

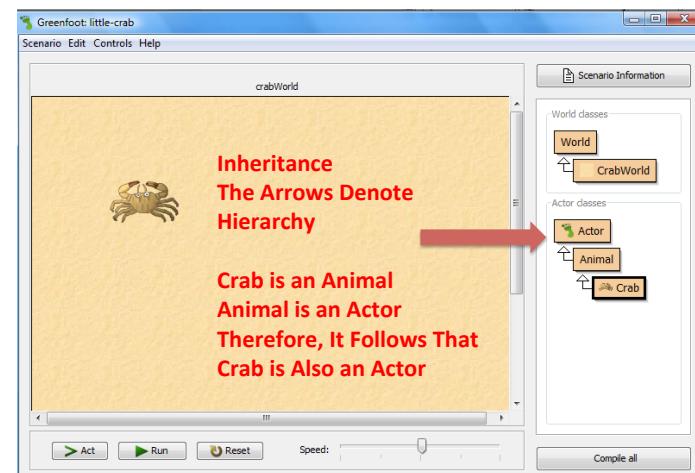
## 2.1 Little Crab Scenario

### Chapter 2 - The First Program: Little Crab

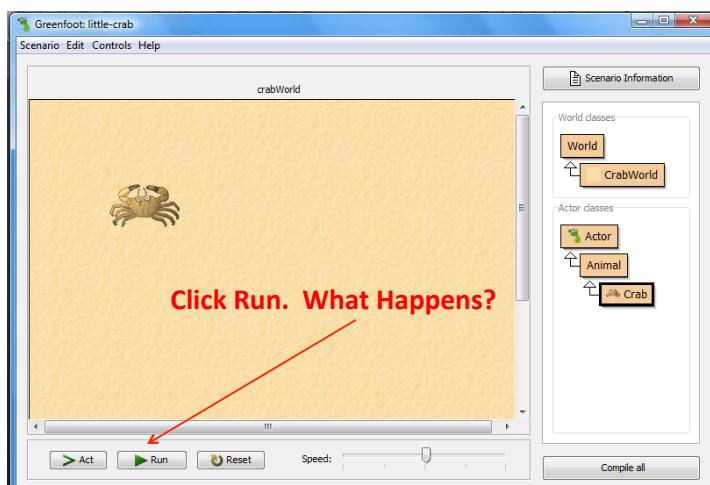
Original slides by Bruce Chittenden

Exercise edits by Scott Blanch

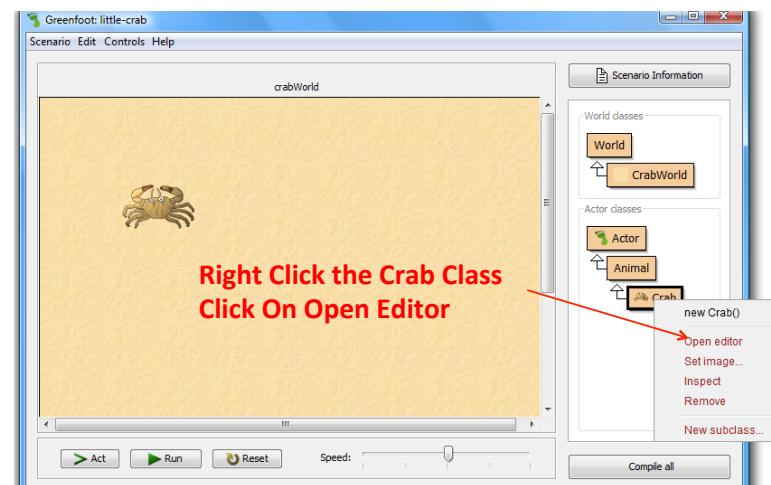
Exercise setup: Open *little-crab* from the book scenarios folder – chapters2-4.



### Exercise 2.1



### Exercise 2.1



## Exercise 2.1

The screenshot shows the Greenfoot source code editor window for the 'Crab' class. The code defines a crab that extends the 'Animal' class. The 'act()' method is currently empty. A red arrow points from the text 'When Run is Clicked the Crab does nothing' to the 'act()' method. Another red arrow points from the text 'This is because there is no Source Code in the act method for the Crab.' to the same method.

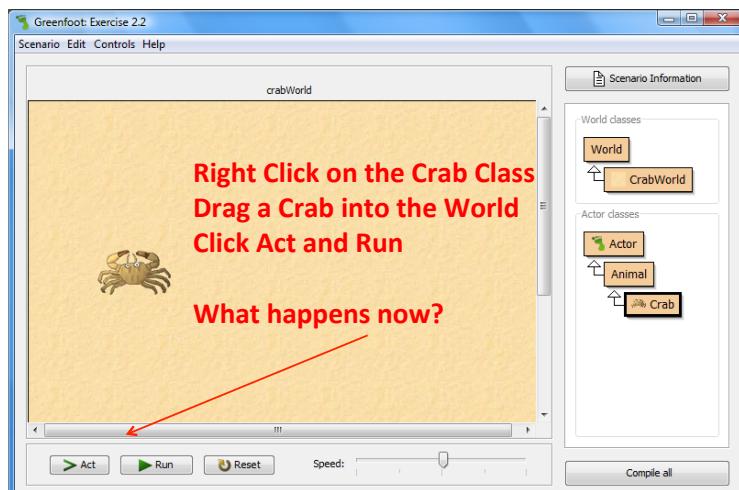
```
Crab
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Find Next Close Source Code
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)
/**
 * This class defines a crab. Crabs live on the beach.
 */
public class Crab extends Animal
{
    public void act()
    {
    }
}
```

## 2.2 Making the Crab Move

```
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)
/*
 * This class defines a crab. Crabs live on the beach.
 */
public class Crab extends Animal
{
    public void act()
    {
        // Add your action code here
    }
}
```

Replace the entire //comment line with  
move();  
Click Compile

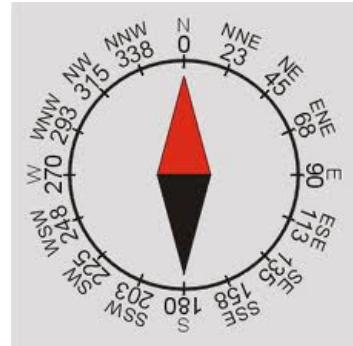
## Exercise 2.2



## Exercise 2.3



## 2.3 Turning in Degrees – 0 to 360 starting from due north



```
public void act ()
{
    turn (5);
}
```

## Exercise 2.4

The screenshot shows the Greenfoot editor with the 'Crab' class selected. The code is as follows:

```
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)

/**
 * This class defines a crab. Crabs live on the beach.
 */
public class Crab extends Animal
{
    public void act()
    {
        move(5);
        turn(5);
    }
}
```

A red arrow points to the 'turn(5);' line with the instruction: 'Change move(); to turn(5); and recompile. Run it. What happens?' Below the code, a message says: 'Can you make it turn left?'

## Exercise 2.6

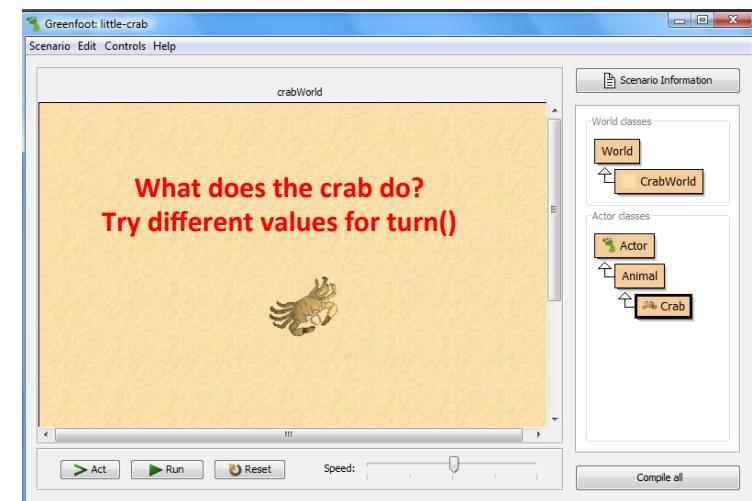
The screenshot shows the Greenfoot editor with the 'Crab' class selected. The code is identical to Exercise 2.4:

```
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)

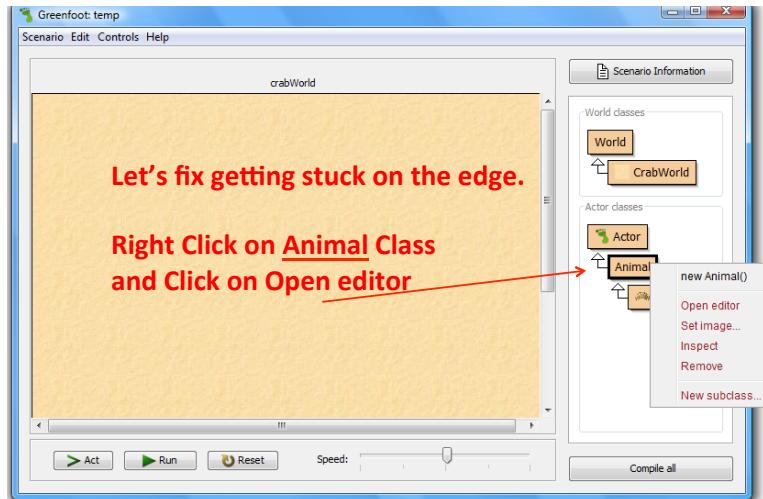
/**
 * This class defines a crab. Crabs live on the beach.
 */
public class Crab extends Animal
{
    public void act()
    {
        move();
        turn(5);
    }
}
```

A red arrow points to the 'turn(5);' line with the instruction: 'Try both together like this – move(); turn(5);'. Below the code, a message says: 'What does the crab do?'

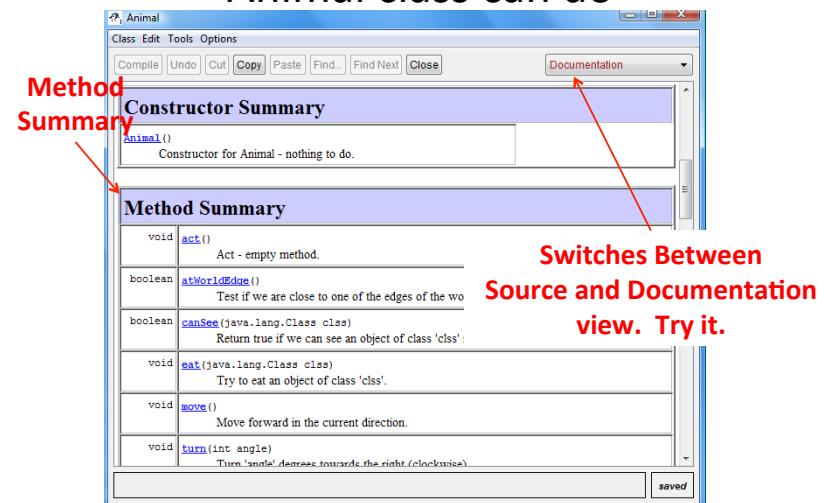
## Exercise 2.6



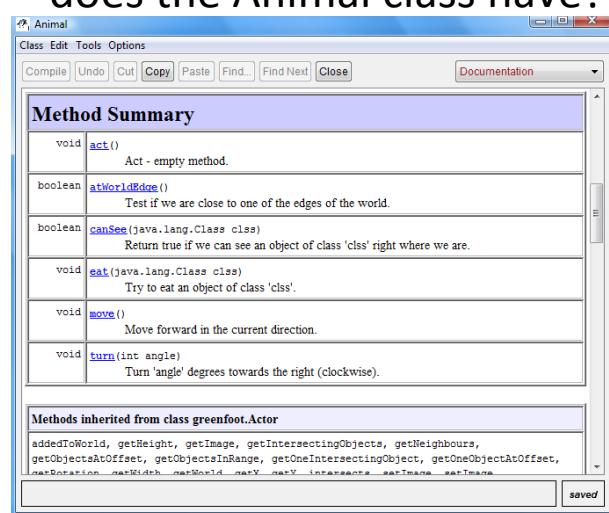
## 2.4 Dealing with Screen Edges



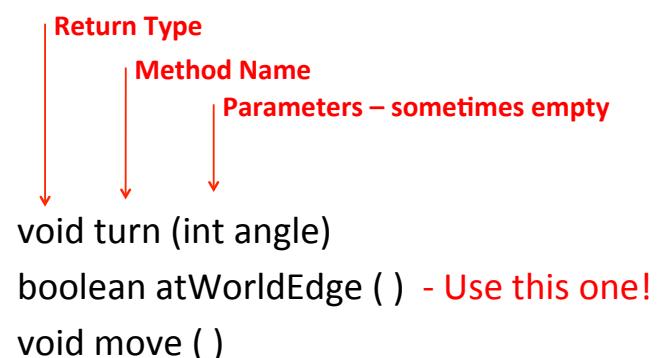
Documentation View – See what the Animal class can do



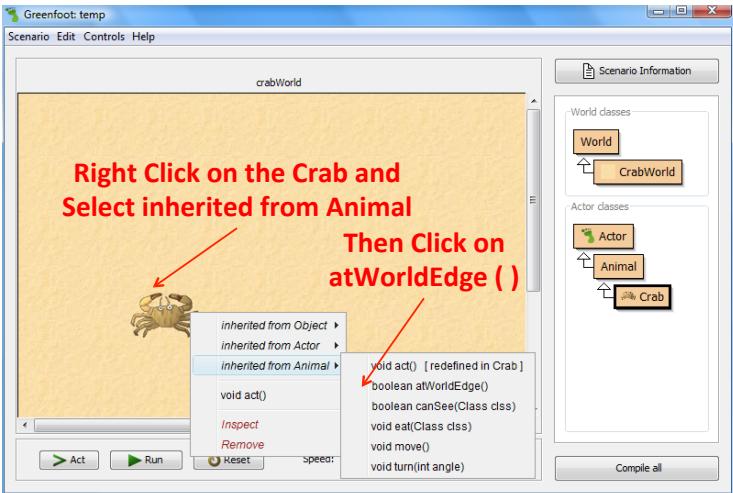
Exercise 2.9 – How many methods does the Animal class have?



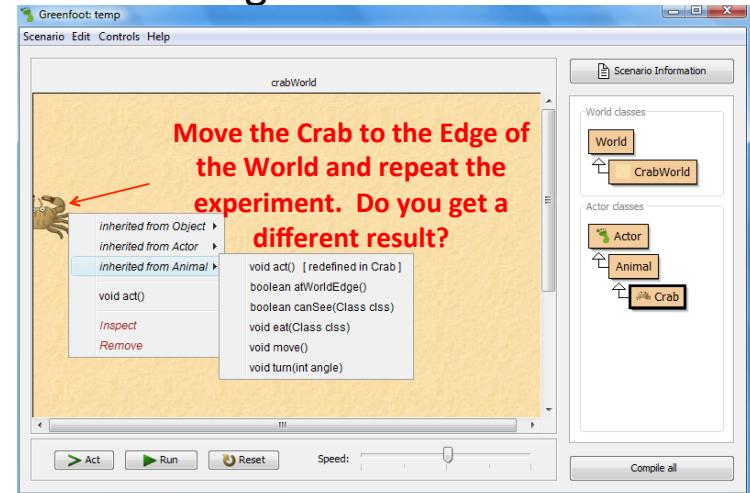
Method Signatures - Most methods have these three parts in their definition.



## Exercise 2.10 – Execute atWorldEdge. What is the result?



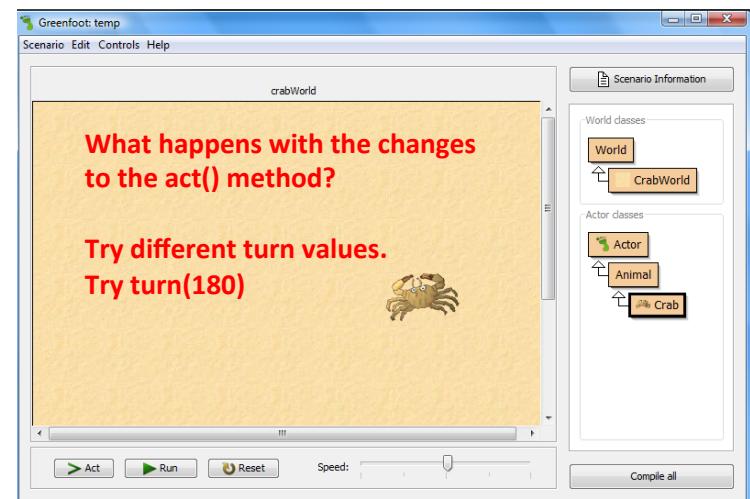
## Exercise 2.11 – Run atWorldEdge again at the edge. What is the result?



## Exercise 2.12 – Edit your Crab act() method to look like this. We want it to turn only at the edge.

```
public class Crab extends Animal
{
    public void act()
    {
        if ( atWorldEdge () )
        {
            turn (17);
        }
        move ();
    }
}
```

## Exercise 2.12



## Exercise 2.14

The screenshot shows the Greenfoot IDE with the 'Crab' class selected. The code defines a crab that moves and turns at the edge of the world:

```
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)

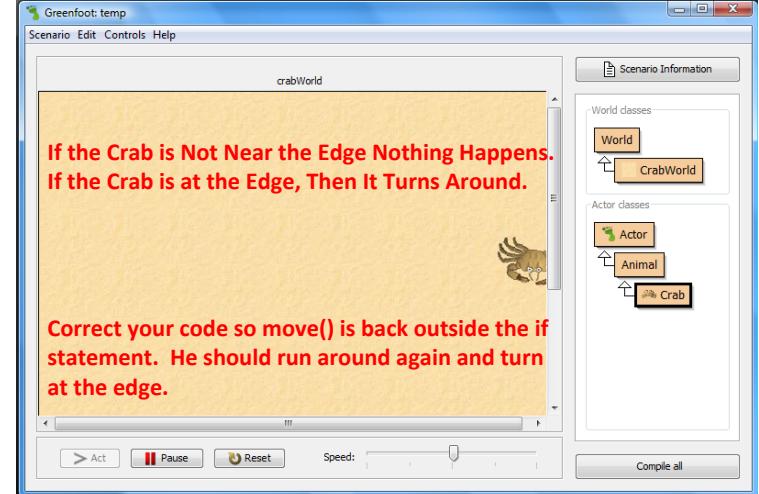
/**
 * This class defines a crab. Crabs live on the beach.
 */
public class Crab extends Animal
{
    public void act()
    {
        if (atWorldEdge())
        {
            turn(17);
            move(); ←
        }
    }
}
```

A red arrow points from the text "Place move () inside the if statement with turn(). What happens? Why?" to the line "move();".

Place move () inside the  
if statement with turn().  
What happens? Why?

Class compiled - no syntax errors

## Exercise 2.14



## 2.5 Summary of Programming Techniques

In this chapter, we have seen how to call methods such as move() and turn(x), with and without parameters. This will form the basis for all further Java Programming.

We have encountered a glimpse of inheritance. Classes inherit the methods from their superclasses. Crab inherits methods from Animal.

And, very important, we have seen how to make decisions. We have used an if-statement for conditional execution.