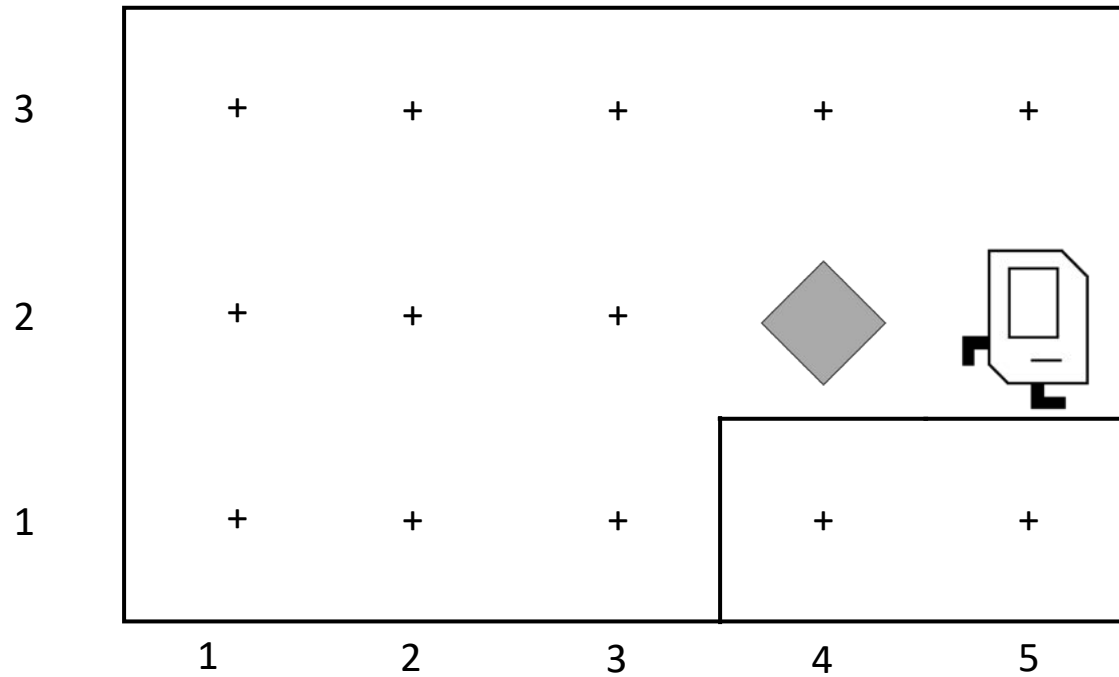
Make Sense?



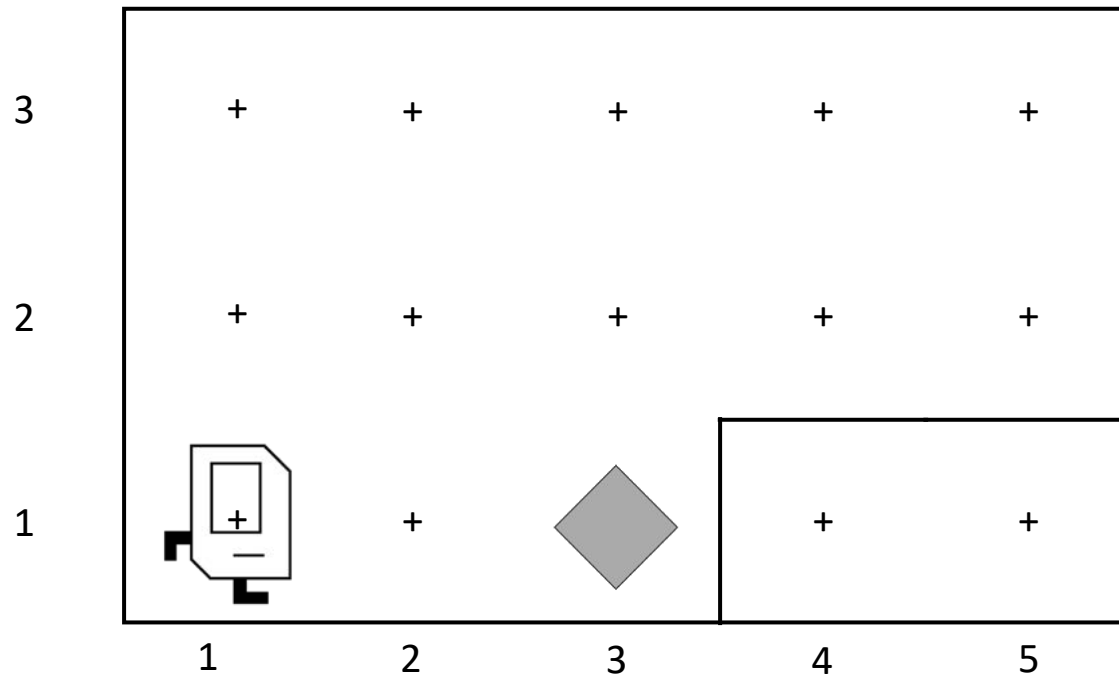
# First Challenge



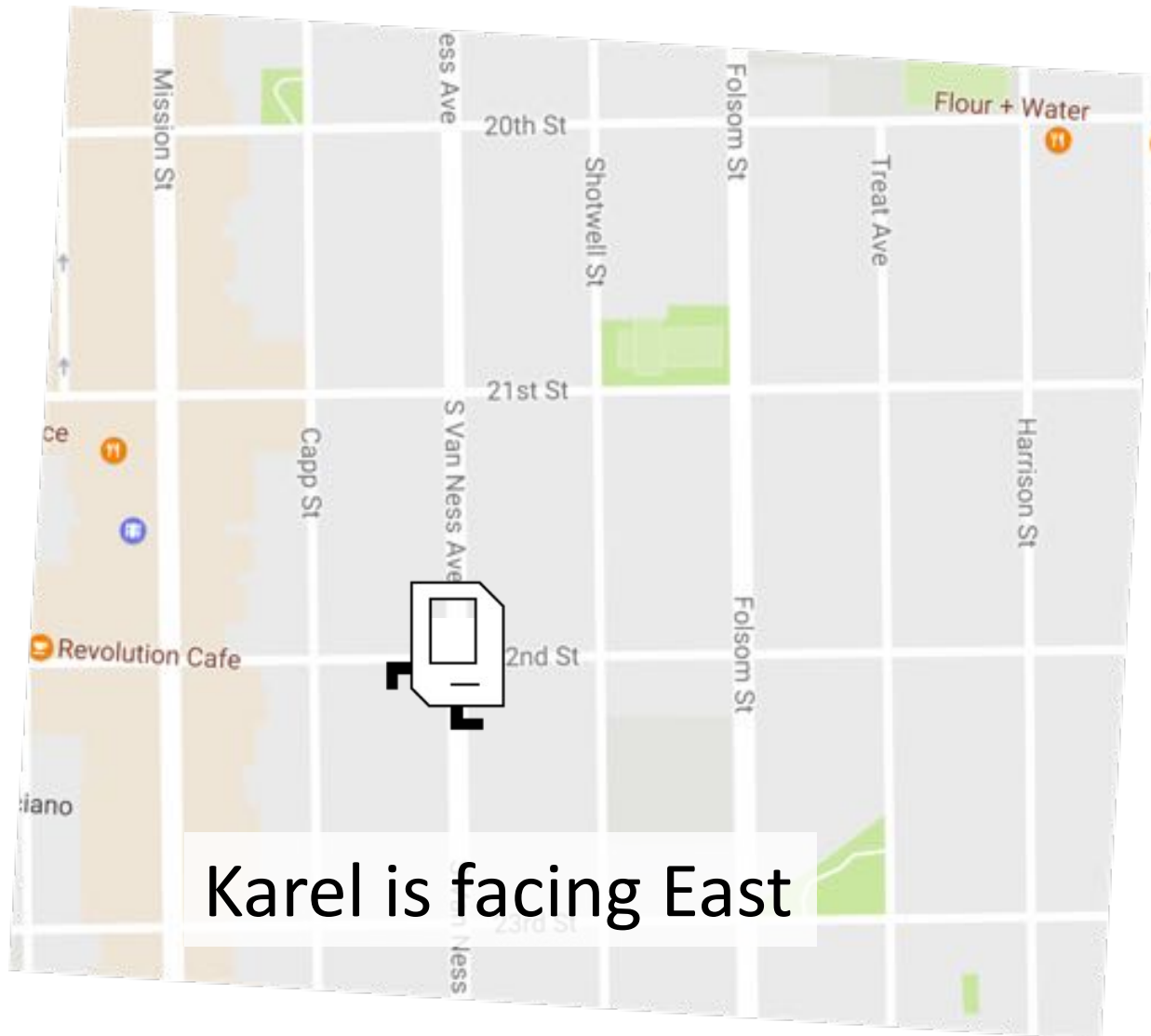
# First Challenge



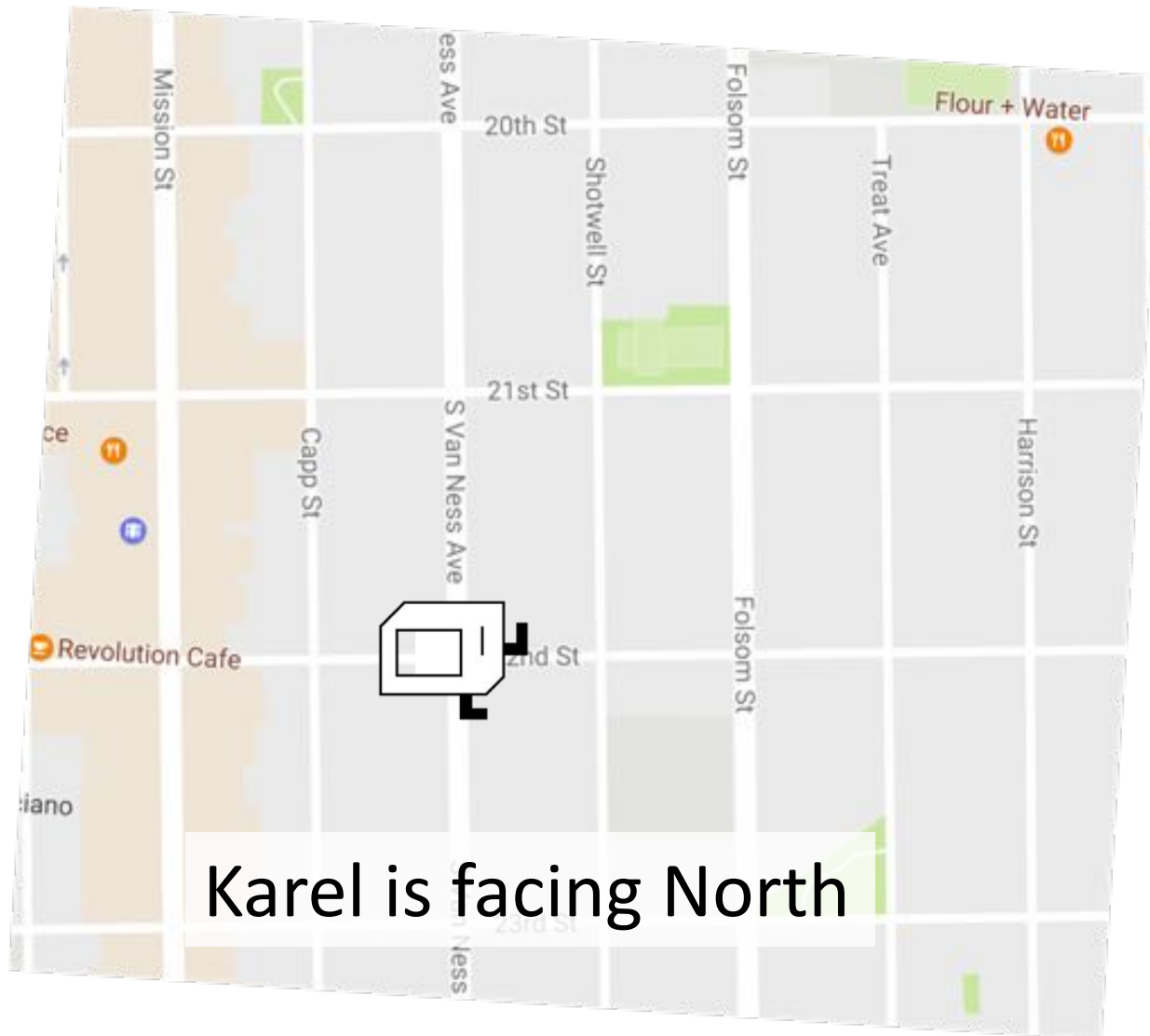
# Bird's Eye View



# Bird's Eye View



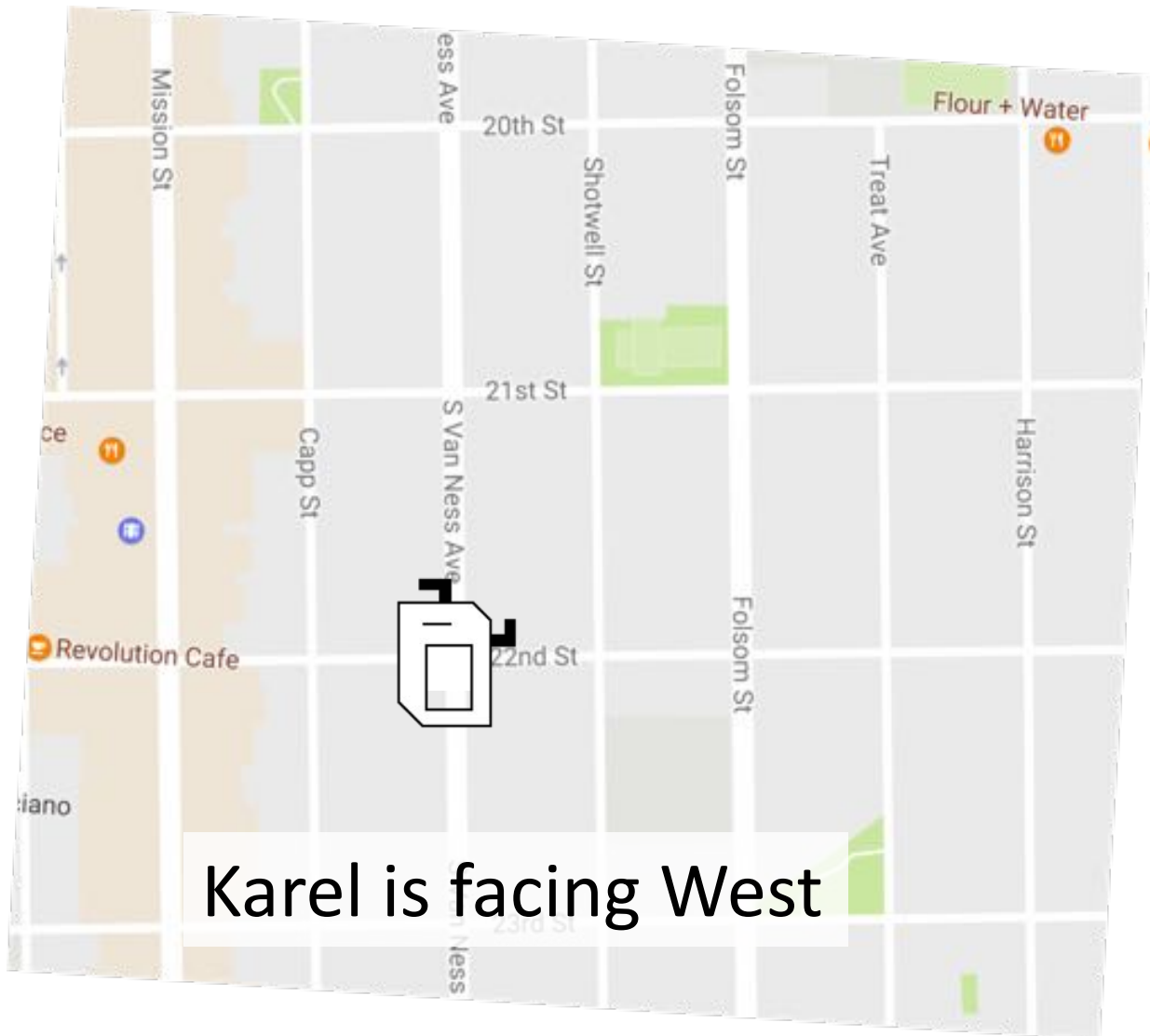
# Turn Left



Karel is facing North



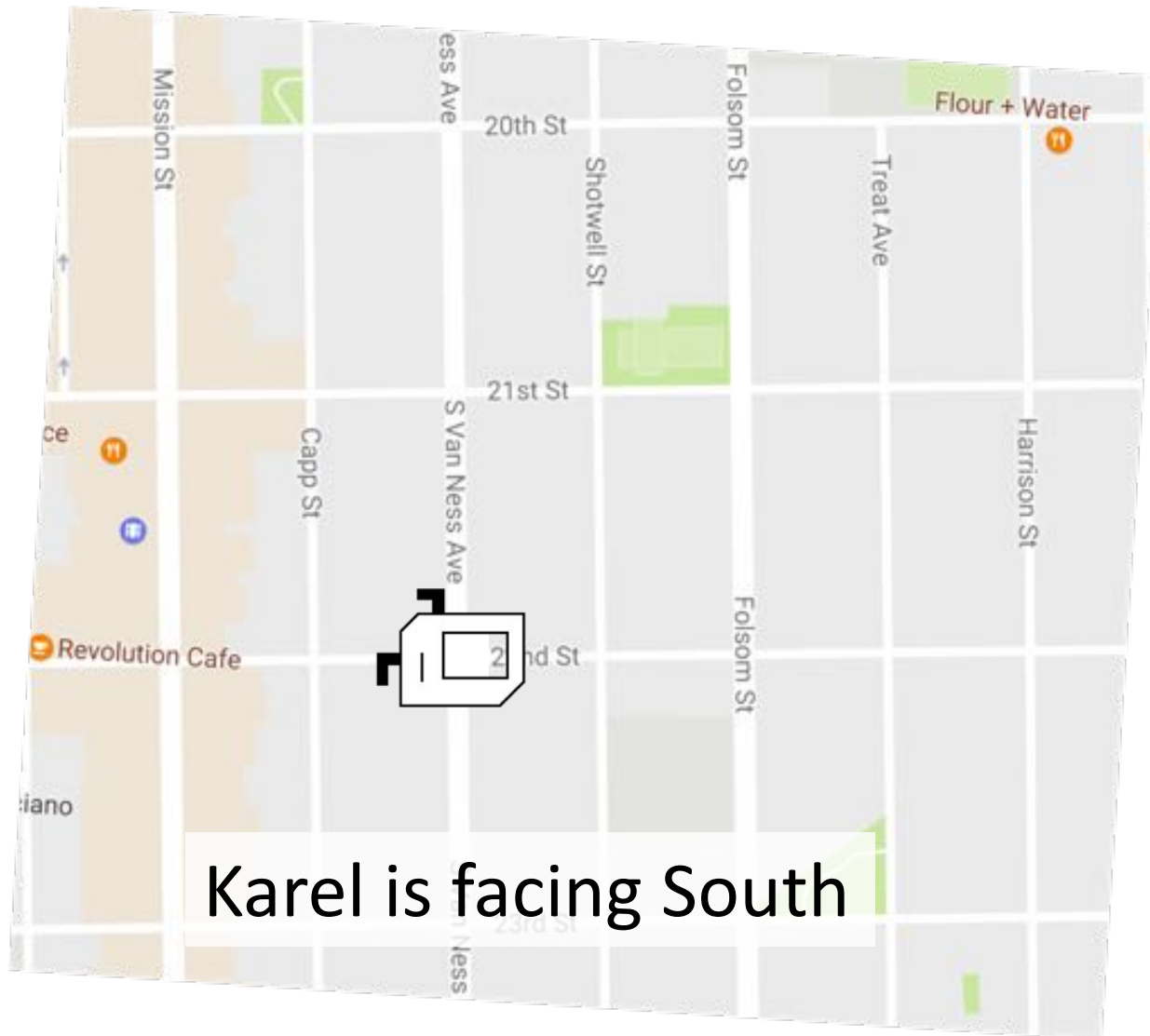
# Turn Left



Karel is facing West



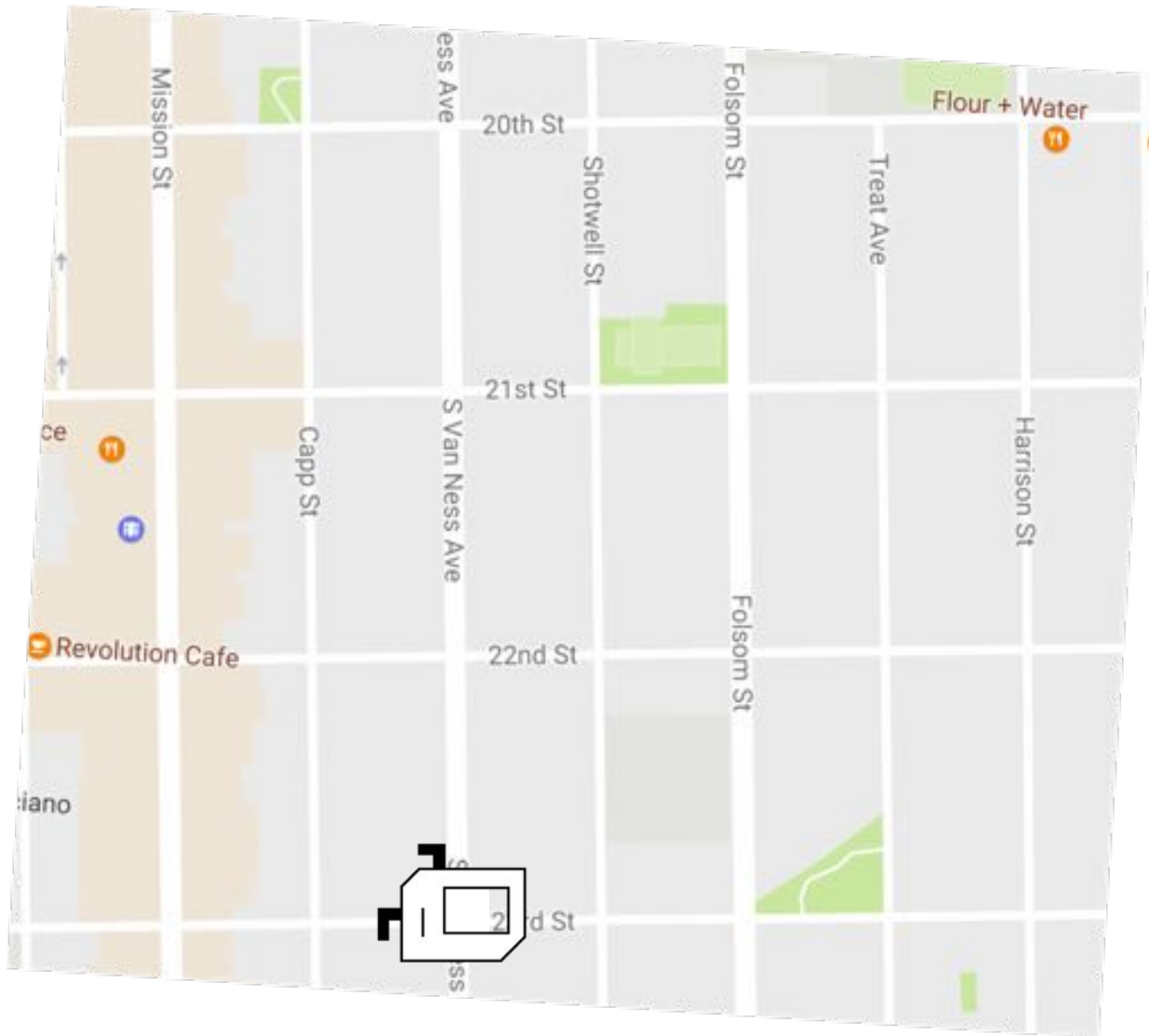
# Turn Left



Karel is facing South



# Move

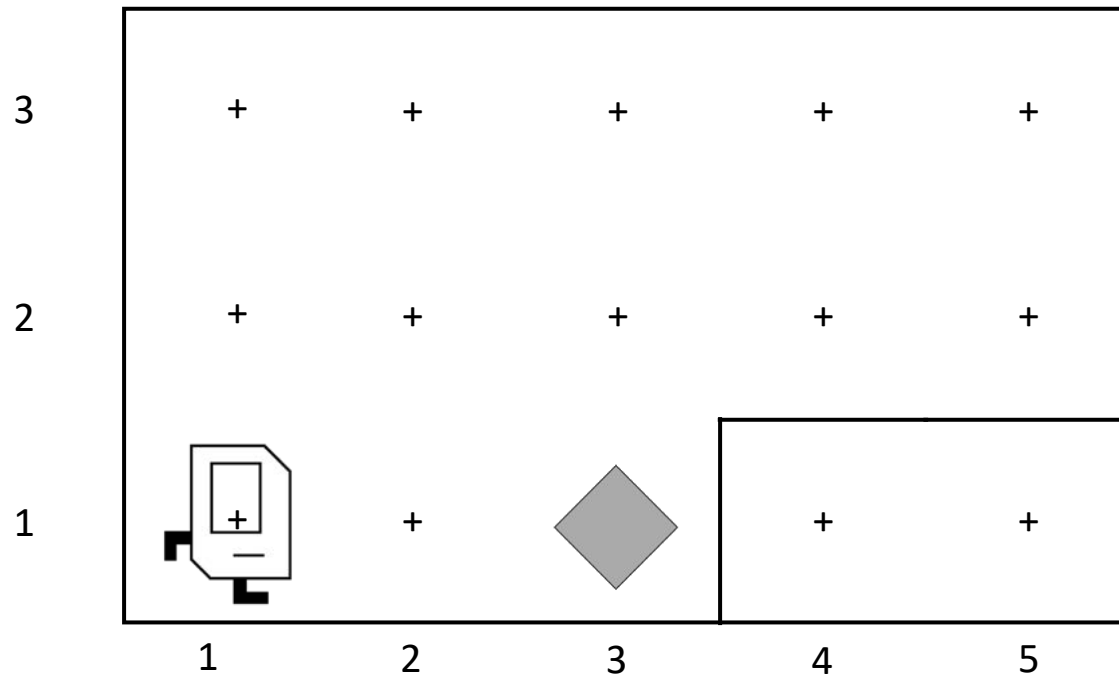


Piech, CS106A, Stanford University

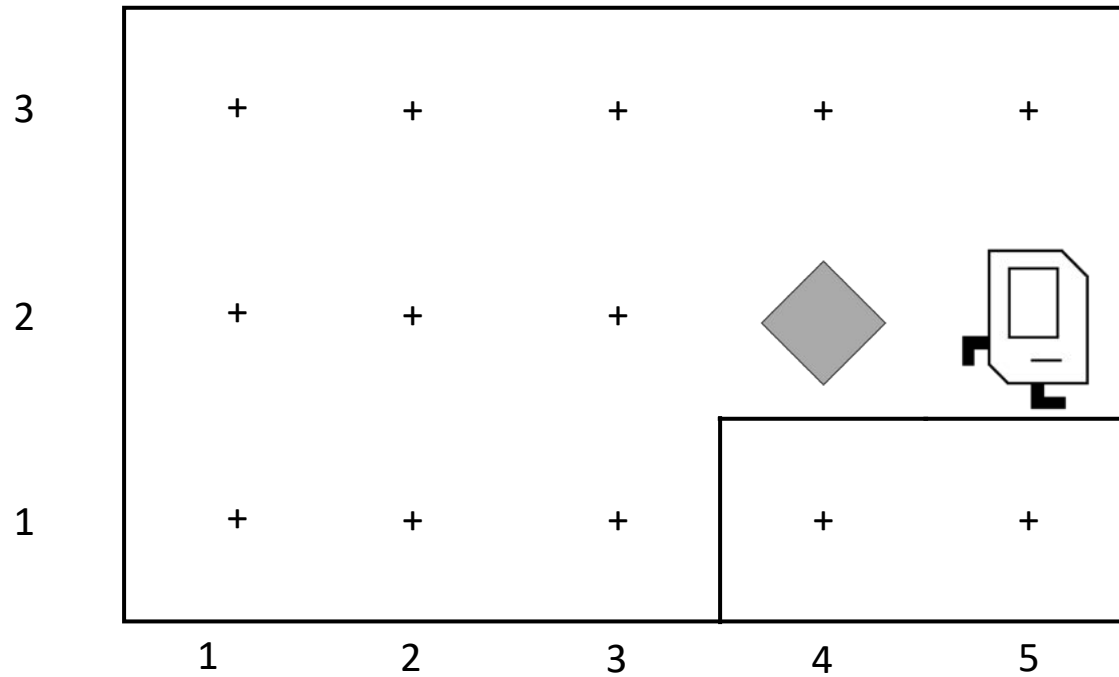




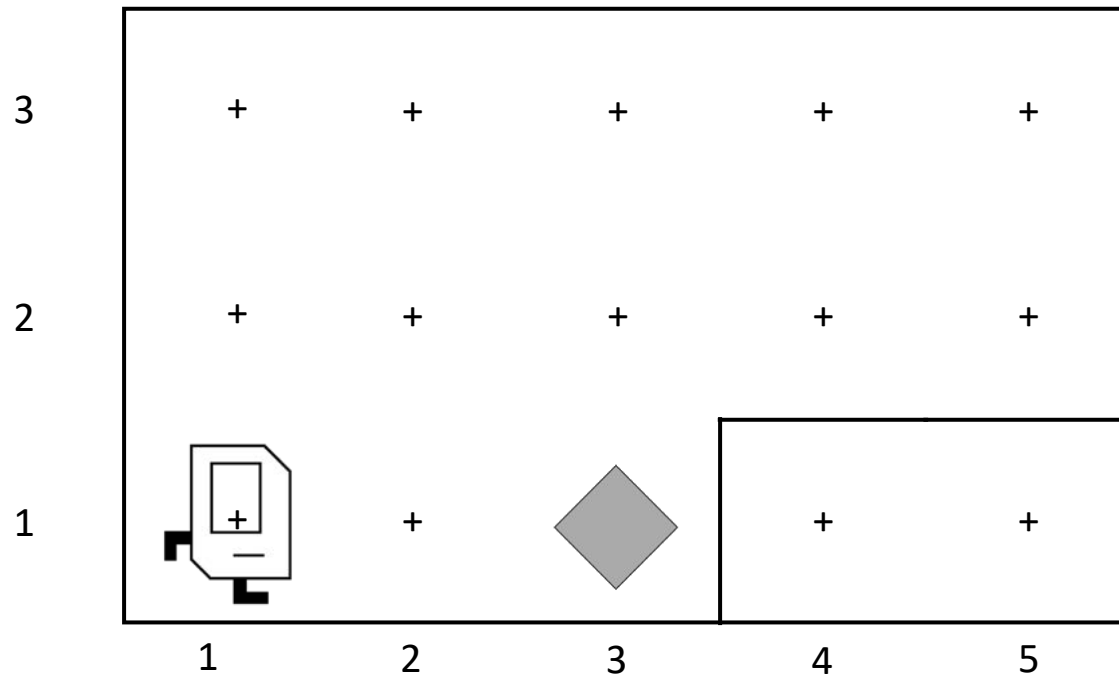
# First Challenge



# First Challenge



# First Challenge



# Learn By Doing



# eclipse



# Method Definition

```
private void name() {  
    method statements  
}
```

This adds a new  
command to Karel's  
vocabulary



# Anatomy of a Program

Import Packages

Program



# Anatomy of a Program

Import Packages

```
public class OurKarelProgram extends Karel {
```

```
}
```





# Anatomy of a Program

Import Packages

```
public class OurKarelProgram extends Karel {
```

run method

helper methods

```
}
```



# Anatomy of a Program

## Import Packages

```
public class OurKarelProgram extends Karel {  
  
    public void run() {  
        move();  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
  
    helper methods  
  
}
```



# Anatomy of a Program

## Import Packages

```
public class OurKarelProgram extends Karel {  
  
    public void run() {  
        move();  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
  
    private void turnRight() {  
        turnLeft();  
        turnLeft();  
        turnLeft();  
    }  
}
```



# Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```



# Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```

This piece of the program's *source code* is called a *method*.



# Anatomy of a Program

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

```
public class OurKarelProgram extends Karel {
```

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

```
        move();
```

```
        pickBeeper();
```

```
        move();
```

```
        turnLeft();
```

```
        move();
```

```
        turnRight();
```

```
        move();
```

```
        putBeeper();
```

```
        move();
```

```
    }
```

```
    private void turnRight() {
```

```
        turnLeft();
```

```
        turnLeft();
```

```
        turnLeft();
```

```
    }
```

```
}
```

This line of code gives the *name* of the method (here, run)



# Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```

This line of code gives the *name* of the method (here, turnRight)



# Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```

This is called a *code block*





# Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```

This is called a *code block*



# Anatomy of a Program

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

```
public class OurKarelProgram extends Karel {
```

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

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

```
    }
```

```
    private void turnRight() {
```

```
        turnLeft();  
        turnLeft();  
        turnLeft();
```

```
    }
```

```
}
```

This is also called a  
*code block*



# Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```



# Anatomy of a Program

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

```
public class OurKarelProgram extends Karel {
```

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

```
        move();
```

```
        pickBeeper();
```

```
        move();
```

```
        turnLeft();
```

```
        move();
```

```
        turnRight();
```

```
        move();
```

```
        putBeeper();
```

```
        move();
```

```
    }
```

```
    private void turnRight() {
```

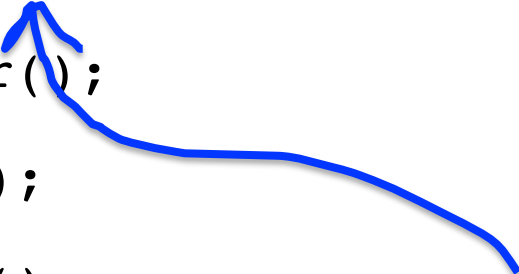
```
        turnLeft();
```

```
        turnLeft();
```

```
        turnLeft();
```

```
    }
```

```
}
```



The run method is “public” so that  
Eclipse can call it.



# Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```

The turnRight method is “private” to indicate it is only visible to our current program.



Why Study CS?

# Joy of Building



# Interdisciplinary



Piech, CS106A, Stanford University





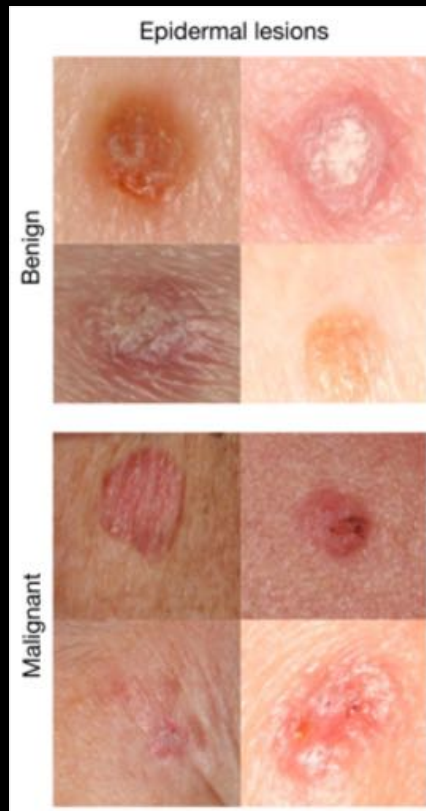
# Closest Thing To Magic



# Now is the Time



# Now is the Time

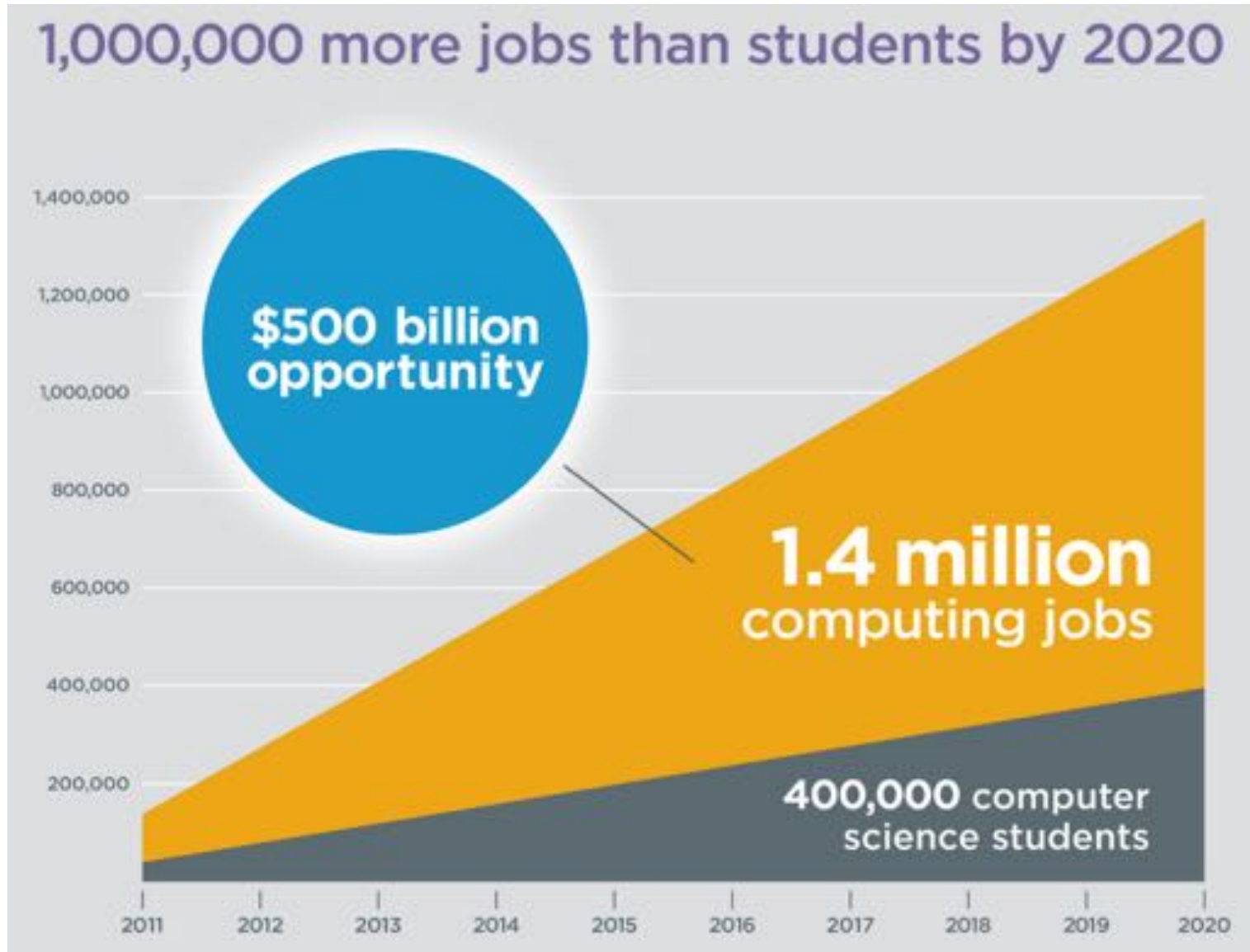


A machine learning algorithm performs **better than** the best dermatologists.

Developed this year, at Stanford.

Esteva, Andre, et al. "Dermatologist-level classification of skin cancer with deep neural networks." *Nature* 542.7639 (2017): 115-118.

# Oh and Its Useful



# Everyone is Welcome



The End



The End?