CS 106A, Lecture 1
Welcome to the course!

required reading:
General Information handout

suggested reading:
Karel, Ch. 1-2
Plan For Today

• Introduction
• Course Policies
• Meet Karel the Robot
Plan For Today

• Introduction
• Course Policies
• Meet Karel the Robot
Nice to meet you!

Instructor: Colin Kincaid

Head TA: Annie Hu
Section Leaders

• Helpful undergraduate assistants who will:
  – run your discussion section each week
  – grade your homework assignments and exams
  – help you when you have questions
  – ... and much more
Nice to meet you!
Nice to meet you!

Michelle McGhee

Ruiqi Chen

Shanon Reckinger

Yoni Lerner

Allison Tielking

Connor Meany

Deanna Garcia

Greg Ramel
What is Computer Science?

• The art of using computing to solve complex problems.
  – Specify *instructions* that computers execute, usually in a *programming language*
• Applicable to art, medicine, linguistics, and more
• Touches many aspects of our daily lives
What is CS 106A?

• Programming *Methodology*
  – “A *system* of methods used in a particular area of study or activity”
  – Focusing on computational problem solving, not Java syntax
  – No prior programming experience required!

• Topics include:
  – Karel the Robot
  – Text-based programs
  – Graphics and animation
  – Games
  – And more...
Is 106A the right fit for you?

- **General topics we will cover:**
  - variables
  - control flow (loops and conditionals)
  - functions or methods (including parameters and return statements)
  - classes and objects
  - arrays
  - console and file I/O

- If those things are familiar to you, consider CS106B!
- If you know lots and still take this class, great! Just be mindful of everyone else
Time-out: Think, Pair, Share

- What is one of your core values?
  – Optionally: How might you use computing in pursuit of that value?
  – If you feel comfortable: Share with a neighbor!
Plan For Today

• Introduction
• Course Policies
• Meet Karel the Robot
cs106a.stanford.edu
Stanford grad student?

- No
  - 5!
- Yes
  - 3-5!
Textbooks

• *Karel the Robot Learns Java*, coursereader (35 pages)
  – used this week and next week as we introduce coding
  – refer to this coursereader for Java features you are
    allowed to use on the first assignment
  – free PDF available online

• *The Art & Science of Java*, by Eric Roberts
  – written here at Stanford; tailored to this course; a valuable reference
  – available on reserve at library
Grading

*****  45%  Programming assignments
*
*  10%  Section participation and lecture feedback
**
**  15%  Midterm Exam
****
****  30%  Final Exam
<table>
<thead>
<tr>
<th>Grading</th>
<th>Percentage</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming</td>
<td>45%</td>
<td>Programming assignments</td>
</tr>
<tr>
<td>assignments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>10%</td>
<td>Section participation and lecture feedback</td>
</tr>
<tr>
<td>**</td>
<td>15%</td>
<td>Midterm Exam</td>
</tr>
<tr>
<td>****</td>
<td>30%</td>
<td>Final Exam</td>
</tr>
</tbody>
</table>
• 6 programming assignments (some individual, some in pairs), completed using Eclipse
  – free software, available on course website
  – homework: set up Eclipse!
  – come to LaIR this Tuesday 7-9PM for troubleshooting

• Graded on functionality (behavior) and style (elegance)
  – you’ll get lots of feedback on each from your section leader
  – grading scale is divided into "buckets"
# The Bucket System

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
</tbody>
</table>

![Symbol Image](image-url)
## The Bucket System

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ν+</td>
<td>Well done; satisfies all assignment requirements</td>
</tr>
<tr>
<td>ν</td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
</tbody>
</table>
# The Bucket System

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v+</td>
<td>Well done; satisfies all assignment requirements</td>
</tr>
<tr>
<td>v</td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
<tr>
<td>v-</td>
<td>Problems serious enough to fall short of assignment requirements</td>
</tr>
</tbody>
</table>
### The Bucket System

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Exceeds expectations; often reflects additional work</td>
</tr>
<tr>
<td>v+</td>
<td>Well done; satisfies all assignment requirements</td>
</tr>
<tr>
<td>v</td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
<tr>
<td>v-</td>
<td>Problems serious enough to fall short of assignment requirements</td>
</tr>
</tbody>
</table>
## The Bucket System

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Exceeds expectations; often reflects additional work</td>
</tr>
<tr>
<td>ν+</td>
<td>Well done; satisfies all assignment requirements</td>
</tr>
<tr>
<td>ν</td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
<tr>
<td>ν-</td>
<td>Problems serious enough to fall short of assignment requirements</td>
</tr>
<tr>
<td>-</td>
<td>Extremely serious problems, incomplete effort or understanding</td>
</tr>
</tbody>
</table>


## The Bucket System

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>++</td>
<td>Absolutely fantastic submission (<em>very rare</em>)</td>
</tr>
<tr>
<td>+</td>
<td>Exceeds expectations; often reflects additional work</td>
</tr>
<tr>
<td>(\nabla^+)</td>
<td>Well done; satisfies all assignment requirements</td>
</tr>
<tr>
<td>(\nabla)</td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
<tr>
<td>(\nabla^-)</td>
<td>Problems serious enough to fall short of assignment requirements</td>
</tr>
<tr>
<td>-</td>
<td>Extremely serious problems, incomplete effort or understanding</td>
</tr>
</tbody>
</table>
### The Bucket System

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>++</td>
<td>Absolutely fantastic submission <em>very rare</em></td>
</tr>
<tr>
<td>+</td>
<td>Exceeds expectations; often reflects additional work</td>
</tr>
<tr>
<td>v+</td>
<td>Well done; satisfies all assignment requirements</td>
</tr>
<tr>
<td>v</td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
<tr>
<td>v-</td>
<td>Problems serious enough to fall short of assignment requirements</td>
</tr>
<tr>
<td>-</td>
<td>Extremely serious problems, incomplete effort or understanding</td>
</tr>
<tr>
<td>--</td>
<td>Little effort</td>
</tr>
</tbody>
</table>
## The Bucket System

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>++</td>
<td>Absolutely fantastic submission (<em>very rare</em>)</td>
</tr>
<tr>
<td>+</td>
<td>Exceeds expectations; often reflects additional work</td>
</tr>
<tr>
<td>v+</td>
<td>Well done; satisfies all assignment requirements</td>
</tr>
<tr>
<td>v</td>
<td>Satisfactory; meets requirements, maybe a few issues</td>
</tr>
<tr>
<td>v-</td>
<td>Problems serious enough to fall short of assignment requirements</td>
</tr>
<tr>
<td>-</td>
<td>Extremely serious problems, incomplete effort or understanding</td>
</tr>
<tr>
<td>--</td>
<td>Little effort</td>
</tr>
<tr>
<td>0</td>
<td>No submission</td>
</tr>
</tbody>
</table>
Getting Help

• Visit the SLs in the LaIR (1st floor of Tressider Union)
  – open Sun-Wed, 7PM – 11PM, starting this Wednesday
  – staffed with multiple section leaders to answer questions
  – use the CLaIR (Conceptual LaIR) queue for course material questions

• other help resources:
  – instructor office hours
  – head TA office hours
  – email SL, TA, instructor

• Eclipse troubleshooting session Tuesday 6/26 7-9PM @ LaIR
2 Minds are Better Than 1

• Some assignments may optionally be done in **pairs**
• Both partners receive the same grade
• A chance to brainstorm ideas and work with another programmer
• **MUST** be in the same section!
  – put the same section preferences to make this happen!
• More info in handout #1 and on the course website
Late Days

- **Start out with 3 “free late days”:** each late day allows you to submit an assignment 24 hours late without penalty.
- Hard deadline 48 hours after original due date
- 1-bucket deduction per day late after late days are exhausted
- Pair late days are assessed individually
- “Pre-granted extensions” – additional extensions granted only in *very special* circumstances. **Head TA** must approve extensions.
Grading

***** 45%  Programming assignments
* 10%  Section participation and lecture feedback
** 15%  Midterm Exam
**** 30%  Final Exam
• Weekly 50-minute sections led by your section leader
• Go over lecture material, do practice problems, answer questions
• Graded on section attendance + participation
• **Homework**: sign up for section on the course website!
Lecture Feedback

• Give me feedback so that I can do better!
• Anonymous, and you get credit no matter what you write
  – unless it is inappropriate
• Give feedback by the Monday after your assigned lecture
• More instructions on the website under “Lectures”
• Homework: figure out which lectures you are assigned for giving feedback
<table>
<thead>
<tr>
<th></th>
<th>Percentage</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>******</td>
<td>45%</td>
<td>Programming assignments</td>
</tr>
<tr>
<td>*</td>
<td>10%</td>
<td>Section participation and lecture feedback</td>
</tr>
<tr>
<td>**</td>
<td>15%</td>
<td>Midterm Exam</td>
</tr>
<tr>
<td>****</td>
<td>30%</td>
<td>Final Exam</td>
</tr>
</tbody>
</table>
Exams

• **Midterm exam** – Monday, July 23rd, 7-9PM
  – Fill out the head TA’s exam form by *July 9* if you have an academic or University conflict or OAE accommodations

• **Final exam** – Friday, August 17th, 12:15-3:15PM
  – No alternate final! You **MUST** be able to take the final exam at the scheduled time (except for OAE accommodations)

• On both exams, you will be allowed two double-sided “cheat sheets”: handwritten or typed notes that you think will be helpful.
  – You will also be provided Java syntax reference sheets
Grading

****** 45%  Programming assignments
* 10%  Section participation and lecture feedback
** 15%  Midterm Exam
**** 30%  Final Exam
The **Honor Code** is an undertaking of the students, individually and collectively:
- that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
- that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

• Please help us ensure academic integrity:
  – indicate any assistance received on HW (books, web sites, friends).
  – do not look at other people's solution code (*outside of your pair*).
  – do not give your solution code to others, or post it on the web.
  – report any inappropriate activity you see performed by others.

• Assignments are checked regularly for similarity with help of software tools.

• If you realize that you have made a mistake, you may retract your submission to any assignment at any time, no questions asked.

• If you need help, please contact us and we will help you.
  – we do not want you to feel any pressure to violate the Honor Code in order to succeed in this course.
Plan For Today

• Introduction
• Course Policies
• Meet Karel the Robot
Meet Karel the Robot!
Meet Karel the Robot!

Hello, world!
Karel's World
Karel's World

1 2 3 4 5

3 2 1

1
Hello, world!
Streets (rows)

Each row is called a street.
Avenues (columns)

Each column is called an avenue.
Corners (locations)

The intersection of a street and an avenue is a corner.
Walls

Karel cannot move through walls.
Beepers mark locations in Karel's world. Karel can pick them up and put them down.
Wrap-up

- Introduction ✓
- Course Policies ✓
- Meet Karel the Robot ✓

Next time: more programming with Karel!