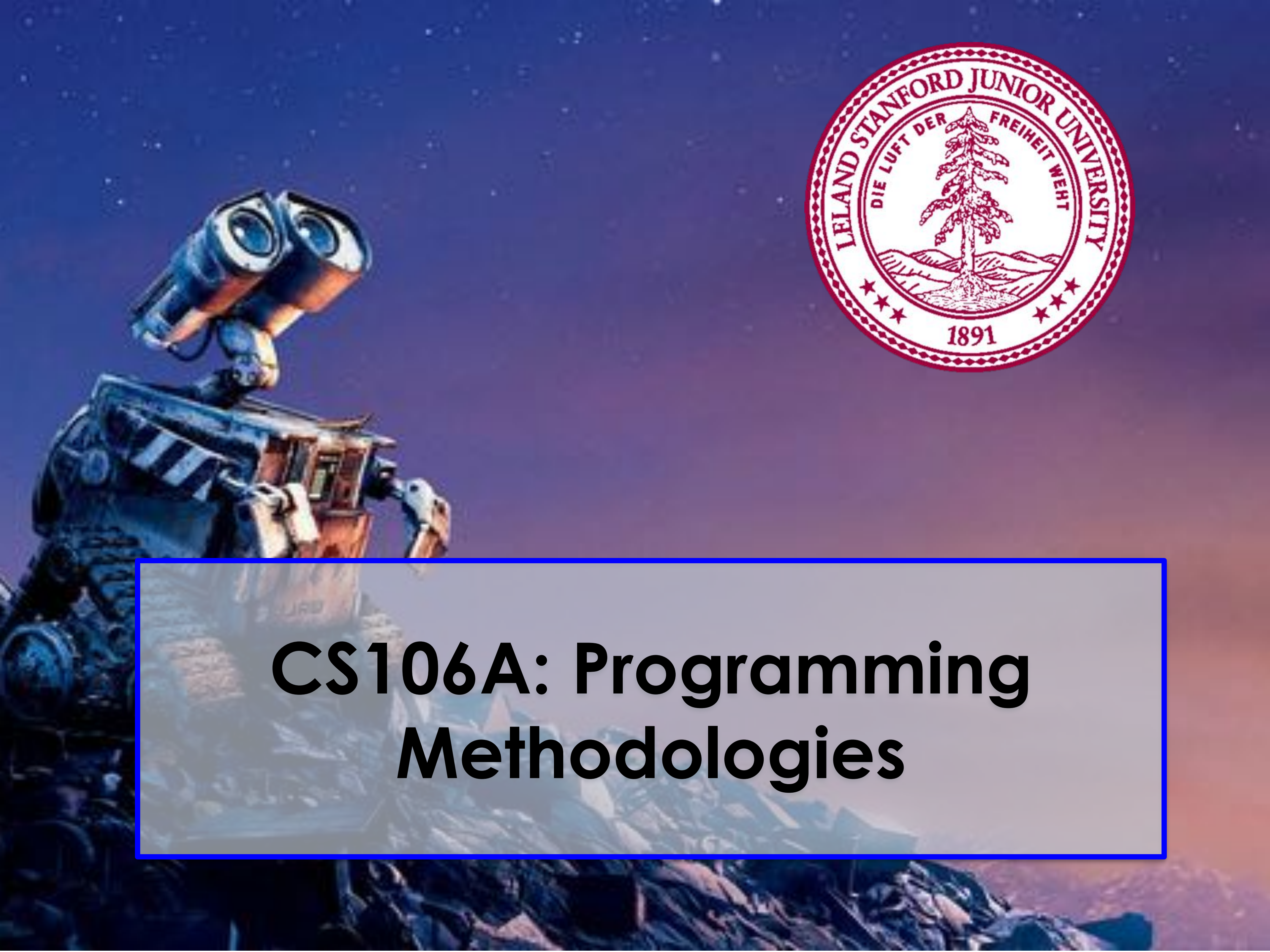


CS106A: Programming Methodologies

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

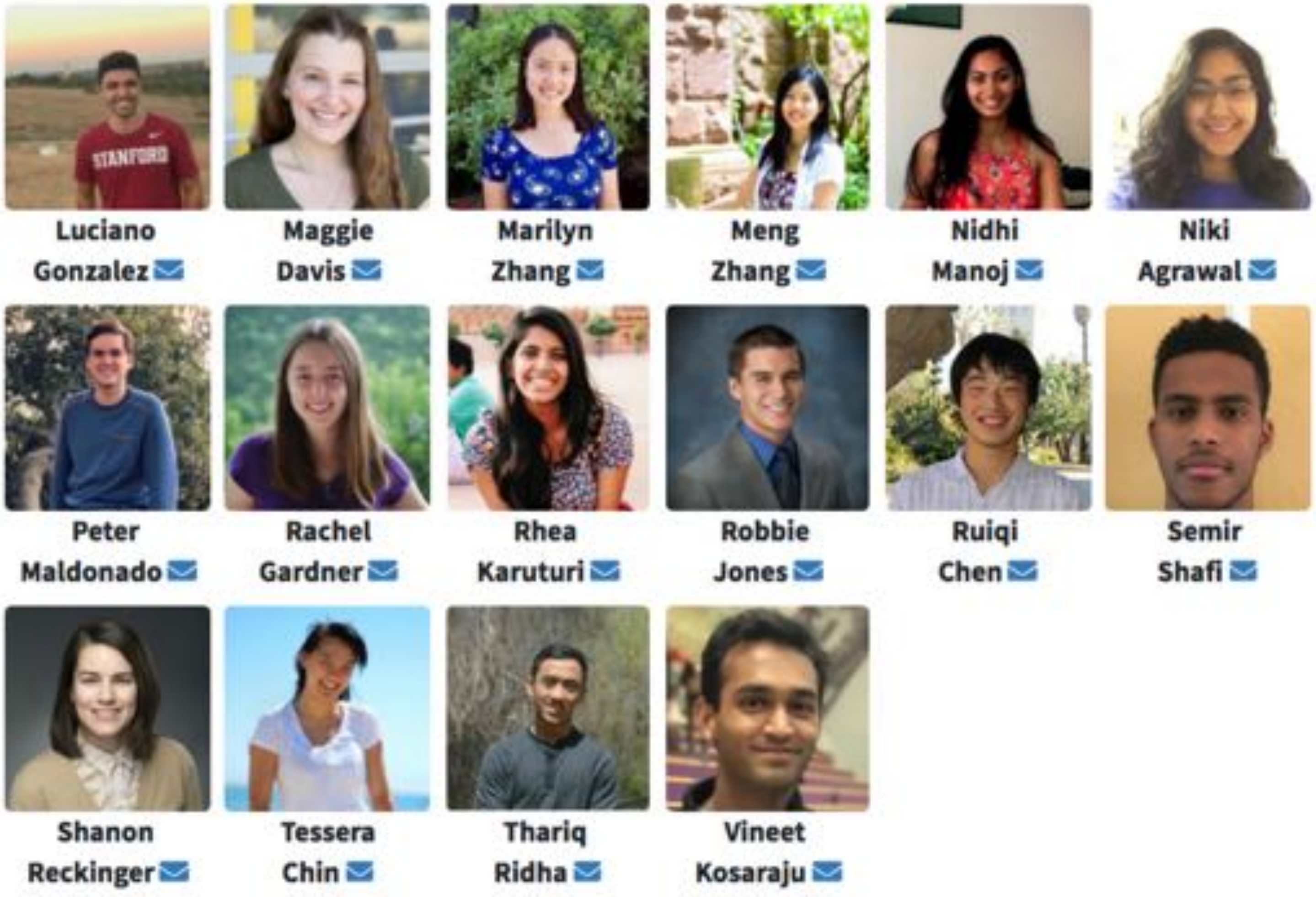
The problem I really want to solve is to make high quality more education accessible

Chris Piech

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

Head TA: Brahm Capoor

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 ✉

* Actually last year's section leaders

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

Office Hours

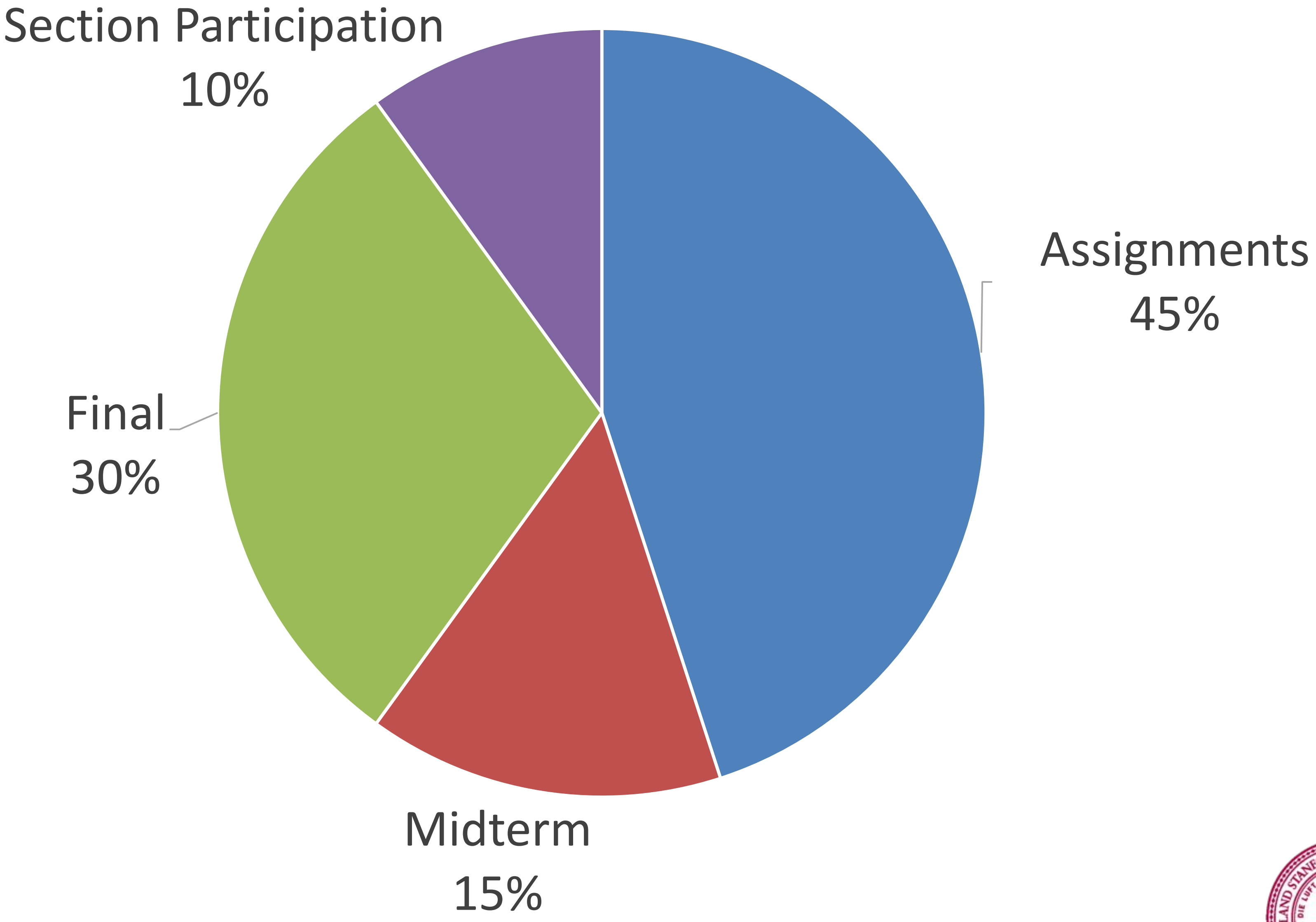
LaIR: evenings Sunday through Thursday
(starting next Sunday)

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



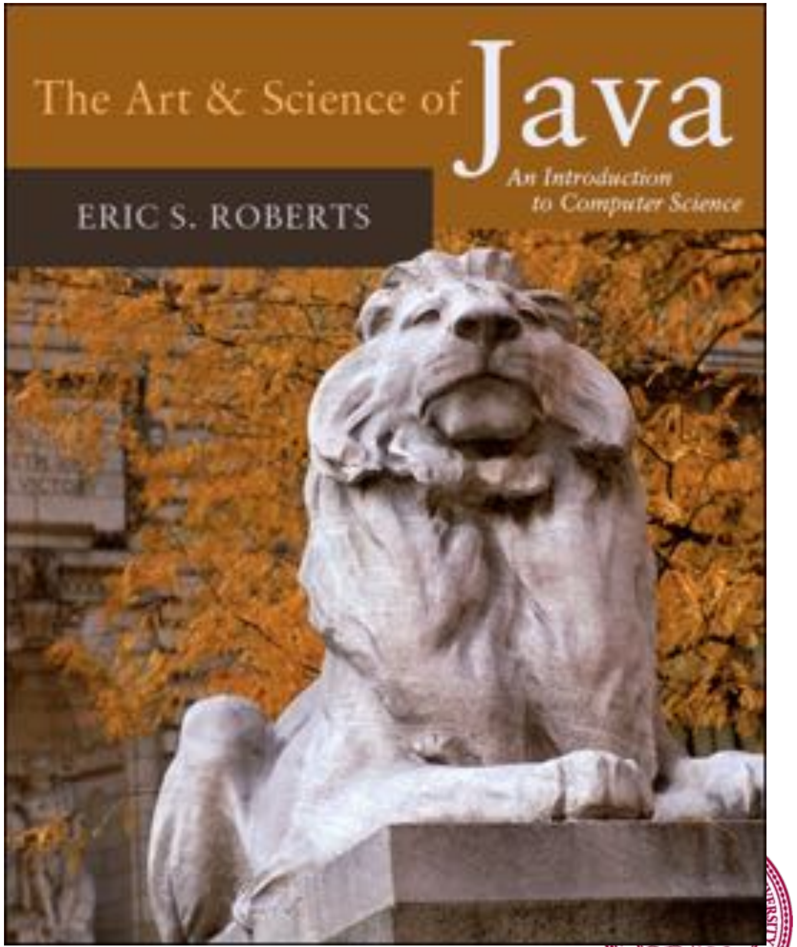
*Two free late days

Optional Contest

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

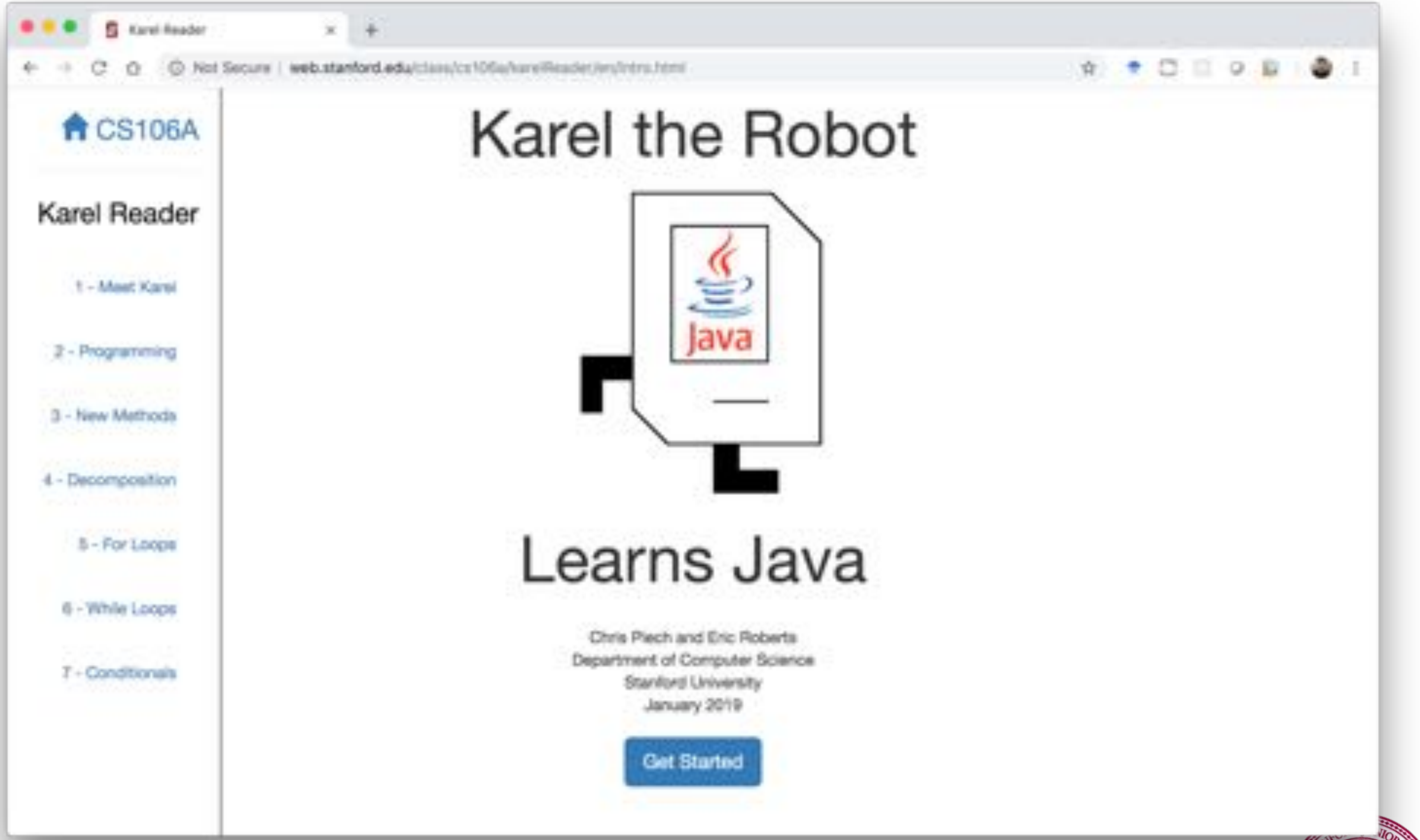


The Art & Science of Java
An Introduction to Computer Science
ERIC S. ROBERTS

Karel the Robot Learns Java

- First week of material

Online Karel Reader

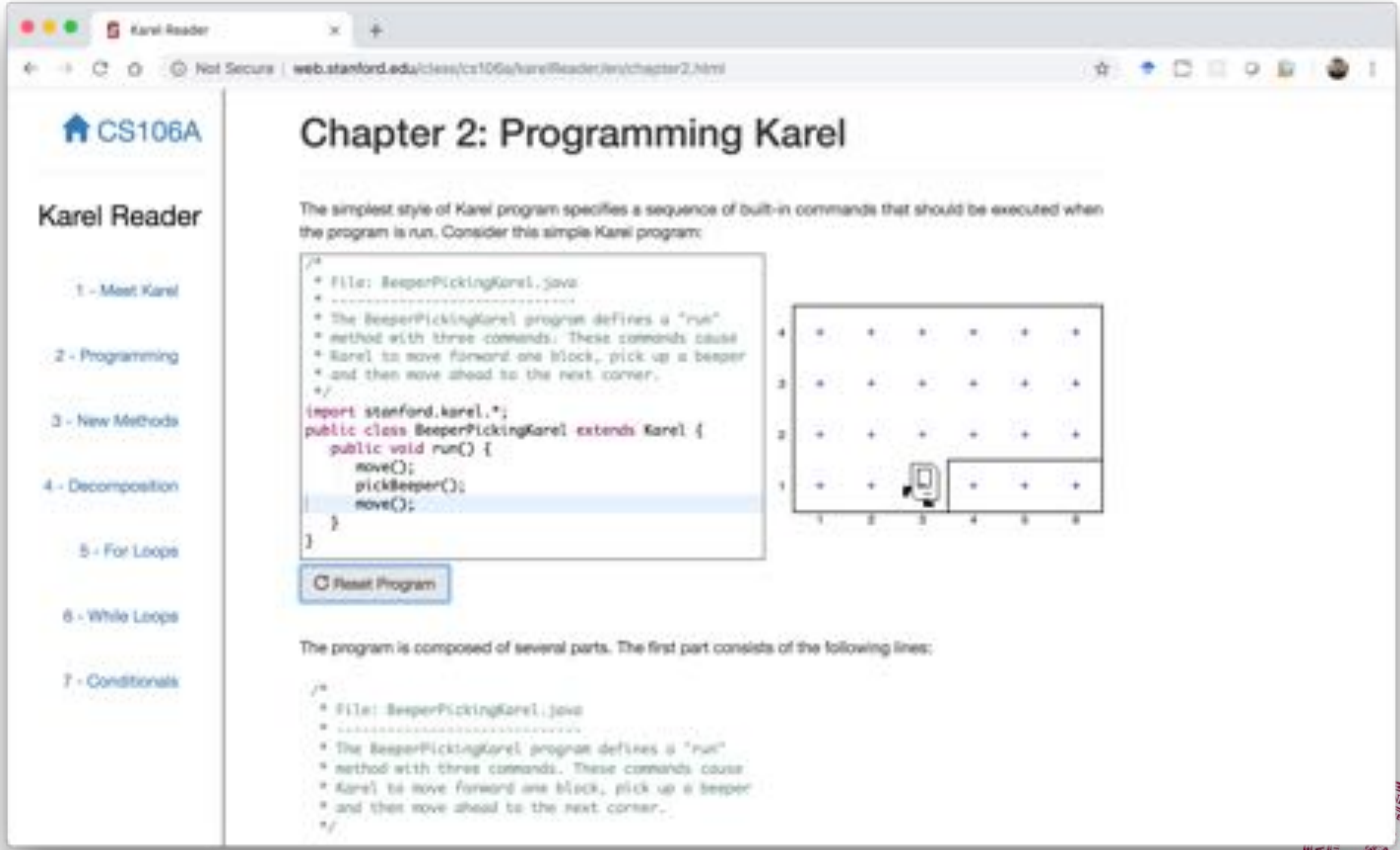


Karel the Robot

Learns Java

Chris Piech and Eric Roberts
Department of Computer Science
Stanford University
January 2019

Online Karel Reader



Karel Reader
Not Secure | web.stanford.edu/class/cs106a/karelReader/en/chapter2.html
CS106A

Karel Reader

- 1 - Meet Karel
- 2 - Programming
- 3 - New Methods
- 4 - Decomposition
- 5 - For Loops
- 6 - While Loops
- 7 - Conditionals

Chapter 2: Programming Karel

The simplest style of Karel program specifies a sequence of built-in commands that should be executed when the program is run. Consider this simple Karel program:

```
/*
 * File: BeeperPickingKarel.java
 * -----------------------------
 * The BeeperPickingKarel program defines a "run"
 * method with three commands. These commands cause
 * Karel to move forward one block, pick up a beeper
 * and then move ahead to the next corner.
 */
import stanford.karel.*;
public class BeeperPickingKarel extends Karel {
   public void run() {
      move();
      pickBeeper();
      move();
   }
}
```

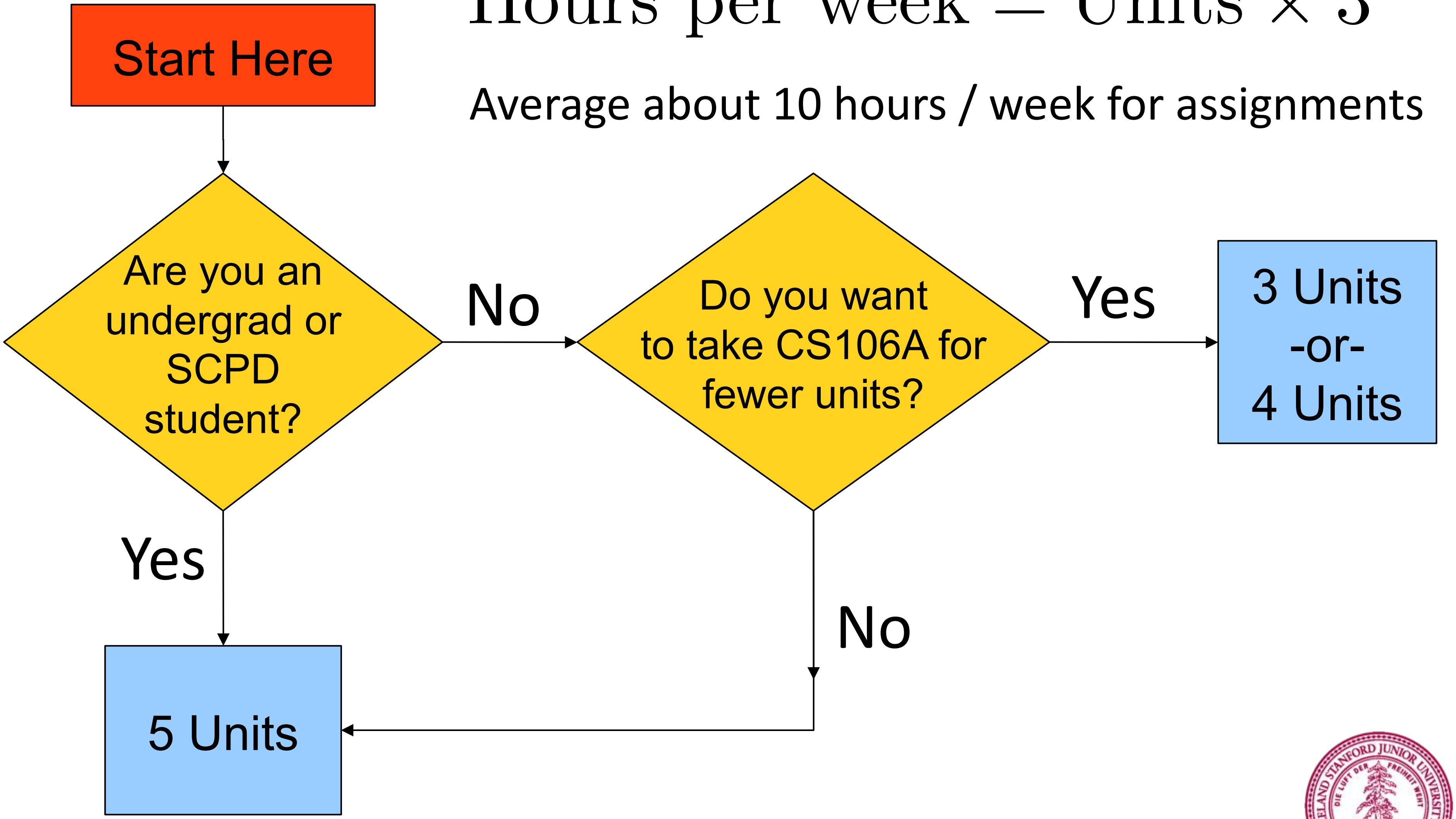
The program is composed of several parts. The first part consists of the following lines:

```
/*
 * File: BeeperPickingKarel.java
 * -----------------------------
 * The BeeperPickingKarel program defines a "run"
 * method with three commands. These commands cause
 * Karel to move forward one block, pick up a beeper
 * and then move ahead to the next corner.
 */
```

CS106A Units

$$\text{Hours per week} = \text{Units} \times 3$$

Average about 10 hours / week for assignments


Start Here
Are you an undergrad or SCPD student?
No
Do you want to take CS106A for fewer units?
Yes
3 Units -or- 4 Units
Yes
No
5 Units

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

Consumer Applications

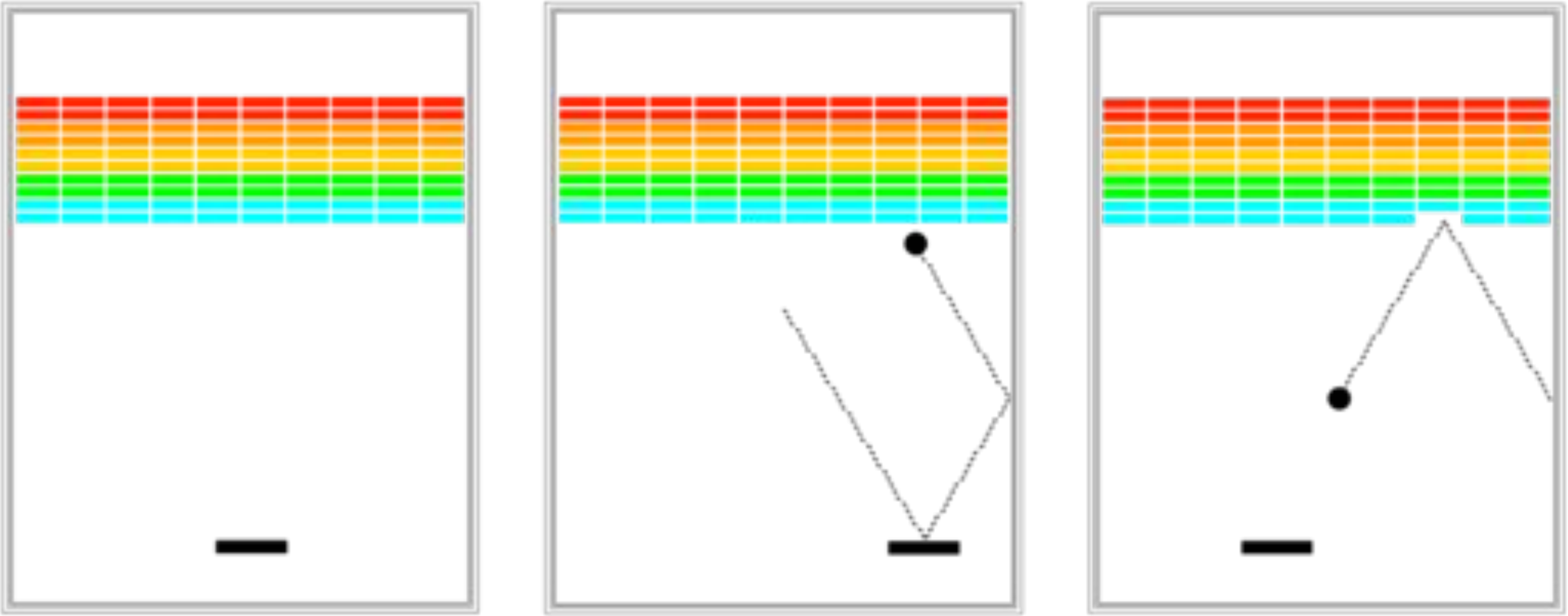
Autonomous Surgery

(c) 2012 Intuitive Surgical, Inc.

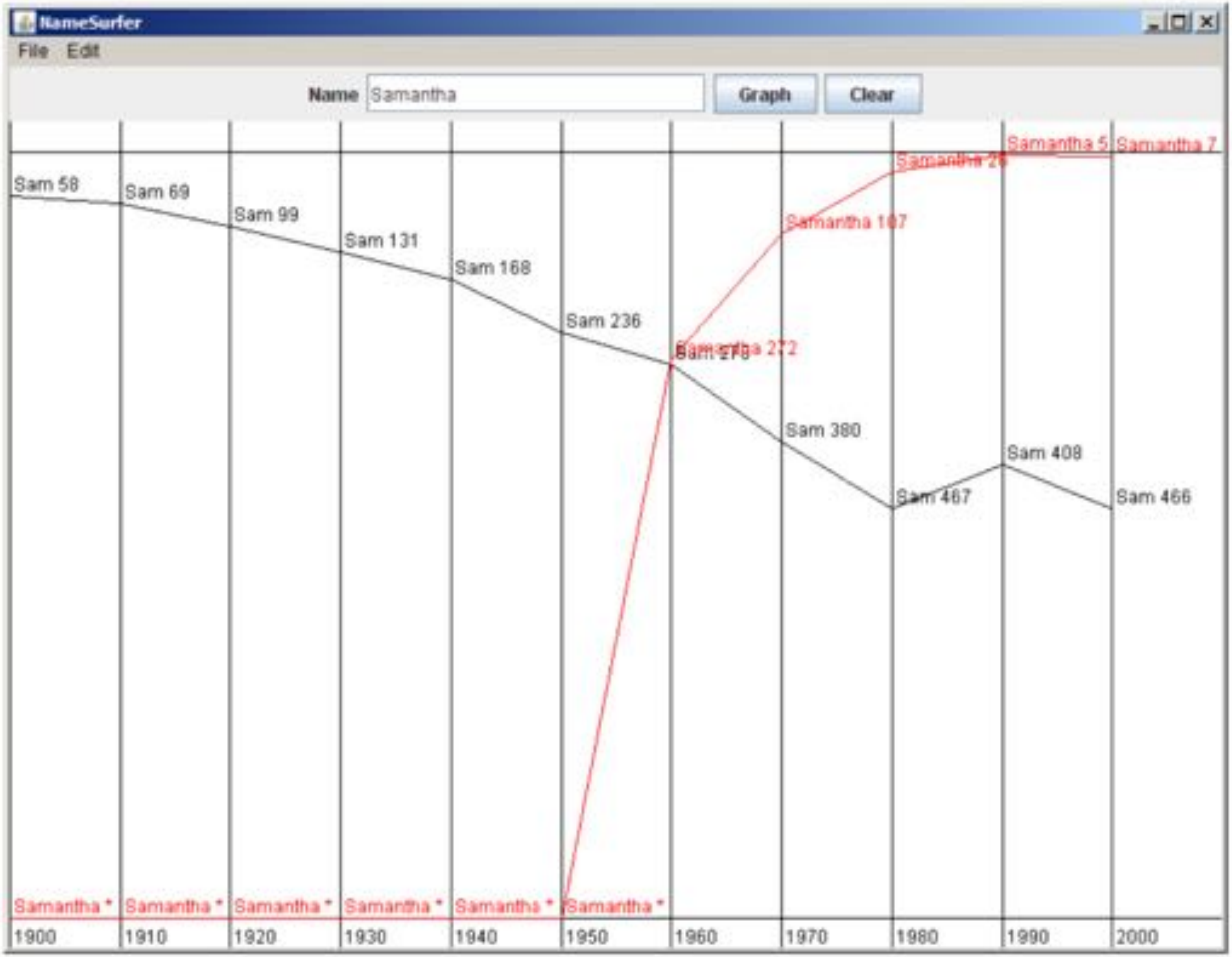
Self Driving Car

If only we could program self driving cars…

Graphical Games

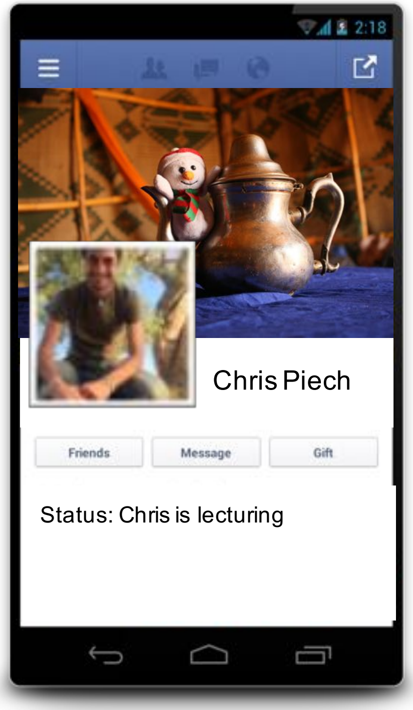


Data Visualization



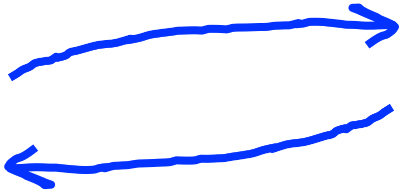
Name Samantha

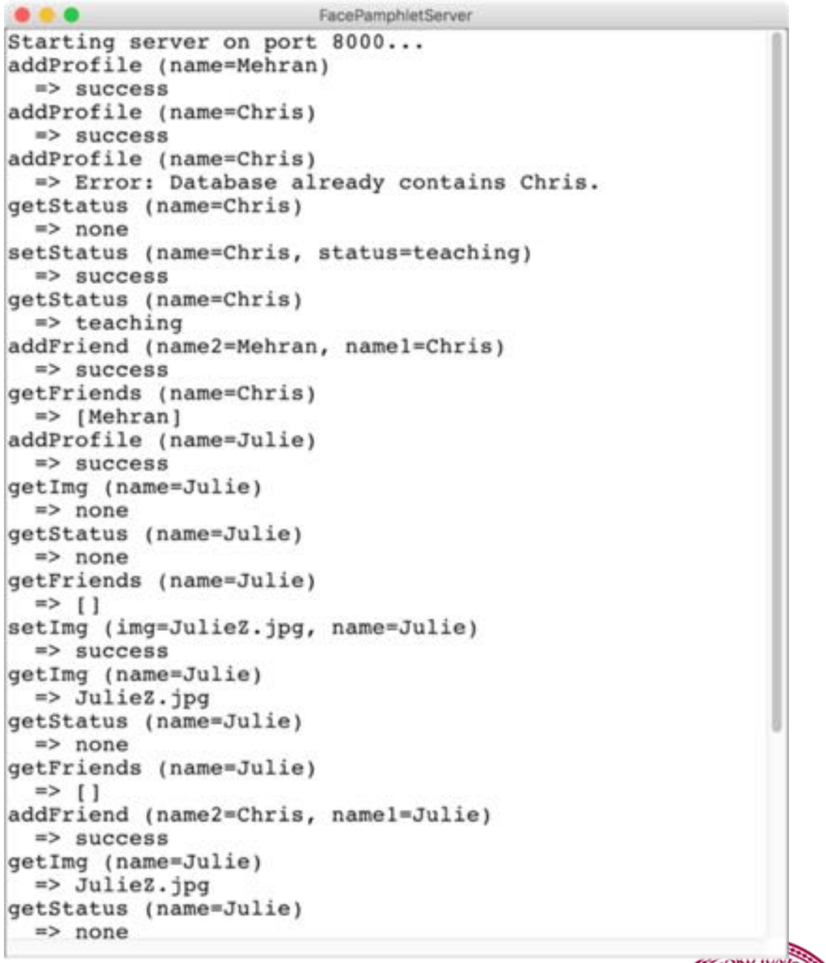
Internet Applications



Chris Piech

Status: Chris is lecturing





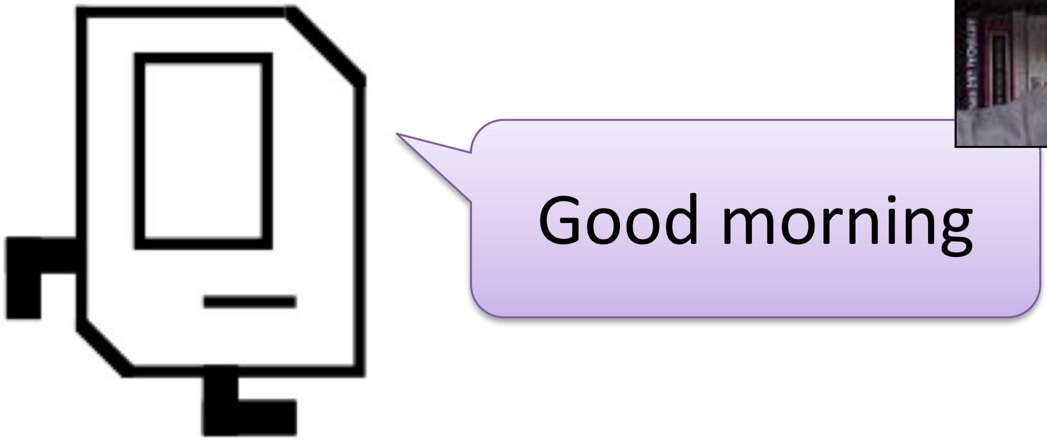
```
Starting server on port 8000...
addProfile (name=Mehran)
  => success
addProfile (name=Chris)
  => success
addProfile (name=Chris)
  => Error: Database already contains Chris.
getStatus (name=Chris)
  => none
setStatus (name=Chris, status=teaching)
  => success
getStatus (name=Chris)
  => teaching
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*

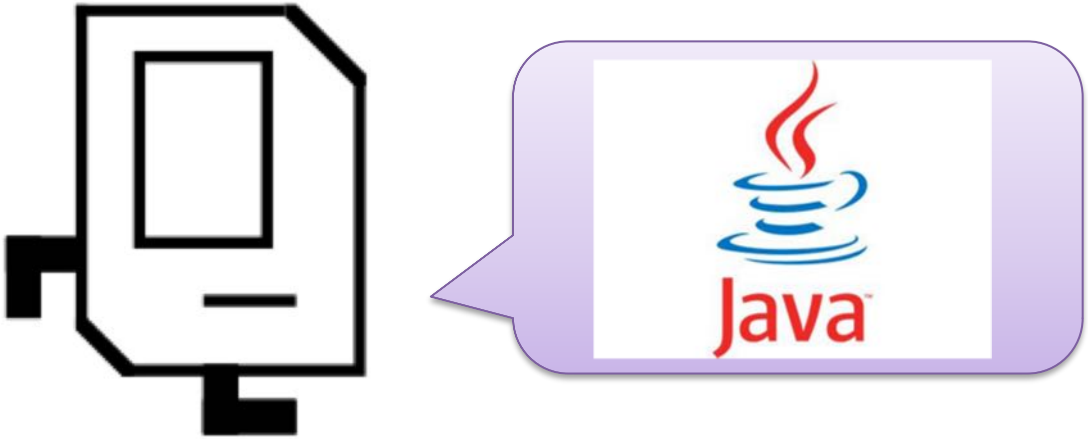
Strive for Everyone to Succeed

Lets Get Started

Meet Karel the Robot

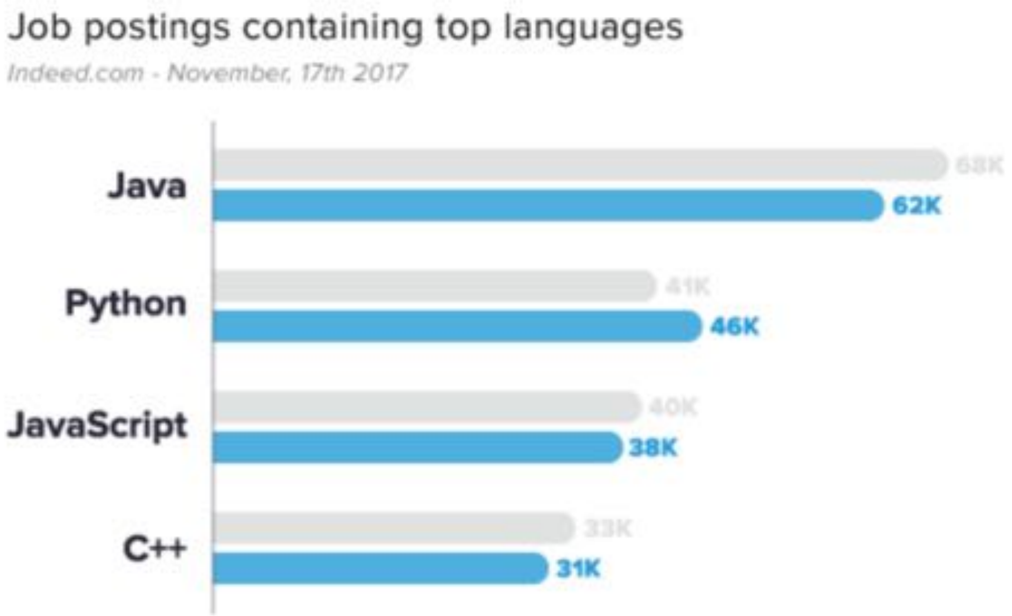


Karel Speaks Java



Why Java?

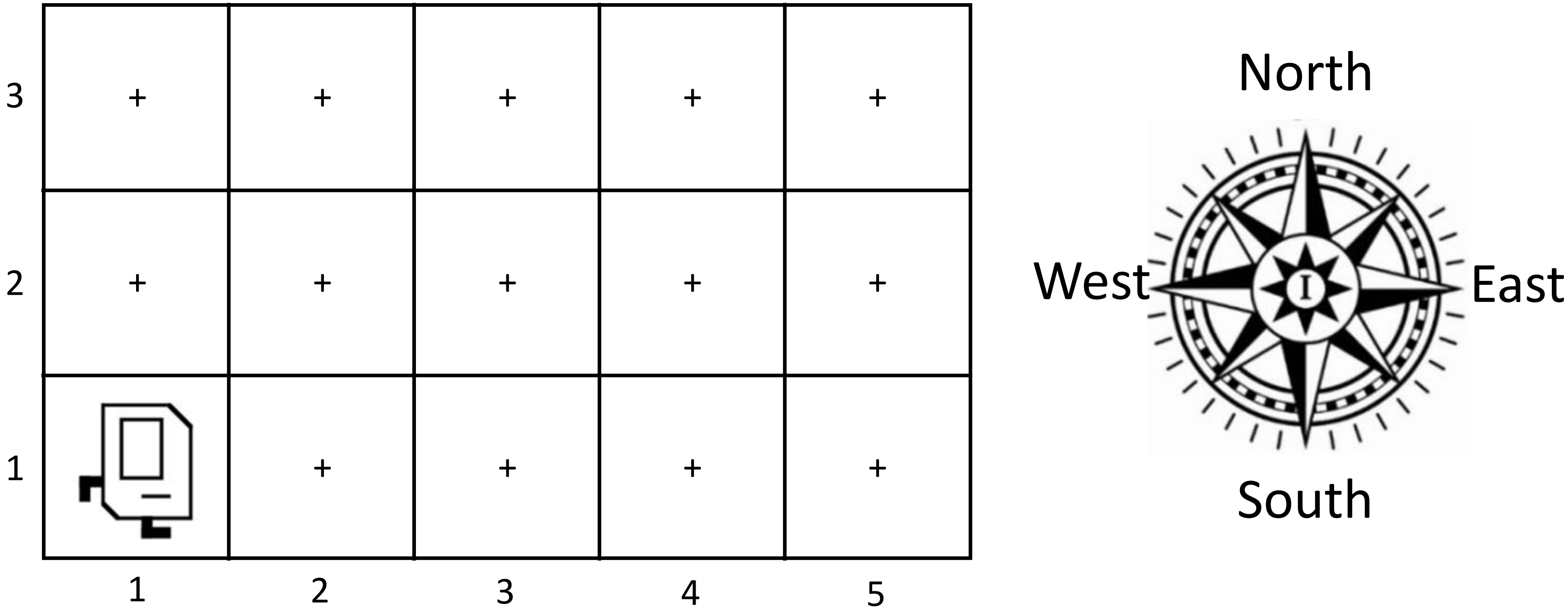
1



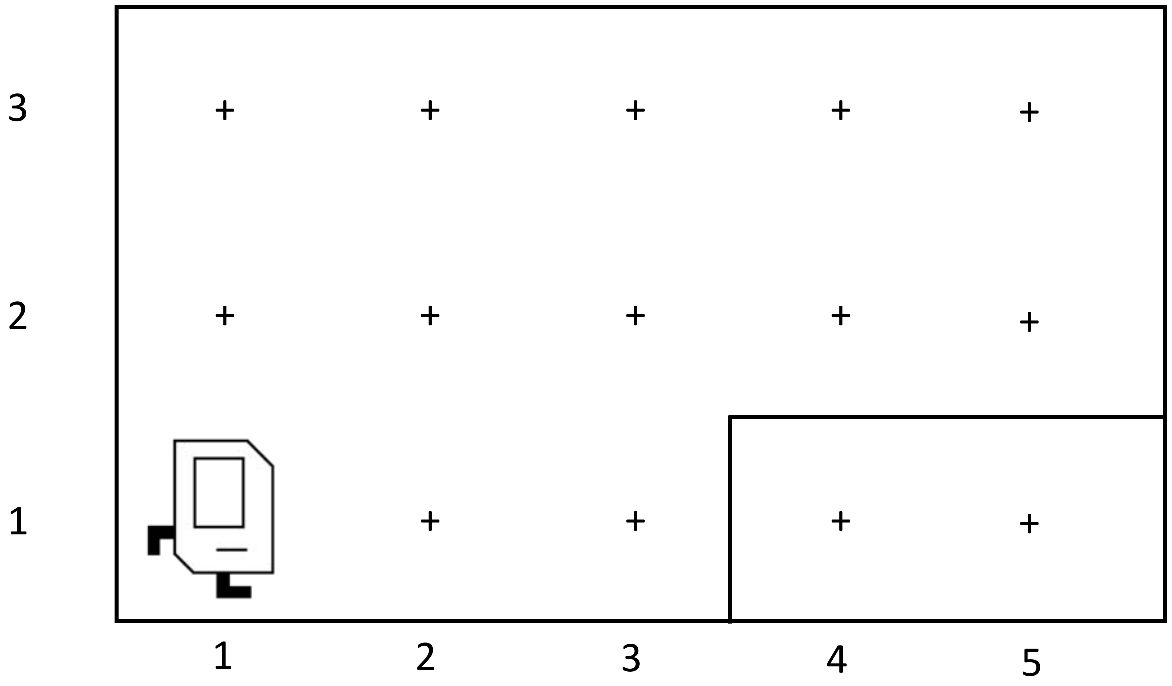
2

http://www.codingdojo.com/blog/7-most-in-demand-programming-languages-of-2018/

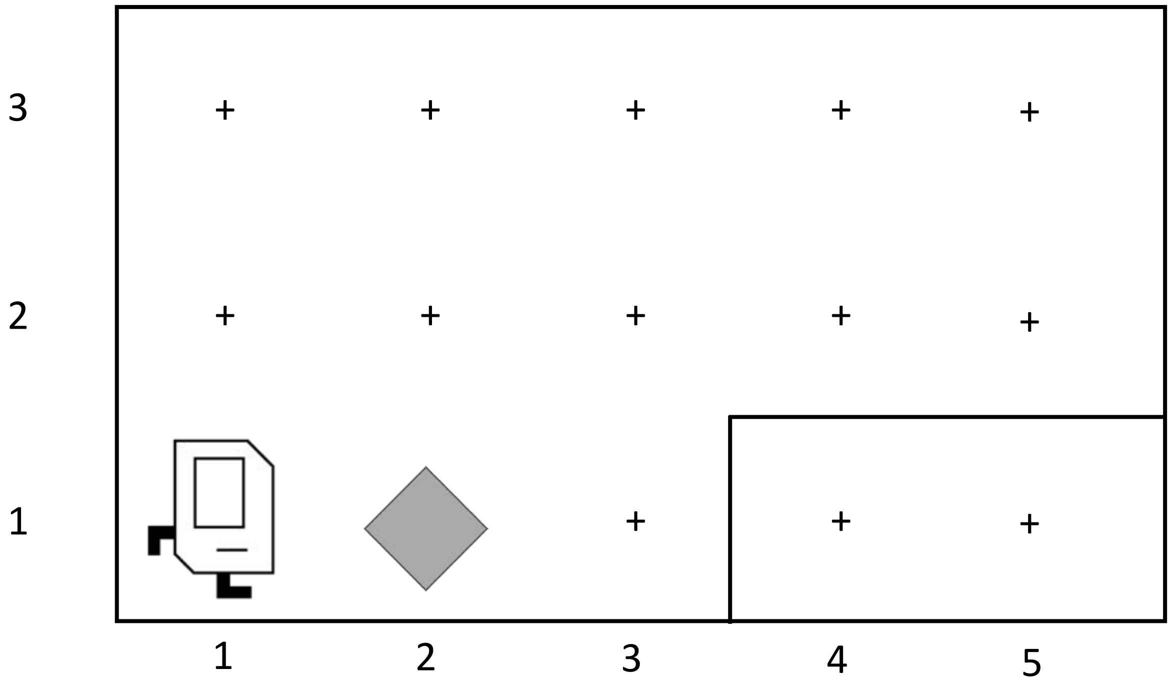
Karel’s World



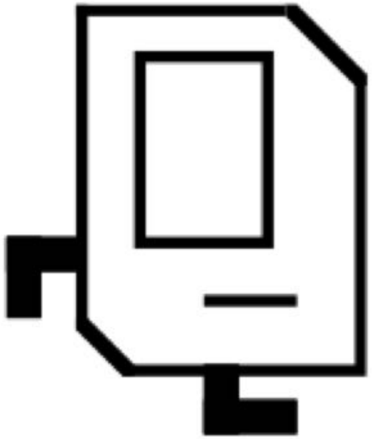
Walls



Beepers



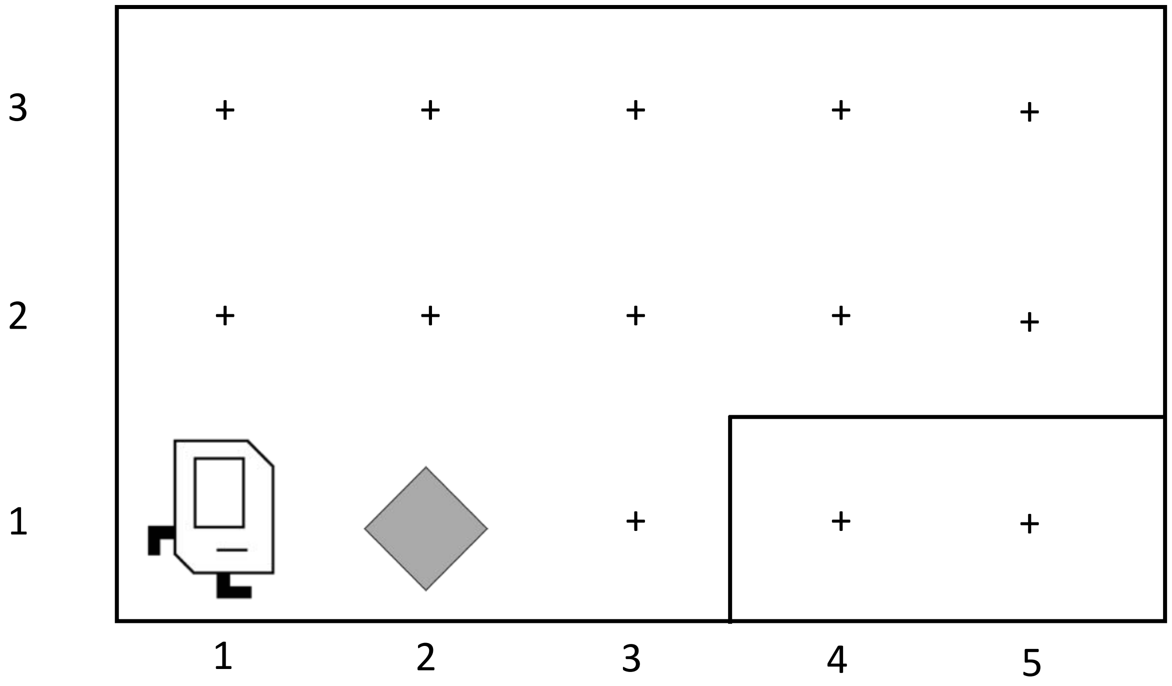
Knows Four Commands



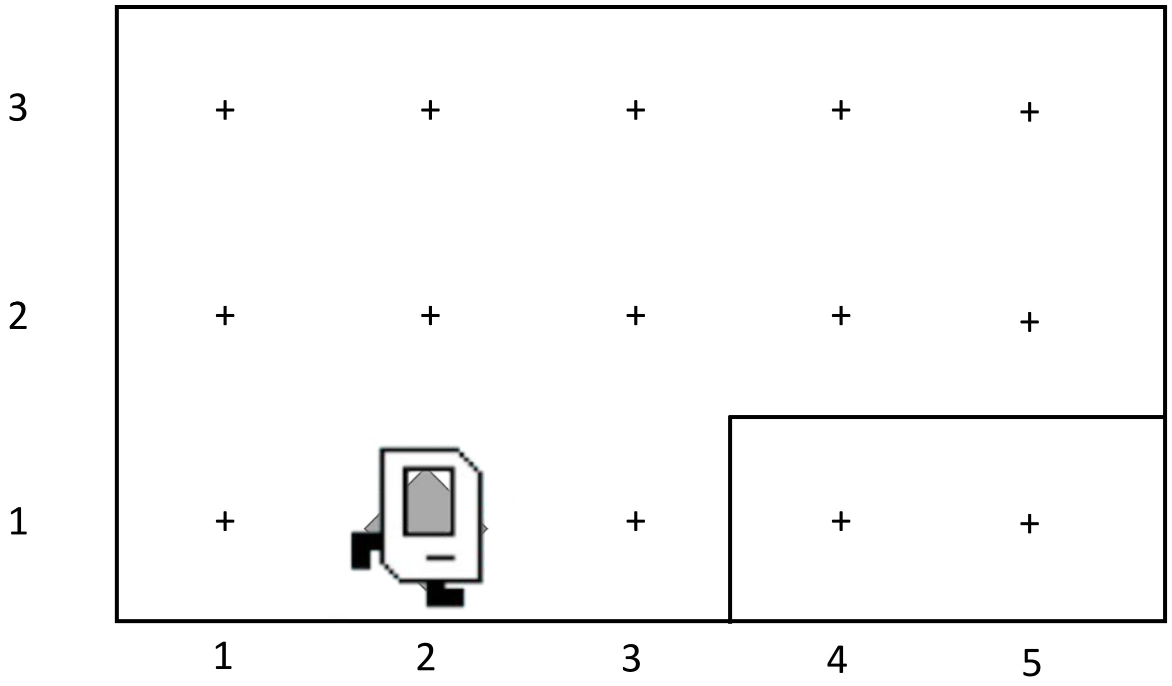
```
move();

turnLeft();

putBeeper();

pickBeeper();
```

```
move();
```

`move();`

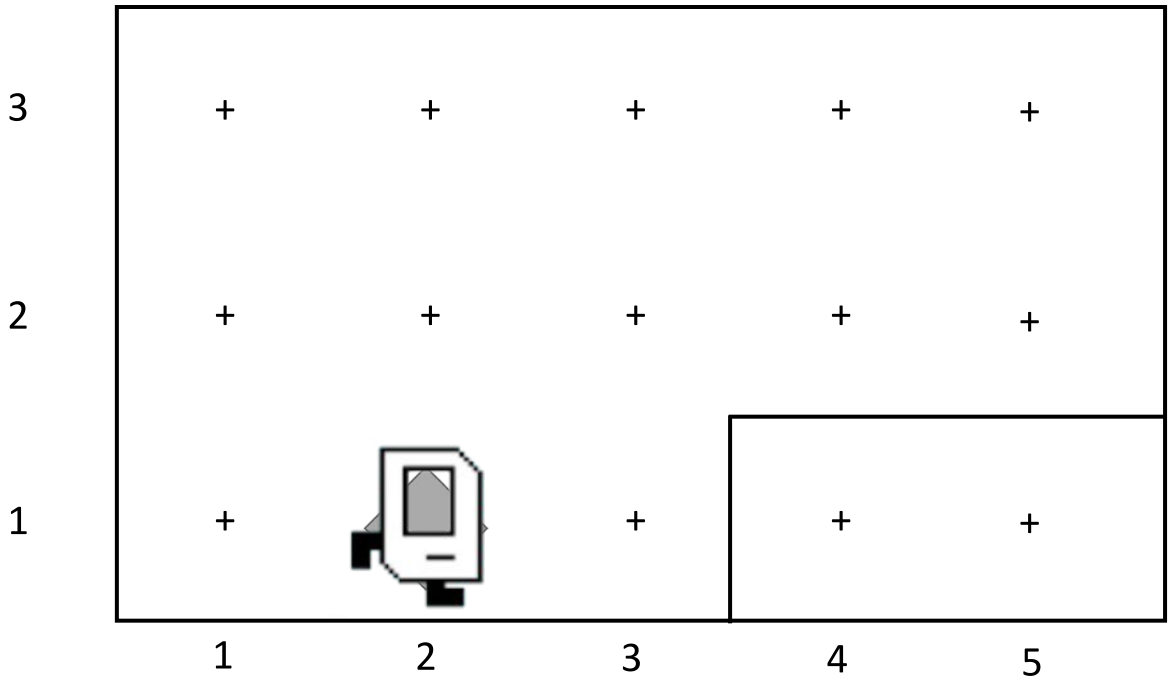


`move();`

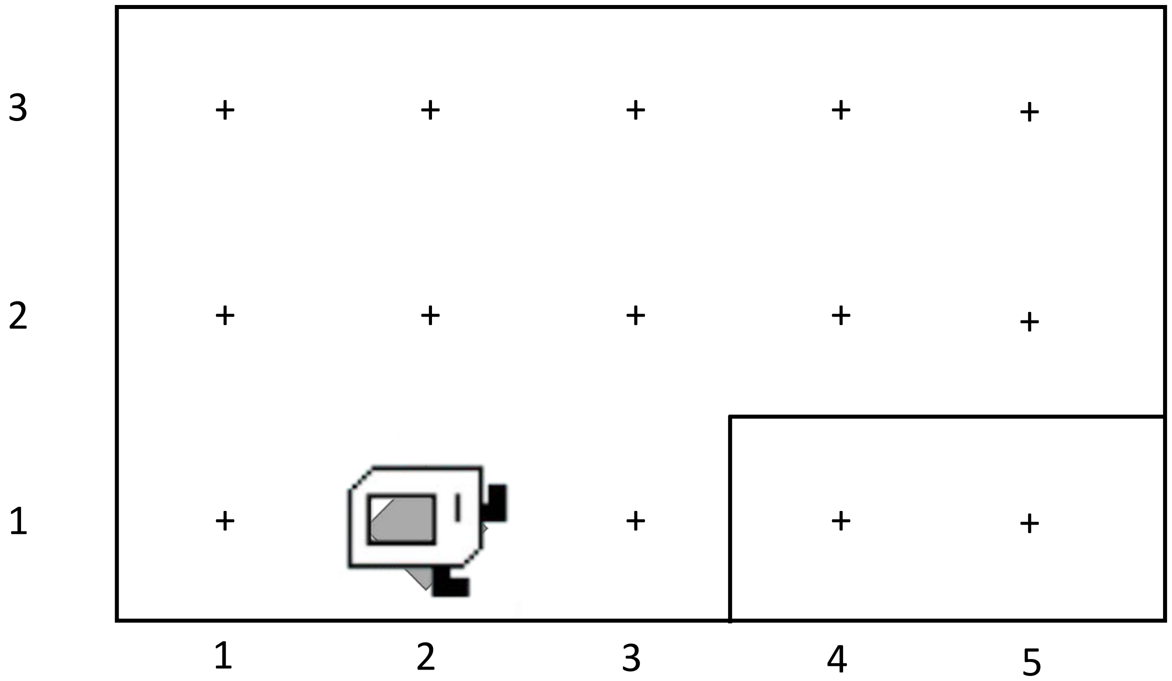


```
turnLeft();
```

`turnLeft();`

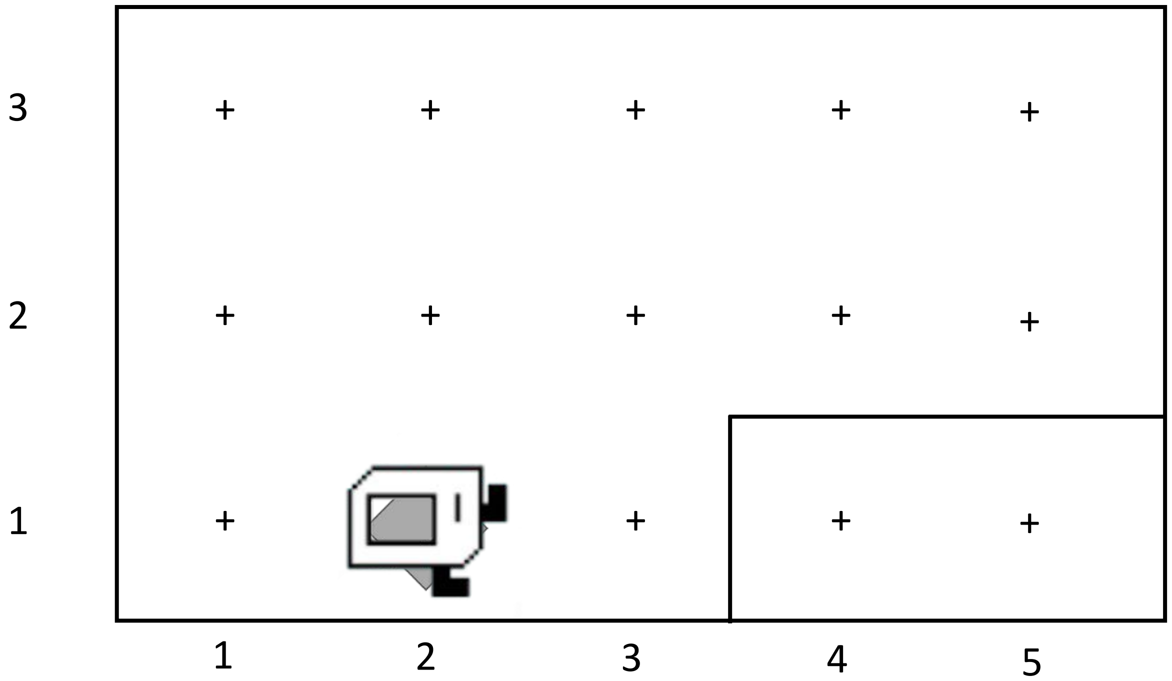


`turnLeft();`

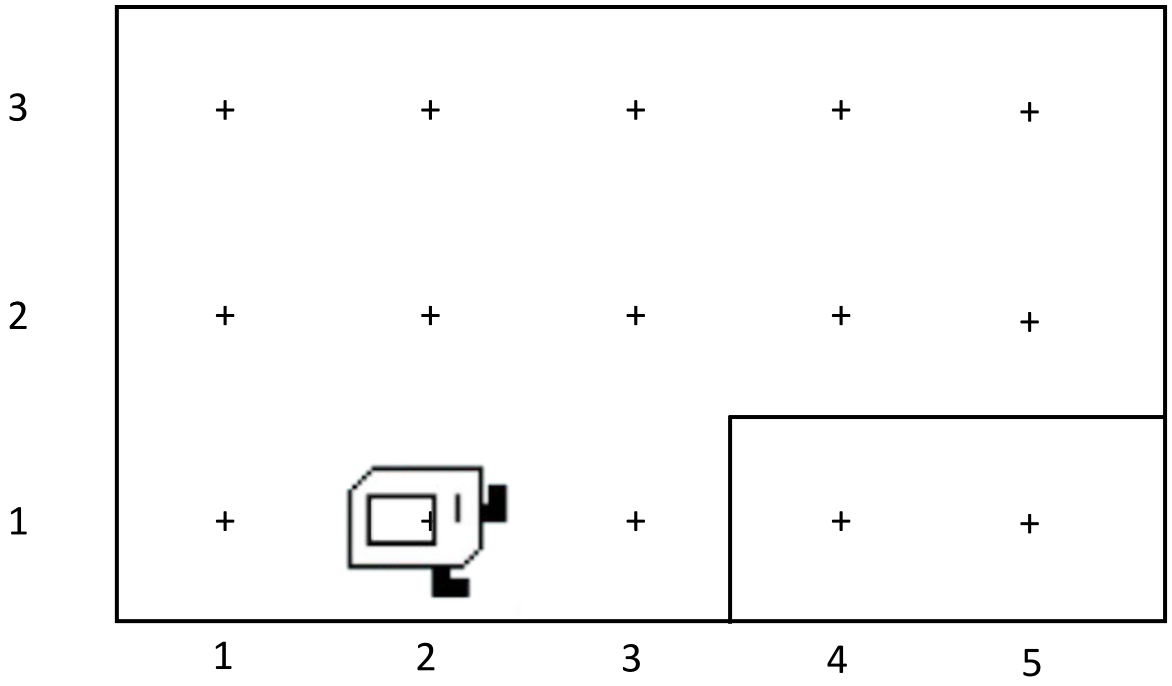


```
pickBeeper();
```

`turnLeft();`

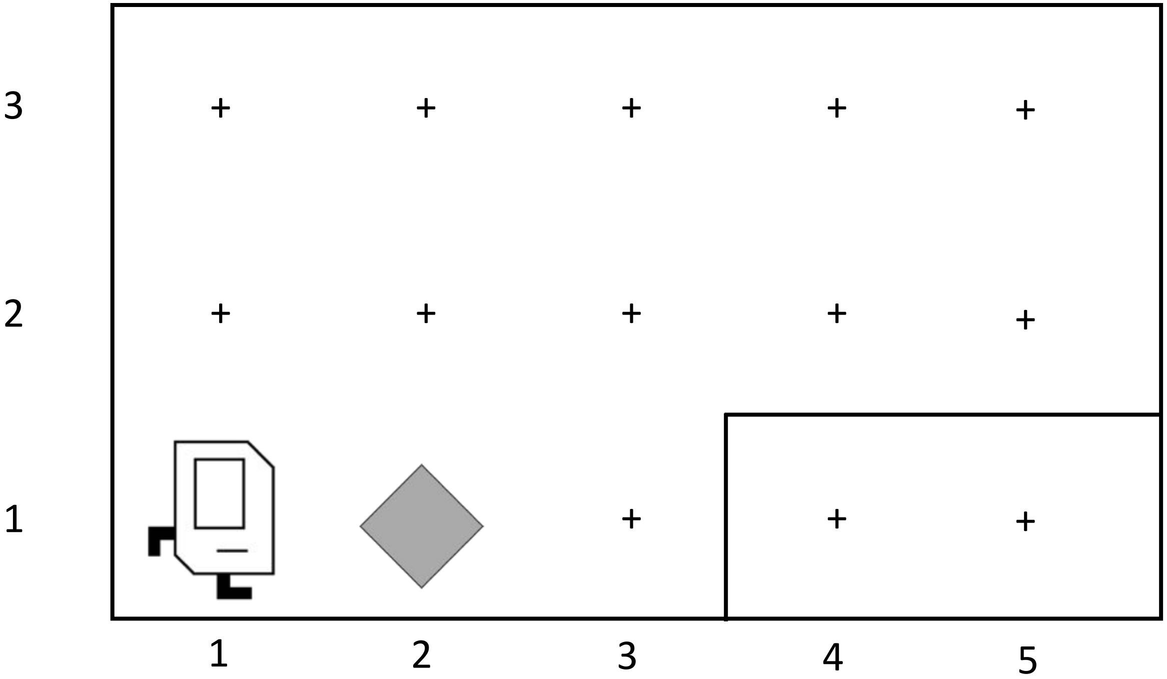


`turnLeft();`

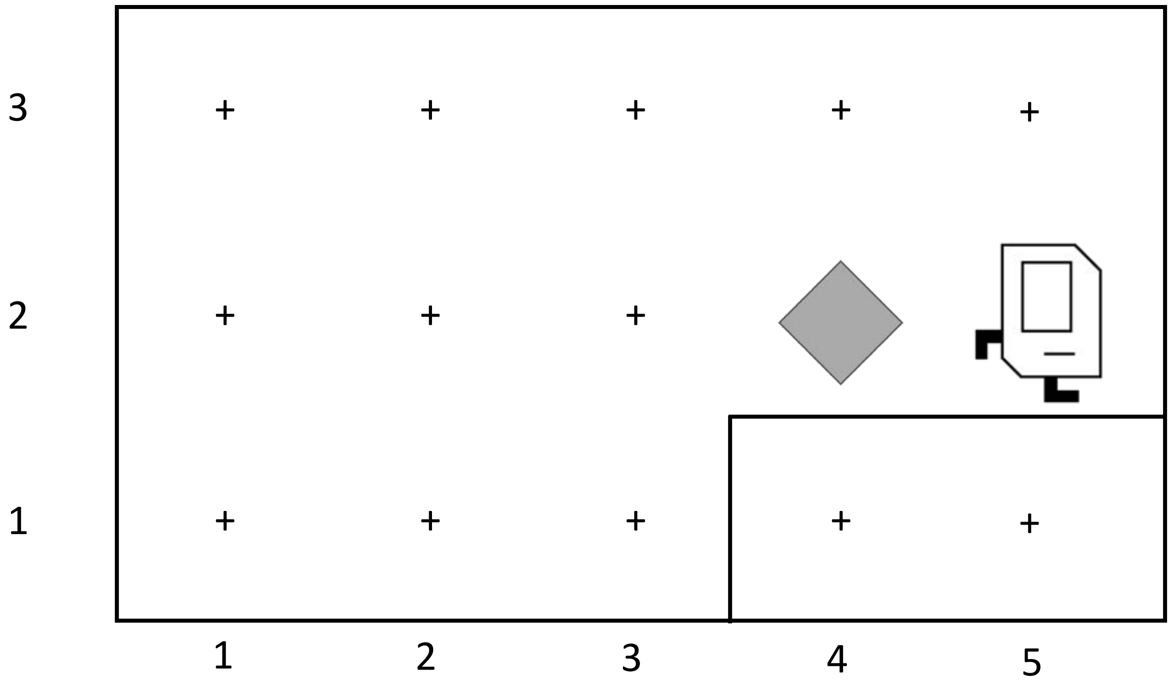


Make Sense?

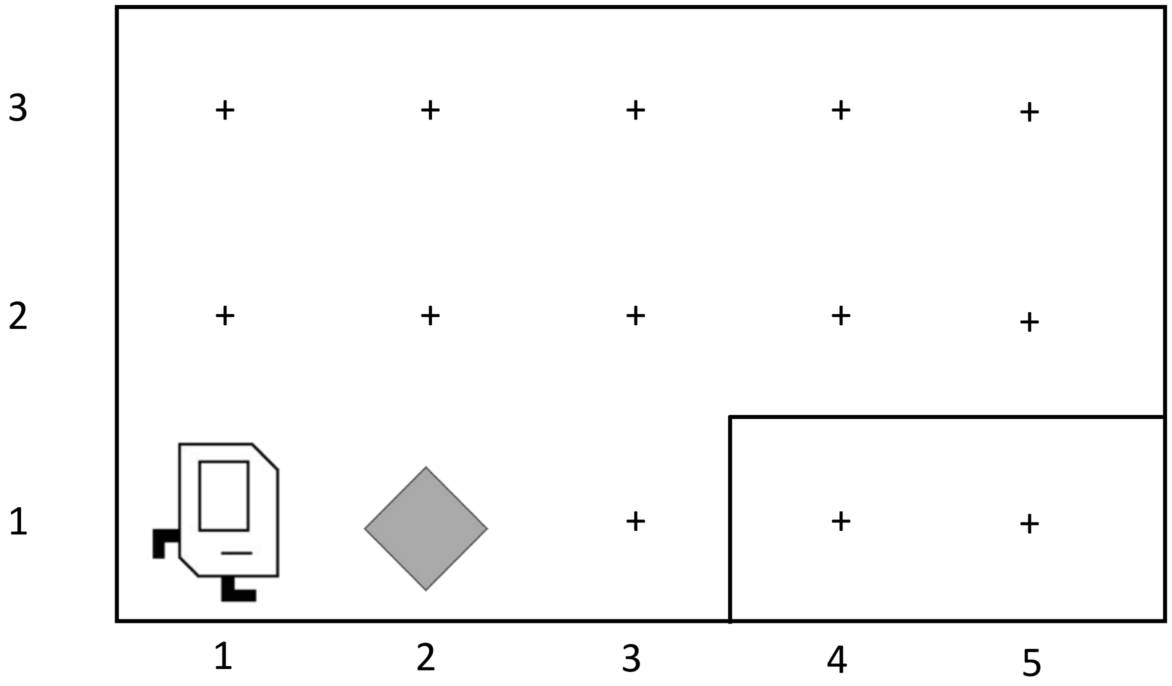
First Challenge



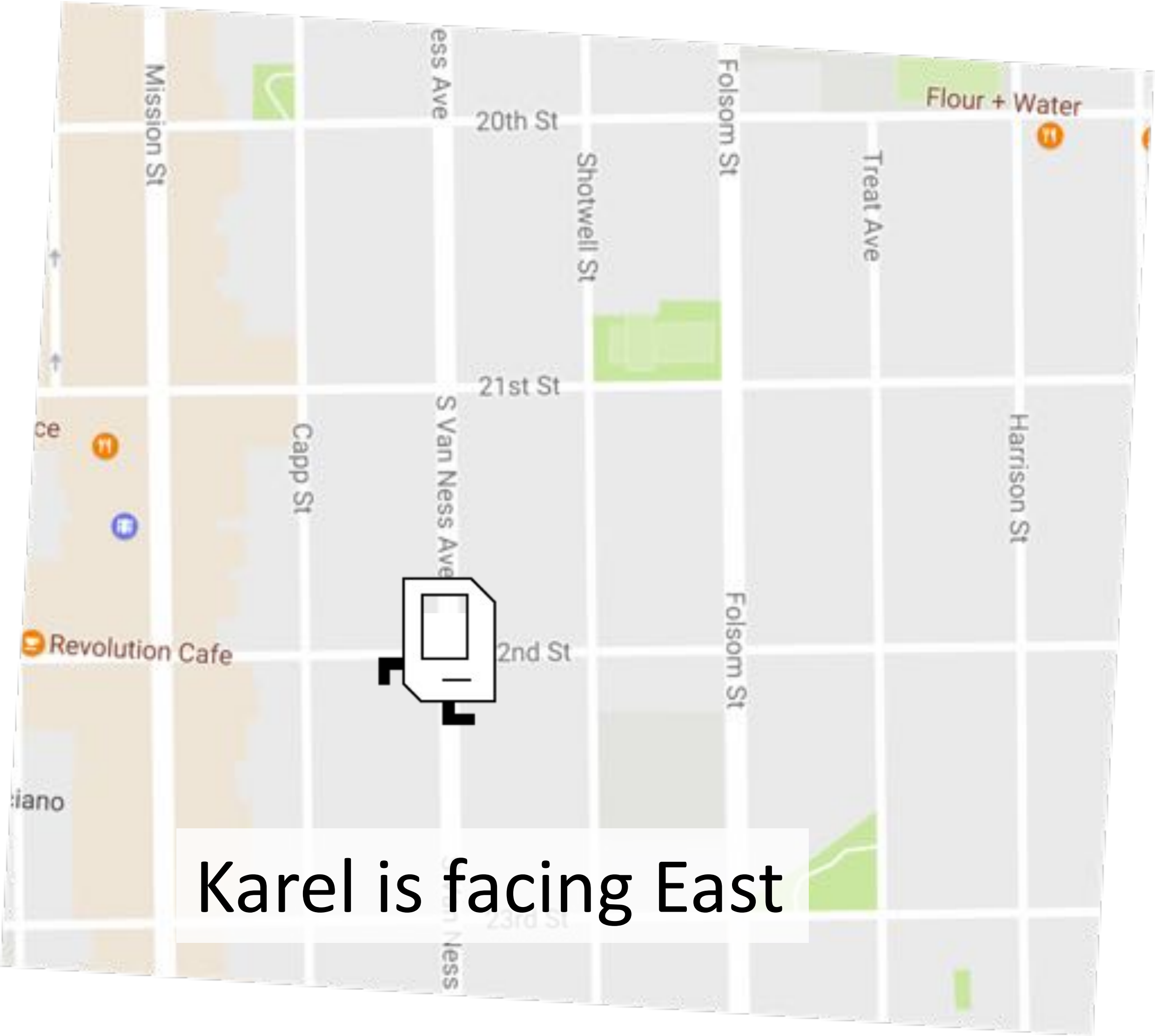
First Challenge



Bird's Eye View

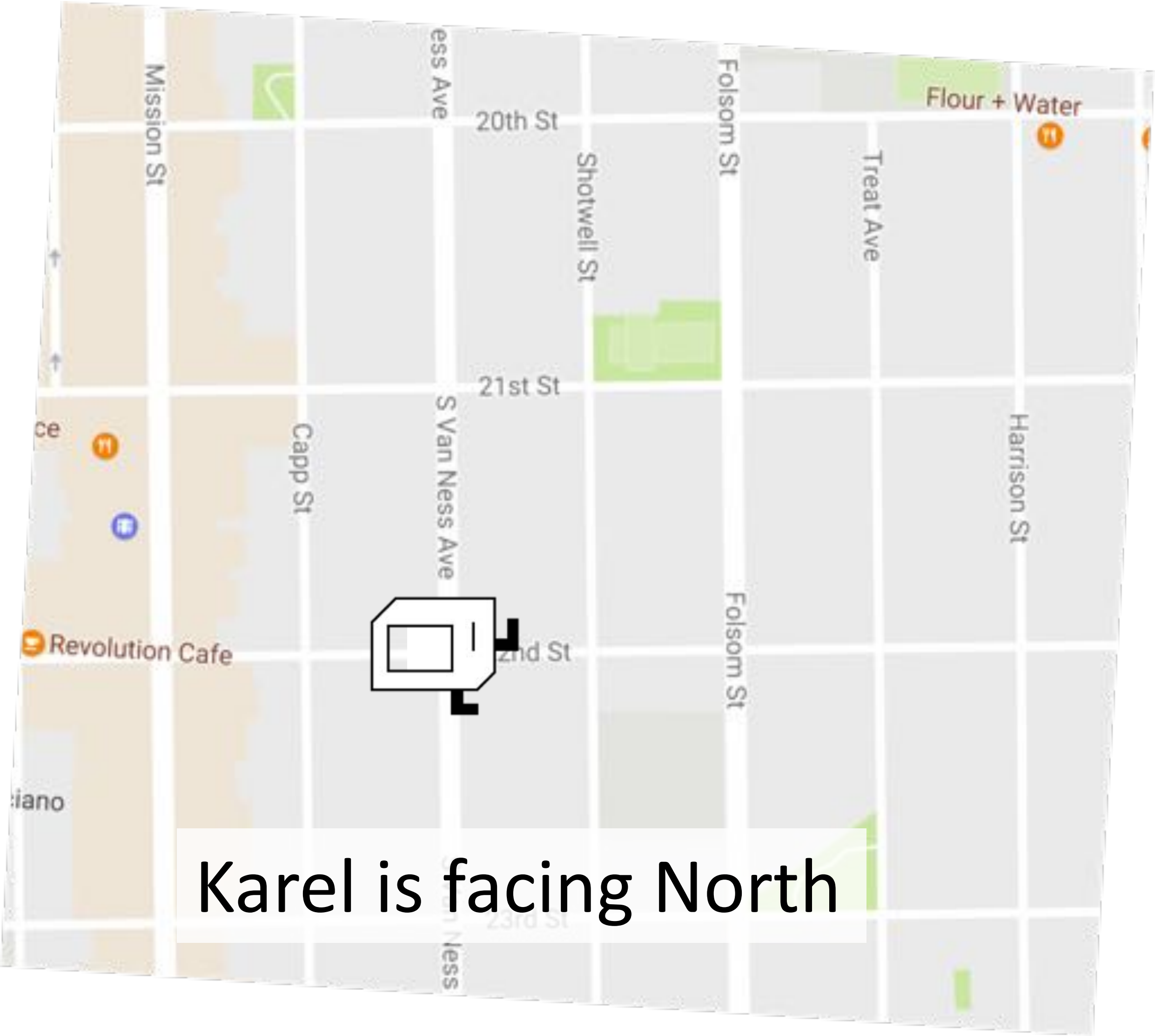


Bird's Eye View



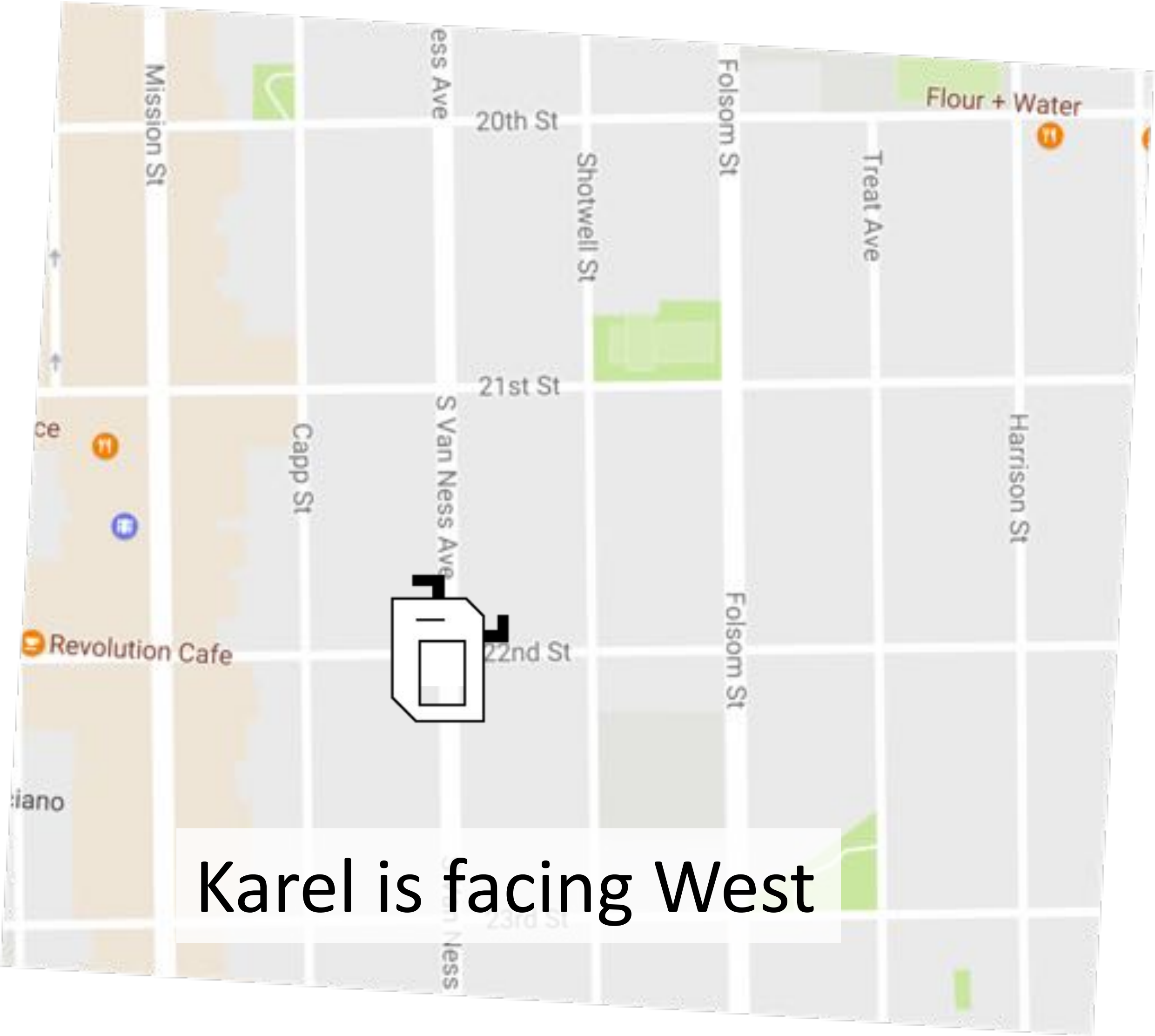
Karel is facing East

Turn Left



Karel is facing North

Turn Left



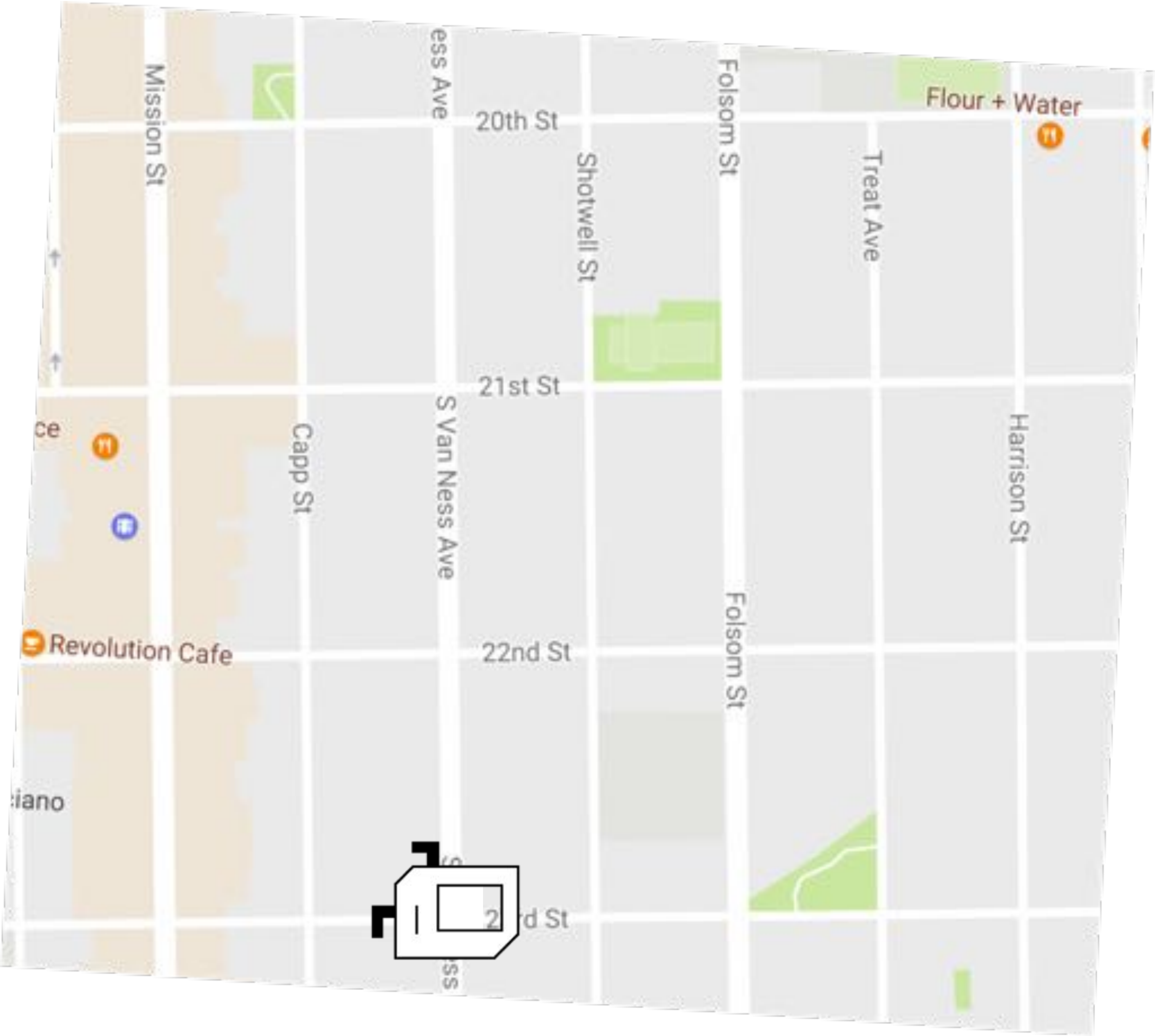
Karel is facing West

Turn Left

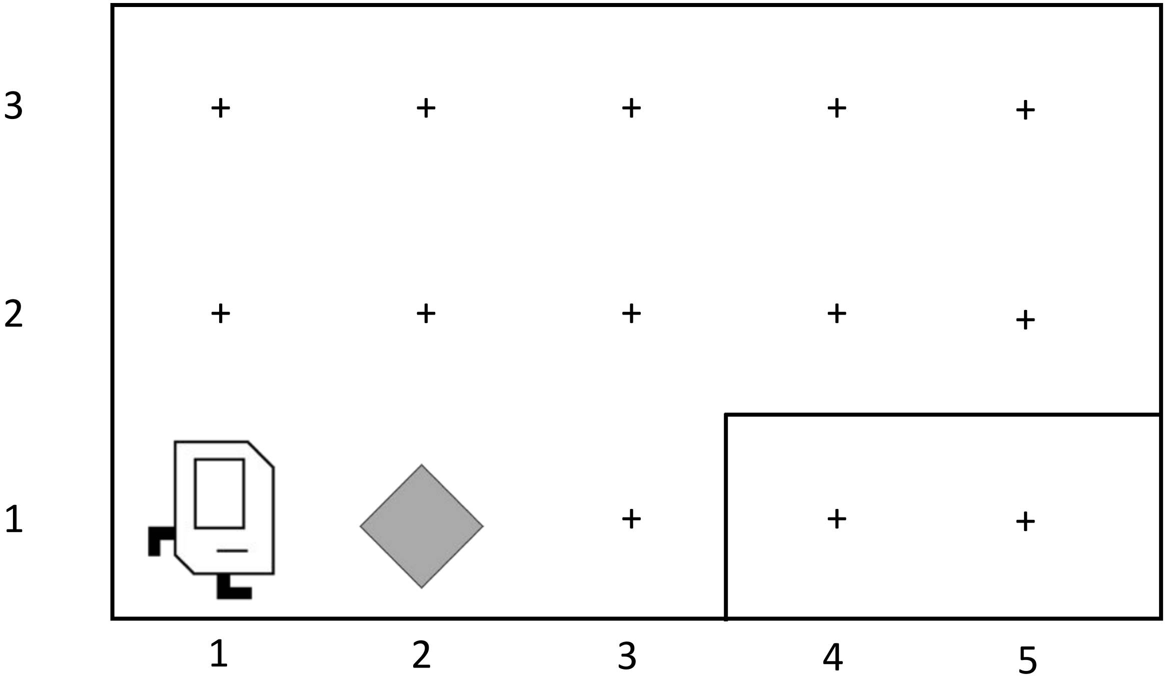
Mission St
ess Ave
20th St
Folsom St
Flour + Water
Shotwell St
Treat Ave
21st St
ce
Capp St
S Van Ness Ave
Harrison St
Revolution Cafe
2 nd St
Folsom St
iano
23rd St
Ness

Karel is facing South

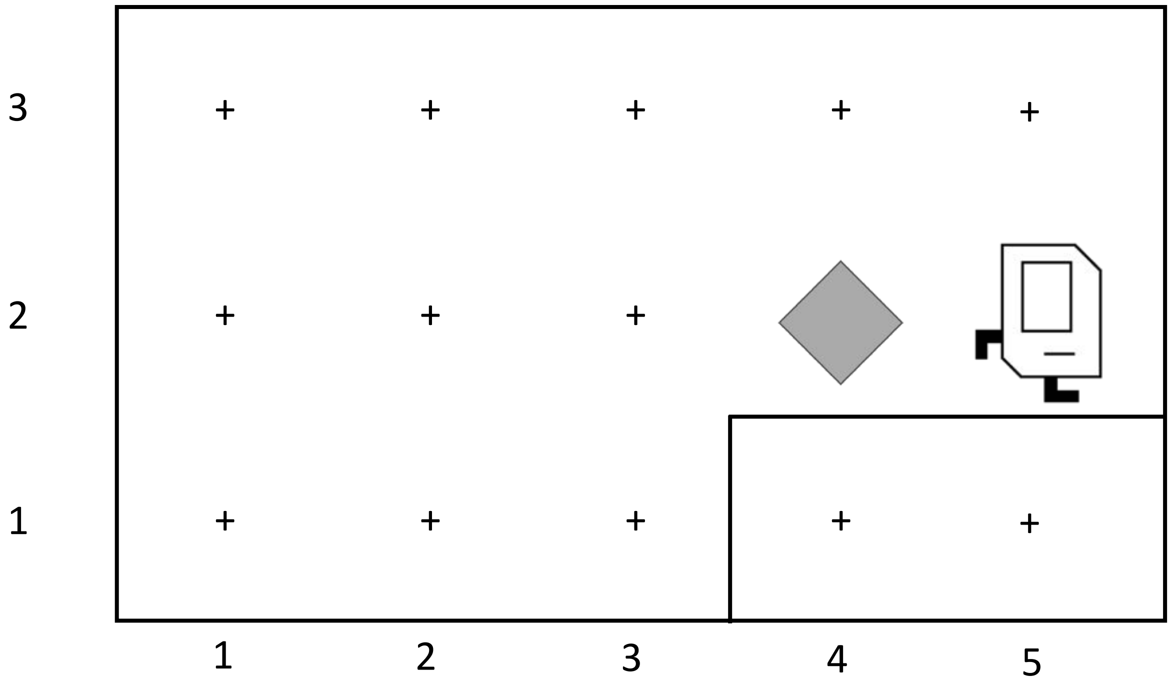
Move


Mission St
ess Ave
20th St
Folsom St
Flour + Water
Shotwell St
Treat Ave
21st St
ce
Capp St
S Van Ness Ave
Harrison St
Revolution Cafe
22nd St
Folsom St
iano
2 d St
ss

First Challenge



First Challenge



Learn By Doing

eclipse

Method Definition

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

This adds a new command to Karels vocabulary

Anatomy of a Program

```
Import Packages
```

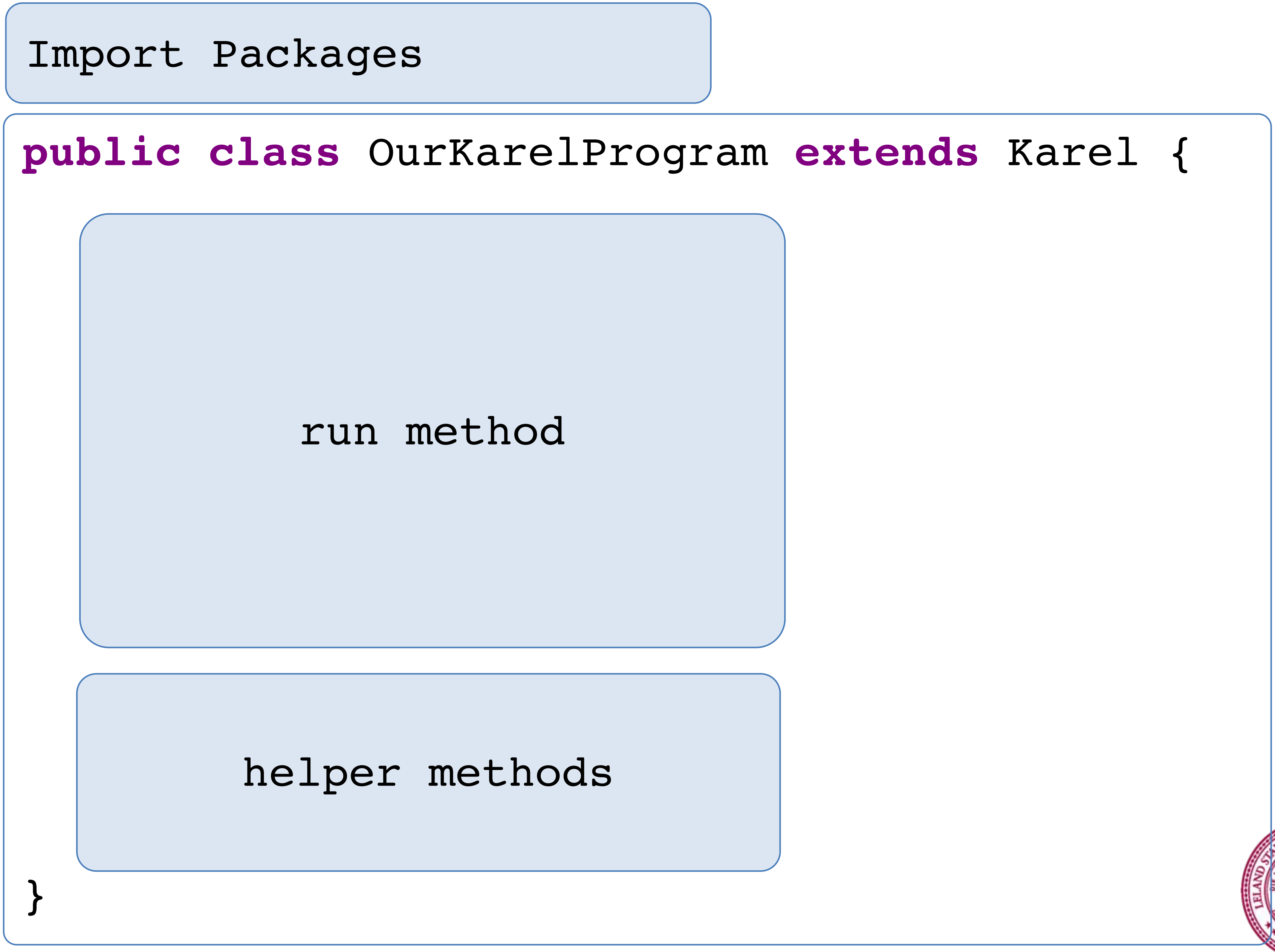
```
Program
```

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

```java
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

```java
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?

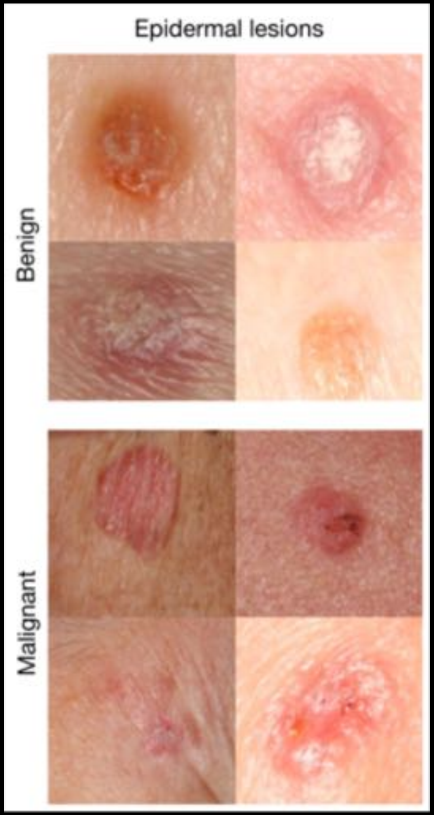
Joy of Building

Interdisciplinary

Closest Thing To Magic

Now is the Time

Now is the Time

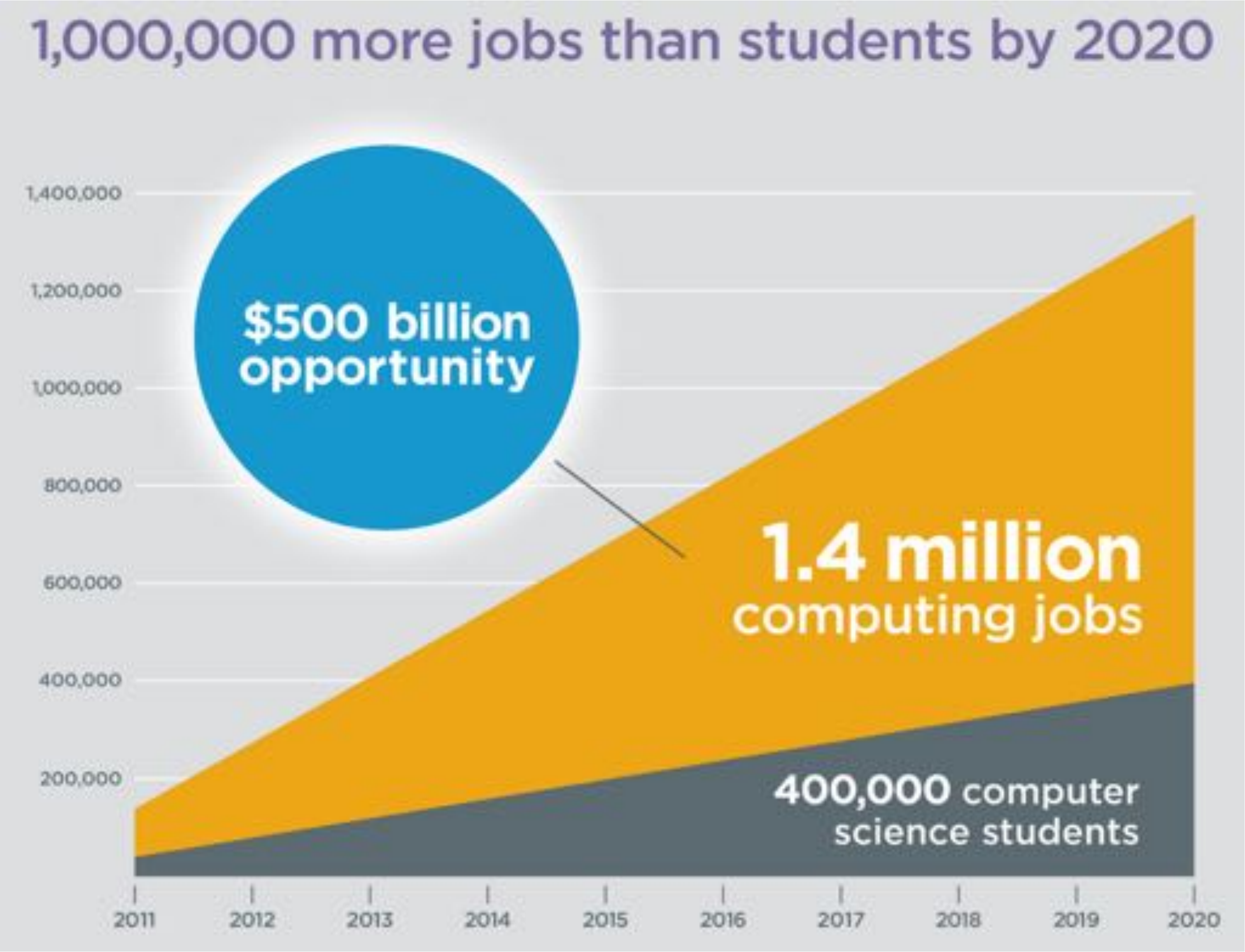


A machine learning algorithm performs **better than** the best dermatologists.

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



Code.org

Everyone is Welcome

The End

The End?

Who are you?