

The background is a solid pink color. It is decorated with various hand-drawn geometric shapes in white and black. These include a dashed line in the top left, a white triangle in the top center, a black zigzag line in the top right, a white circle in the top right, two parallel black lines in the top right, a white triangle in the top right, a large black circle in the bottom right, a white circle in the bottom right, a black plus sign in the bottom left, a white circle in the bottom left, a white triangle in the bottom center, and a dashed line in the bottom center.

Welcome!

We'll get started shortly ...



CS 49 Section

Week 3

Surajit A Bose





Agenda

- Logistics and check-ins
- Review of lecture concepts
 - Decomposition
 - Iterative testing
 - Code style
- Section Problem: [Spread beepers](#)

Logistics





How to get hold of me / get help+

- The [section forum](#), 24 hr turnaround
- Email: boresurajit@fhda.edu, 24 hr turnaround
- Office hours:
 - On campus: Tuesdays 12:00 noon to 1:30 pm, room 4218 in the STEM center. Entry is from room 4213
 - By appointment on Zoom
- Other resources:
 - Contact Lane via Canvas
 - [Online](#) or [in-person](#) tutoring via the STEM center (Room 4213)





Quick Check-in



Let's quickly go round the room! Pick any one of the following to answer:

- What has been your favorite Karel problem to work on so far?
- What have you found most surprising about the class so far?
- What have you found most difficult about the class so far?



Any Questions?


Flashback to last week ...



Functions



Given a problem to solve:

- Start with the big picture
 - Break the problem down into smaller, self-contained building blocks
 - These smaller building blocks are functions
 - The process of breaking down the problem into functions is **decomposition**
 - Any set of steps that will need to be repeated is a good candidate for a function
 - So is any logically self-contained portion of the problem
 - Assume the building blocks are done (use **pass** keyword)
 - Assemble building in **main** to solve the big problem
 - Implement each building block!
- 



Lecture Review





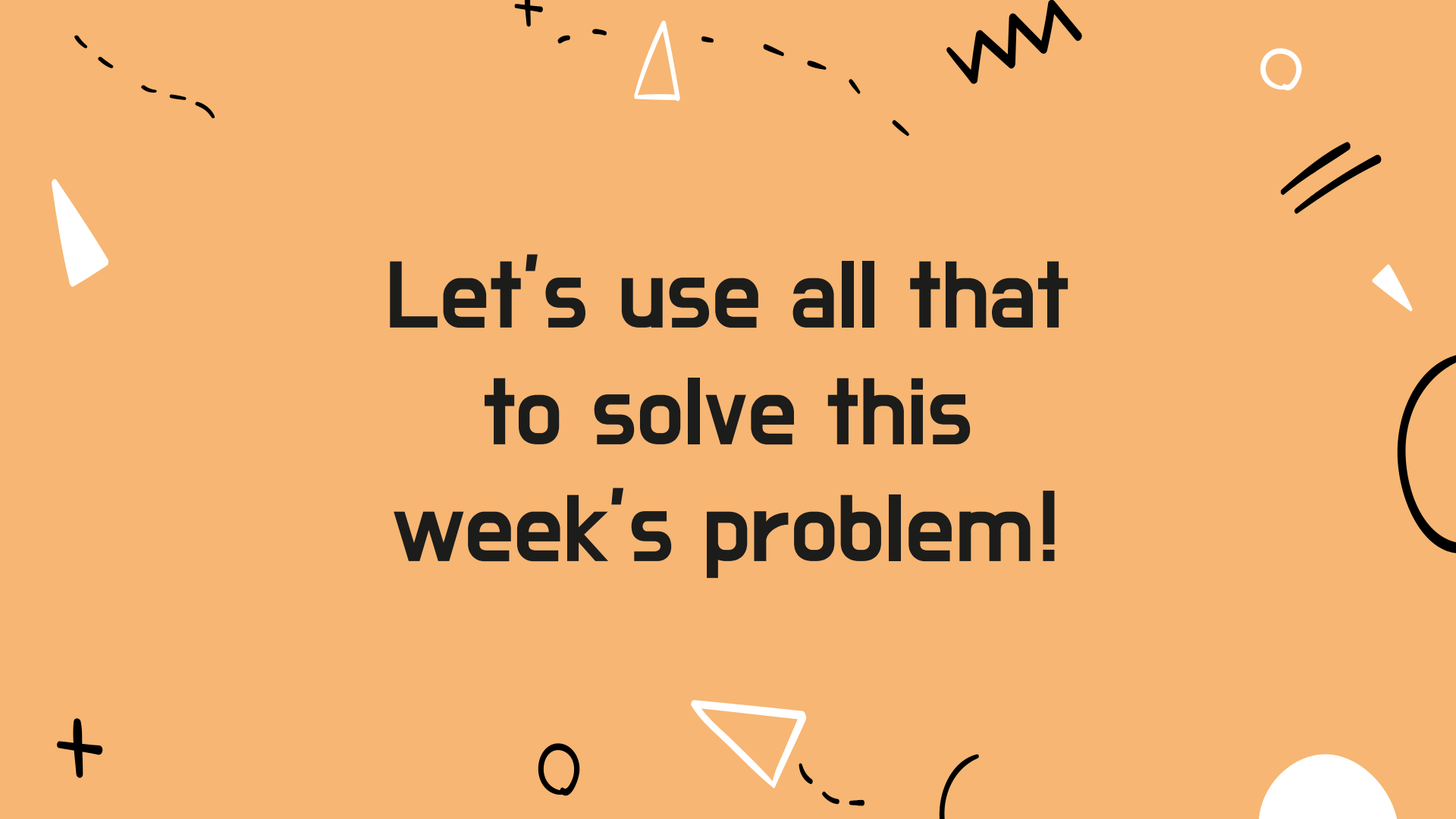
What We've Learned This Week

Before we get into our sample problem for today, let's review a bit. This week we've learned:

- Decomposition, aka stepwise refinement or top-down design
 - Breaking down the problem into smaller chunks
- Solving for the general case
- Iterative testing at each step
- Elements of good code style
 - Function names: short, descriptive, **in_snake_case**
 - Comments
 - Length of code
 - Indentation levels

The background is a solid orange color. It is decorated with various hand-drawn geometric shapes in white and black. These include: a dashed line in the top left; a white triangle in the top center; a black zigzag line in the top right; a white circle in the top right; two parallel black lines in the top right; a white triangle in the top right; a large black circle in the bottom right; a white circle in the bottom right; a black plus sign in the bottom left; a white circle in the bottom center; a white triangle in the bottom center; and a black plus sign in the bottom center.

Any Questions?

The background is a solid orange color. It is decorated with various white and black geometric shapes and symbols. In the top left, there is a dashed line and a small white triangle. In the top center, there is a dashed line with a small white triangle and a black zigzag line. In the top right, there is a small white circle and two parallel black lines. In the middle right, there is a small white triangle and a large black arc. In the bottom left, there is a black plus sign. In the bottom center, there is a small white circle and a white triangle with a dashed line. In the bottom right, there is a black arc and a large white circle.

**Let's use all that
to solve this
week's problem!**



Section problem: Spread Beepers

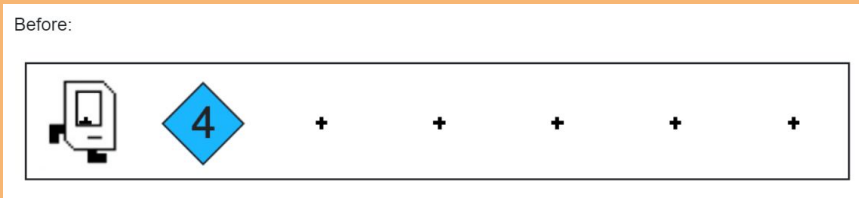
<https://codeinplace.stanford.edu/foothill-cs49/ide/a/spreadbeepers>



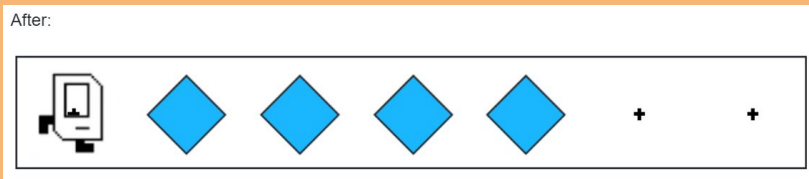


Spread Beepers

- Karel stands directly in front of a pile of a certain number of beepers:



- Those beepers should be spread out across the row:



- The number of beepers is the same before and after they are spread out



Key Details (Assumptions)

- Karel has an infinite number of beepers in its bag!

This affects what solution you can use.

- There is only one row in the world
- The pile of beepers is in the second column



The background is a solid orange color. It is decorated with various hand-drawn geometric shapes in white and black. These include a dashed line in the top left, a white triangle in the top center, a black zigzag line in the top right, a white circle in the top right, two parallel black lines in the top right, a white triangle in the top right, a black plus sign in the bottom left, a white circle in the bottom center, a white triangle in the bottom center, a black plus sign in the bottom center, a black circle in the bottom center, and a white circle in the bottom right.

Questions Before We Begin?

The background is a solid pink color. It is decorated with various white and black geometric shapes and symbols. In the top left, there is a dashed white line and a solid white triangle. In the top center, there is a dashed black line with a small black triangle on it. In the top right, there is a solid black zigzag line, a small white circle, and two parallel black lines. In the middle right, there is a small white triangle. In the bottom left, there is a solid black plus sign. In the bottom center, there is a small black circle and a solid white triangle. In the bottom right, there is a solid black circle and a large white circle. The text "Let's get to work!" is written in a bold, white, sans-serif font in the center of the image.

Let's get to work!

Tackling the big picture



Given a problem to solve:

- What is the current situation? : **Preconditions**
 - Note the assumptions in the section handout
- What is the expected outcome? : **Postconditions**
- What steps are needed to get from preconditions to postconditions?
 - Not “how” but “what”; do not worry about the implementation yet
- Do the steps need to be repeated?
 - A certain number of times (**for** loop)
 - As long as a condition is true (**while** loop)
- Tell me what comments and statements to type in the IDE!



Tackling the building blocks




For each building block: **iterate** and **refine**

- What is the current situation? : **Preconditions**
- What is the expected outcome? : **Postconditions**
- What steps are needed to get from preconditions to postconditions?
 - Implement steps and/or decompose further: **iterate**
 - After each block is built, test it and make improvements: **refine**
- Do the steps need to be repeated?
 - A certain number of times (**for** loop)
 - As long as a condition is true (**while** loop)
- Tell me what comments and statements to type in the [IDE](#)!

Bonus! Extending the problem



Typically*, programs should be written to tackle the general case, not the specific case.

- What if the pile of beepers were anywhere in the row, not just on the second corner? (Assume that the row is still wide enough to spread out the number of beepers in the pile.)
 - What if there were multiple rows?
- 

Try it [here](#).

*If we have certain guarantees, then we need not over-engineer the solution to fit the general case: a program to run a task on the last days of January, March, May, July, September, and November need not check if it's a leap year.

The background is a solid pink color. It is decorated with various hand-drawn geometric shapes in white and black. These include a dashed line in the top left, a white triangle in the top center, a black zigzag line in the top right, a white circle in the top right, two parallel black lines in the top right, a white triangle in the top right, a black circle in the bottom right, a white circle in the bottom right, a black plus sign in the bottom left, a white circle in the bottom left, a white triangle in the bottom left, and a black line in the bottom left.

Goodbye, Karel!

Next up: Console Programming!