

# CSE 113 A

February 8-12, 2010

## Announcements

- ⚙ Pick up your Exam 1 papers  
(before/after class or in Adrienne's  
office hours)
- ⚙ Lab 1 assessment this week – you  
need to report at the time given in the  
email sent on Friday 2/5.
- ⚙ Lab 2 will begin week of 2/15 in  
recitation.



## Methods

### ① calling methods

↳ asking the method to perform its job

- move
- turn
- eat

3



### ② writing methods

↳ Defining what a method should do when it is called.

4



Calling

Syntax of method call:

nameOfMethod (            );  
   argument list

5

Writing/defining a method:

Method header/signature

public returnType nameOfMethod (            )  
   parameter list

Method body -  
 enclosed in { } (and includes { } )

6



Method body defines how the method works

↳ can call other methods

↳ if-statements

⋮

7



## Greenfoot Class Documentation

- ⊗ Available from inside Greenfoot
- ⊗ Information about classes built-in that you may want to use/have used in the past
  - ⊗ Actor
  - ⊗ World
  - ⊗ Greenfoot

8



# Constructors

- ⊗ Constructors are special methods that are called every time an object is created – they set up the initial state of our objects.
- ⊗ Explicit constructors (ones that you can see in the source code) look like this:

```
public NameOfClass()  
{  
}  
}
```

9



# Constructors

- ⊗ A constructor has the same name as the name of the class.
- ⊗ It does not have a return type.
- ⊗ If there is no explicit constructor in the source code for a class, Java provides an implicit one that you do not see in the source code, but is inserted at compile time.

10



## CarWorld Class

- ⊗ Looking at the constructor of CarWorld, we can see a method call that looks like this:

```
super(x, y, z)
```

- ⊗ Here, we are not calling a method called super, but rather super is a keyword that indicates the superclass. In this case, we are calling the superclass' constructor.

11



## Adding objects at startup

- ⊗ We can add objects to the world when it is created by calling the addObject method from the world.

- ⊗ Example

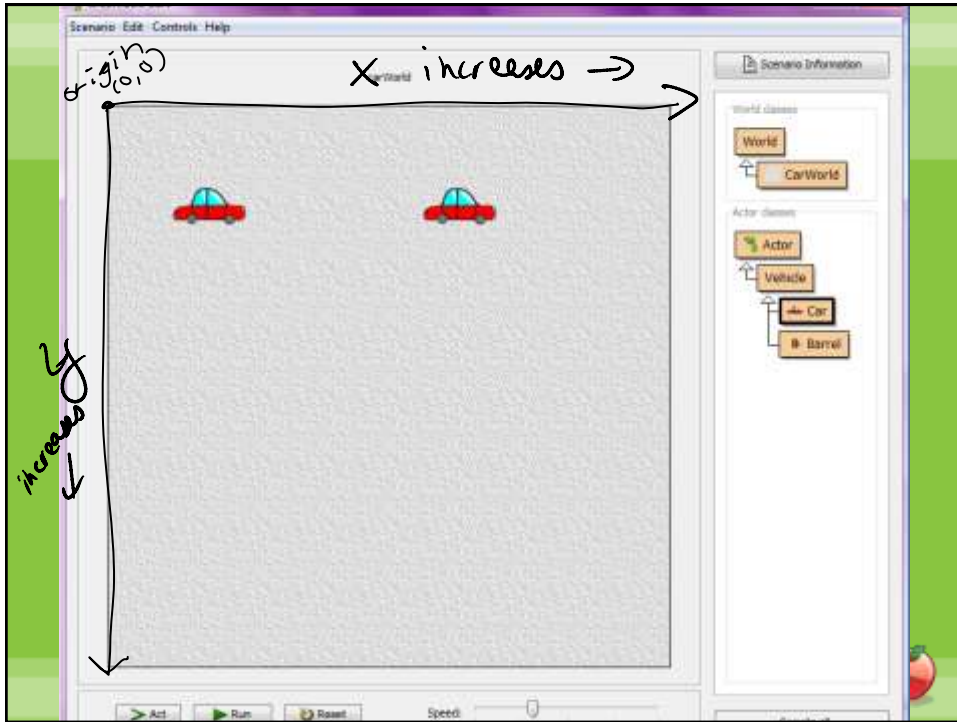
```
addObject(new Car(), 34, 56);
```

- ⊗ Note that we need to create a new Car object to add by using the expression new Car(). This expression creates an object and calls the constructor of that object.

- ⊗ The numbers that follow are the x and y coordinates of where we would like the object to be in the world.

12





If we want to store information inside our actors, we need to use variables.

Variables are used in programs to store information.



Instance variables are used to keep track of "class-level" information. Information that the object is holding onto.

Syntax (for instance variable):

```
private    type    identifier;
                ||
                "name"
```

15



private vs. public

private  
 ↓  
 accessible only within the same class

public  
 ↓  
 allows anyone to have access (anyone = from somewhere else in our program)

16





All variables in Java are given a type.

The type tells us what "kind" of values we can store.



17

## Variables

- ⊗ After we declare the instance variables, it is good practice to give it an initial value.
- ⊗ We would give an instance variable and initial value in the constructor of the class.
- ⊗ Example

```
_barrelsHit = 0;
```

- ⊗ Note that this expression uses the assignment operator (=) and takes the values on the right hand side and assigns them to the variable on the left hand side.



18

# Moving Actors

- ⚙ If we create direct sub-classes of Actor, we do not have a “move” or “turn” method.
- ⚙ The Feb 12 version of the scenario shows how to begin to teach Actors how to move on their own.

