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
11 years ago to this day, I was sitting in your seats
Head TA: Nick Troccoli

Piech, CS106A, Stanford University
Section Leaders

Alisha
Amy
Brian
Bryce
Çağatay
Dilara
Göktuğ
Julia
Kerem
Kevin
Pınar Ö
Şahin
Pınar Y
Nick
Asena

Piech, CS106A, Stanford University
Who are you?
Prerequisite Test

Intro to Computer Science

The Idea of the Course

The point of this two-week course is to teach you the fundamentals of computer programming to the point where you can go and learn on your own. It is taught by a collaboration of instructors from Stanford and Bogazici University. You will learn to program using material for Stanford’s Introduction to Computer Science course.

Piech, CS106A, Stanford University
Course mechanics

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

http://cs106a.stanford.edu
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: 6pm until midnight Sunday through Thursday (starting next Sunday)

Piech, CS106A, Stanford University
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

Assignments 45%

Midterm 30%

Final 15%

Section Participation 10%

*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

*Karel the Robot Learns Java*

- First week of material
CS106A Units

Start Here

Are you an undergrad or SCPD student?

Yes

5 Units

No

Average about 10 hours / week for assignments

Hours per week = Units $\times$ 3

Do you want to take CS106A for fewer units?

Yes

3 Units -or- 4 Units

No

No
Lectures are Online

## Programming Methodology

**CS106A**

### Course Sessions (29):

#### Week 11

<table>
<thead>
<tr>
<th>Day/Date</th>
<th>Watch Online:</th>
<th>Duration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wednesday, December 7, 2016</td>
<td>WATCH NOW</td>
<td>00:48:08</td>
</tr>
<tr>
<td>Monday, December 5, 2016</td>
<td>WATCH NOW</td>
<td>00:48:19</td>
</tr>
</tbody>
</table>

### Course Details

**Course Description**

This introduction to the engineering of computer applications emphasizes modern software development techniques. The "D"s" of development: Design, Development, Debugging, Deployment.

[Course Link](https://videox.stanford.edu/Course/825)
Are you in the right place?
What is CS106A?
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
Pat Hanrahan, one of the founders of Pixar is a professor here
Consumer Applications
Autonomous Surgery
Self Driving Car
If only we could program self driving cars...
Graphical Games
Data Visualization
Mini Facebook

Mehran Sahami

Friends:

Mehran Sahami is coding like a fiend

Displaying Mehran Sahami
Strive for Everyone to Succeed
Art of Computer Science
Something cool
Autonomously Generating Hints by Inferring Problem Solving Policies
Chris Piech, Mehran Sahami, Jonathan Huang, Leo Guibas
Each node is a unique intermediate solution.

Each edge is what a teacher suggested.

Pink dots are students.

Solution

Autonomously Generating Hints by Inferring Problem Solving Policies
Chris Piech, Mehran Sahami, Jonathan Huang, Leo Guibas
Autonomously Generating Hints by Inferring Problem Solving Policies
Chris Piech, Mehran Sahami, Jonathan Huang, Leo Guibas
Process Predicts Future

We want to help you learn the process of programming.

Effect is large and logarithmic.
Meet Karel the Robot

Good morning
Karel Speaks Java
Why Java?

Job postings containing top languages
Indeed.com - November, 17th 2017

1. Java
   - 68K
2. Python
   - 46K
3. JavaScript
   - 40K
4. C++
   - 31K

## Karel’s World

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>![Robot Icon]</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

- **North**
- **West**
- **South**
- **East**
Walls
Knows Four Commands

move();

turnLeft();

putBeeper();

pickBeeper();
move();
`move();`
move();
turnLeft();
```javascript
turnLeft();
```
turnLeft();

---

Piech, CS106A, Stanford University
pickBeeper();
`pickBeeper();`
pickBeeper();
Make Sense?
First Challenge
First Challenge
Bird’s Eye View

Karel is facing East
Bird’s Eye View

Karel is facing East

Piech, CS106A, Stanford University
Turn Left

Karel is facing North
Turn Left

Karel is facing West
Turn Left

Karel is facing South
First Challenge
First Challenge
First Challenge
eclipse
Method Definition

private void name() {
    method statements
}

This adds a new command to Karels vocabulary
Anatomy of a Program

Import Packages

Program
public class OurKarelProgram extends Karel {

}
import packages

public class OurKarelProgram extends Karel {
    run method
    helper methods
}

Anatomy of a Program
public class OurKarelProgram extends Karel {

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

    helper methods
}
Anatomy of a Program

public class OurKarelProgram extends Karel {

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

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

}
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();
        turnLeft();
    }

}
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();
        }

}
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();
turnLeft();
    }

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

This is also called a code block
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();
    }
}
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();
turnLeft();
    }

}
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();
        turnLeft();
    }

}
Why Study CS?
Joy of Building
Interdisciplinary
Closest Thing To Magic
Now is the Time
Oh and Its Useful

1,000,000 more jobs than students by 2020

$500 billion opportunity

1.4 million computing jobs

400,000 computer science students

Code.org
Everyone is Welcome
The End
The End?