Welcome!

We'll get started shortly ...

Welcome to Section!

Week One

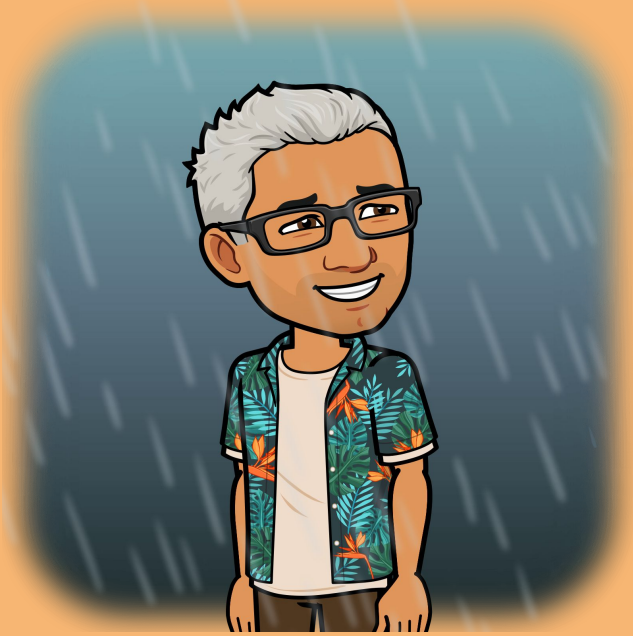
Surajit Bose

Agenda

- Introductions
- Logistics
  - Section purpose and setup
  - Getting help
- Review of lecture concepts
  - Karel's world
  - Karel conditions
  - Karel commands
- Worked Example: Move Beeper

Introductions

A Little About Me



- ★ My name is Surajit (he/him), and I'll be your Section Leader for CS 49
- ★ I am retired from a tech career
- ★ In my free time I like to read novels and poetry
- ★ I enjoy Indian classical music

What About You?

Please let us know:

1. Your name, and if you would like to share them, your pronouns
2. Where you're tuning in from: locally from the Bay Area, or farther away?
3. Whether you have any prior programming/coding experience
4. What you'd like to get out of this class

Another question: Is there something I can do to help you feel comfortable during section? Please feel free to private message me in the chat if so!

Breaking the Ice

- Share your names one more time!!!
- Icebreaker Question:
  - What is your favorite home-cooked dish?
  - Who makes it?
- If no one wants to share first, the person who is geographically closest to Foothill shares first!
- After your turn, you get to "popcorn" to someone, i.e., choose who goes next

Logistics

What is Section?

- Section is your chief support system for CS 49
- The goal of section is to ensure your success in the class
- Up to ten students per section meet weekly with a volunteer tutor
  - This is likely the smallest class meeting you'll have at Foothill
- Major part of class participation component of grade
  - 30% of the grade for the class
  - Active participation: ask questions, provide answers
- Occasional extra credit opportunities
  - Up to 3% of total grade for the class

What happens during section?

- Zoom poll at the start of the meeting (so please show up on time!)
- Any needed logistics (~5 minutes)
- Overview of week's material (~10 minutes)
  - Ask for clarification of any difficulties with the lectures or readings
- Coding practice (~25 minutes)
  - One or more section problems separate from weekly assignments
  - Generally two or three students work on these in breakout rooms
  - One student shares screen and all solve the problem together
- Discussion of the solution with the entire group (~10 minutes)
- Optional lab/office hours after section time
  - For working on the weekly assignments or getting help

Section Norms

- Section is flipped—it is your time to get help with, review, and practice the week's material
- The expectation is that prior to section time, students will have:
  - Viewed the lecture videos
  - Done the reading
  - Begun working on the assignments for the week
- If you're behind on the week's work, come anyway!
- Typically, video camera and microphone must be on during section
  - If you cannot have your camera on, let Lane and me know
  - In noisy environments, please mute your mic

Section infrastructure (1 / 2)

- Section area of Code in Place website
  - Section zoom link
  - Section problem descriptions
  - Coding area (IDE)
  - Section problem solutions (posted on Saturdays)
- Section github repo
  - Lecture slides from Mehran and Chris
  - Section slides (Note: links do not work in preview; must download slides)
  - Coding examples and starter code
  - Section problem solutions (different from the ones on the CiP site)

Section infrastructure (2 / 2)

- Pronto for announcements and messaging
  - A highly recommended brief video on using Pronto within Canvas
  - Download and quickstart guide for using the mobile app
- Section announcements via Pronto
  - Usually two per week: one before, one after section meeting
- Messaging via Pronto
  - To reach me, Canvas inbox is preferred
  - Can DM others in this section
  - Can also DM Lane

Any Questions?

How to get hold of me / get help+

- The class forum. Feel free not only to ask, but also to answer questions!
- Surajit's office hours:
  - Fridays 12 noon–1p, directly after section
  - By appointment on Zoom
- Lane's office hours
- Canvas inbox or Pronto inbox for Lane
- Canvas inbox (preferred) or Pronto for Surajit
- Sina's support section, Fridays 2p–3p on Zoom
- Email bosesurajit@fhda.edu, 24 hr turnaround
- Online or in-person tutoring via the STEM center (Room 4213)
- The section GitHub repo has lecture and section slides and solutions

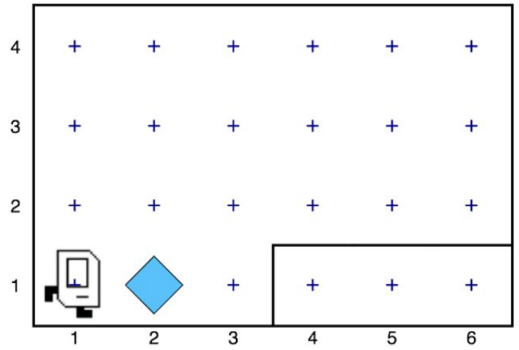
Lecture Review: Karel

Karel the Robot

- Karel is a robot who occupies a certain **world**
  - The world has certain elements like corners, beepers, and walls that Karel interacts with or navigates
- Karel and its world have certain **conditions**
  - Conditions are statements that can be either `True` or `False`
  - In our world, for example, one condition can be "it is raining"
  - Determining whether a condition is `True` or `False` is called **evaluating** the condition
- Karel can perform certain **commands** to navigate its world
- Let's look at Karel's world, conditions, and commands in the next few slides

Karel's World

Karel's world is defined by rows running horizontally (east-west) and columns running vertically (north-south). The intersection of a row and a column is called a corner. Karel can only be positioned on corners and must be facing one of the four standard compass directions (north, south, east, west). A sample Karel world is shown below. Here Karel is located at the corner of 1st row and 1st column, facing east.



Several other components of Karel's world can be seen in this example. The object in front of Karel is a beeper. As described in Rich Pattis's book, beepers are "plastic cones which emit a quiet beeping noise." Karel can only detect a beeper if it is on the same corner. The solid lines in the diagram are walls. Walls serve as barriers within Karel's world. Karel cannot walk through walls and must instead go around them. Karel's world is always bounded by walls along the edges, but the world may have different dimensions depending on the specific problem Karel needs to solve.

Karel Conditions

Test	Opposite	What it checks
`front_is_clear()`	`front_is_blocked()`	Is there a wall in front of Karel?
`beepers_present()`	`no_beepers_present()`	Are there beepers on this corner?
`left_is_clear()`	`left_is_blocked()`	Is there a wall to Karel's left?
`right_is_clear()`	`right_is_blocked()`	Is there a wall to Karel's right?
`beepers_in_bag()`	`no_beepers_in_bag()`	Does Karel have any beepers in its bag?
`facing_north()`	`not_facing_north()`	Is Karel facing north?
`facing_south()`	`not_facing_south()`	Is Karel facing south?
`facing_east()`	`not_facing_east()`	Is Karel facing east?
`facing_west()`	`not_facing_west()`	Is Karel facing west?

Karel Commands

Command	Description
`move()`	Asks Karel to move forward one block. Karel cannot respond to a `move()` command if there is a wall blocking its way.
`turn_left()`	Asks Karel to rotate 90 degrees to the left (counterclockwise).
`pick_beeper()`	Asks Karel to pick up one beeper from a corner and stores the beeper in its beeper bag, which can hold an infinite number of beepers. Karel cannot respond to a `pick_beeper()` command unless there is a beeper on the current corner.
`put_beeper()`	Asks Karel to take a beeper from its beeper bag and put it down on the current corner. Karel cannot respond to a `put_beeper()` command unless there are beepers in its beeper bag.

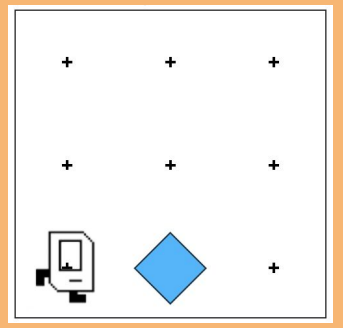
Any Questions?

Worked Example: Move Beeper

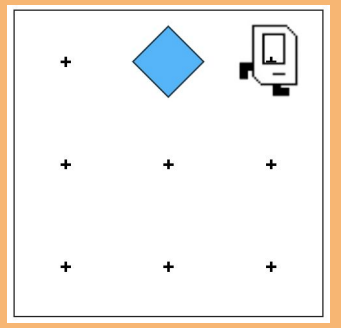
https://codeinplace.stanford.edu/cs49-w24/ide/a/movebeeper

Move Beeper

- Preconditions
  - Karel is in the bottom left corner, facing east
  - There is a beeper immediately in front of Karel



- Postconditions
  - Karel is in the top right corner, facing east
  - Beeper has been moved and is immediately behind Karel



The Anatomy of a Program

- File name, comment, your name, date
- Import statement
- **`main()`** module with further comments. Indentation is important!
- Guard clause and invocation of **`main()`**

```
"""
main.py

Karel moves a beeper from the bottom row to the top row

Programmer: Surajit A Bose, Date: January 6, 2025
"""

from karel.stanfordkarel import *

def main():
    """
    Move Karel and beeper to top row

    Preconditions:
    - Karel is in the bottom left corner, facing east
    - There is a beeper immediately in front of Karel

    Postconditions:
    - Karel is in the top right corner, facing east
    - Beeper has been moved and is immediately behind Karel
    """
    pass  # Delete this line and write your code here! :)


# There is no need to edit code beyond this point

if __name__ == '__main__':
    main()
```

Let's get to work!