

# For Wednesday

- Read Becker, chapter 4, sections 4-5
- Recommended practice problems:
  - chapter 4, problems 5-10

# Program 2

- Any questions?

Questions before the quiz?

# Quiz

# Helper Methods

- What is a helper method?
- Who should call a helper method?
- How do we enforce that?

# Access Modifiers

- **Public**
  - Access from anywhere
  - Use for the services a robot provides to others
- **Protected**
  - Access from methods in the class or its subclasses
  - Use for methods that may be overridden but don't need to be public
- **Private**
  - Access from only methods in the class
  - Use for anything that doesn't need to be public or protected

# Control Structures

- Allow programs to make decisions
- Why the name “control structure”?

# Motivation

- So far, we've written programs that
  - executed one statement after another (pattern?)
  - executed all statements in a method and returned
- Some problems can't be solved this way
  - Have a robot move forward until it reaches a wall.
  - Have a robot pick up things from intersections when not all intersections have things on them
- Other examples?
  - Robot

# Different Kinds of Decisions

- Should this group of statements be executed once, or not at all?

```
if (karel.canPickThing())  
{  
    karel.turnLeft();  
}  
karel.move();
```

- Should this group of statements be executed again?

```
while (karel.canPickThing())  
{  
    karel.turnLeft();  
}  
karel.move();
```

# Tracing an if statement

```
if (karel.frontIsClear())  
{  
    karel.move();  
}  
karel.turnLeft();
```

# Tracing a while statement

```
while (karel.frontIsClear())  
{  
    karel.move();  
}  
karel.turnLeft();
```

# What's the Pattern?

# Pseudocode and Decisions

# Questions Robots Can Ask

- Can I pick up a Thing from this intersection?
- How many Things are in my backpack?
- Is the path in front of me clear?
- What avenue am I on?
- What street am I on?
- What direction am I facing?
- What is my speed?
- What is my current label?

## Robot

int street  
int avenue  
Direction direction  
ThingBag backpack

+Robot(City aCity, int aStreet, int anAvenue,  
          Direction aDirection)  
+boolean canPickThing()  
+int countThingsInBackpack()  
+boolean frontIsClear()  
+int getAvenue()  
+Direction getDirection()  
+String getLabel()  
+double getSpeed()  
+int getStreet()

# Predicates

- Questions with the answers yes or no (actually true or false)
- Can be directly used in if and while statements.
- Return values of type boolean.

# Negatives

- Sometimes we want to test the opposite of a predicate
- Use the symbol !

# Integer Queries

```
if (karel.getStreet() == 1)
{
    karel.turnAround();
}
```

```
while (karel.countThingsInBackpack < 4)
{
    karel.pickThing();
}
```

# Comparison Operators

- $==$
- $!=$
- $>$
- $<$
- $>=$
- $<=$

# Practice

- Write code to move two spaces forward, picking up any things in any space the robot enters.

# Problem

- Write a method to make a Robot go completely around the inside of a box created by walls.