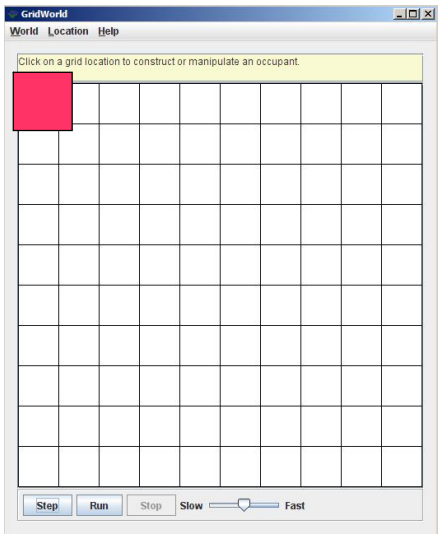

GridWorld

What is GridWorld?



Row = 0

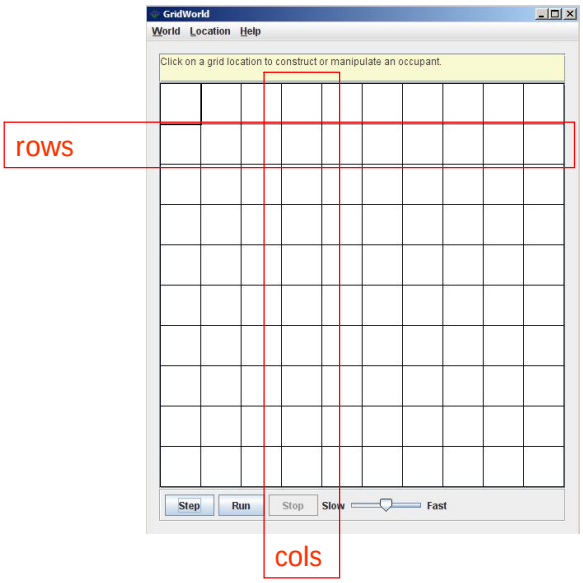
Column = 0

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

What is GridWorld?



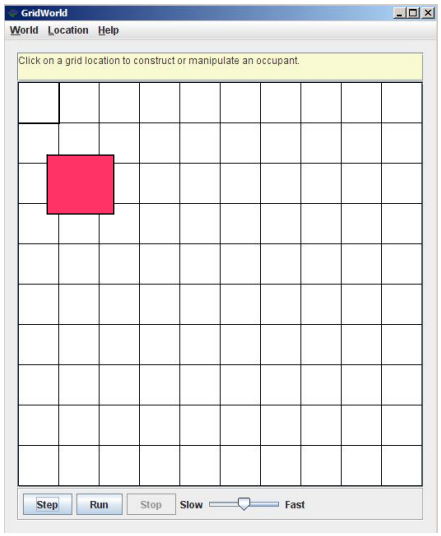
A grid is a structure that has rows and columns.

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

What is GridWorld?



Row = 2

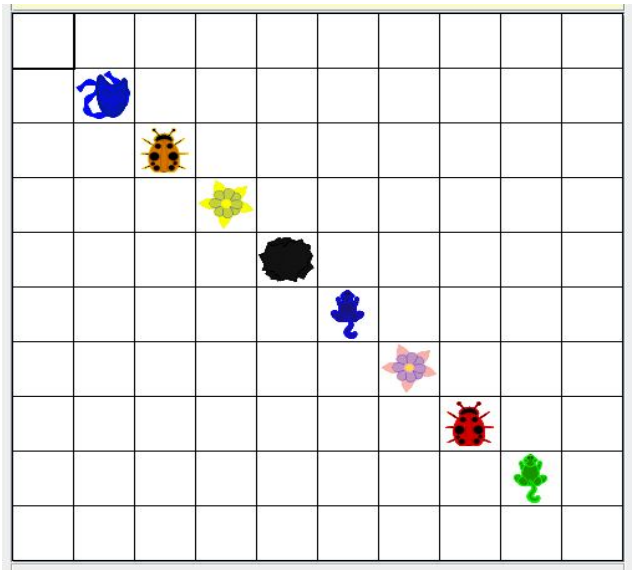
Column = 1

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

What is GridWorld?



GridWorld Classes


ActorWorld
© A+ Computer Science - www.apluscompsci.com

ActorWorld

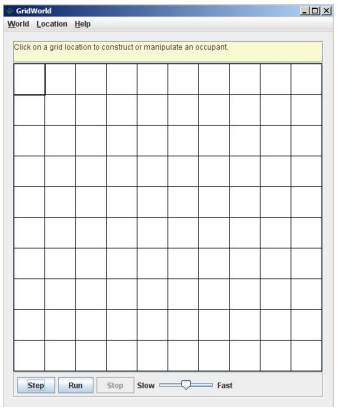
frequently used methods

Name	Use
ActorWorld()	creates a new world using 10X10 grid
ActorWorld(grid)	creates a new world using grid
add(loc, thang)	add thang at spot loc
add(thang)	add thang at a random empty loc
show()	makes the world visible

```
import info.gridworld.actor.World;
```

ActorWorld

```
ActorWorld earth = new ActorWorld();
earth.show();
```



ActorWorld is used to graphically display a grid of Actors.

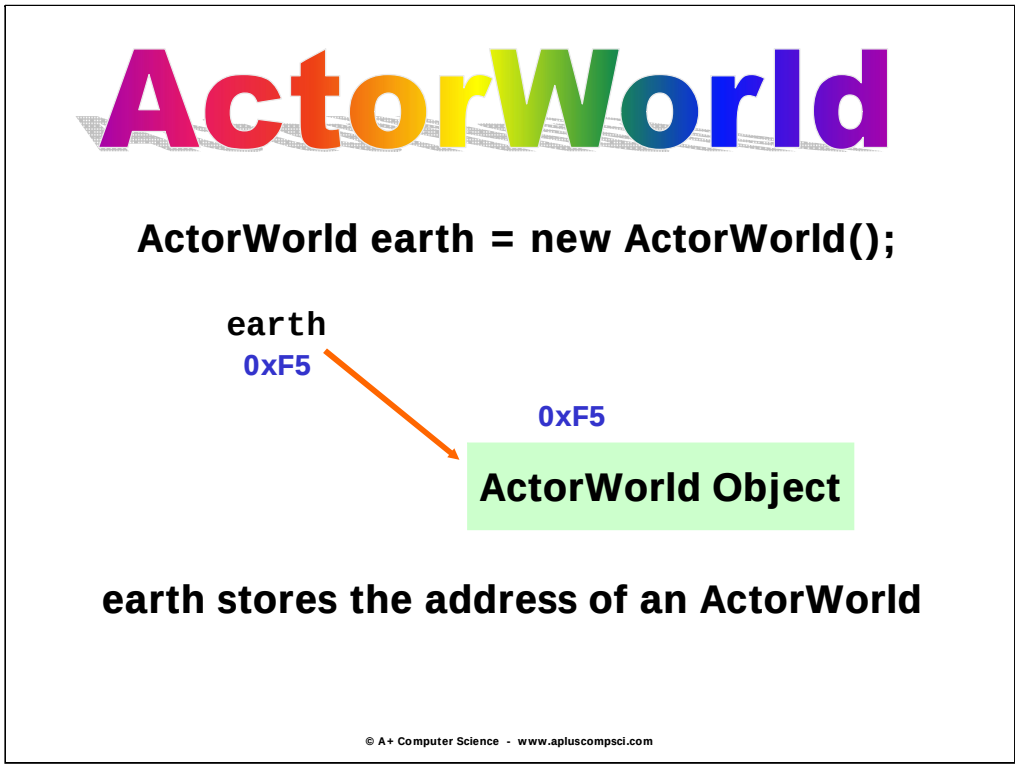
ActorWorld

A reference variable stores the memory address of an object.

```
ActorWorld earth = new ActorWorld();
```

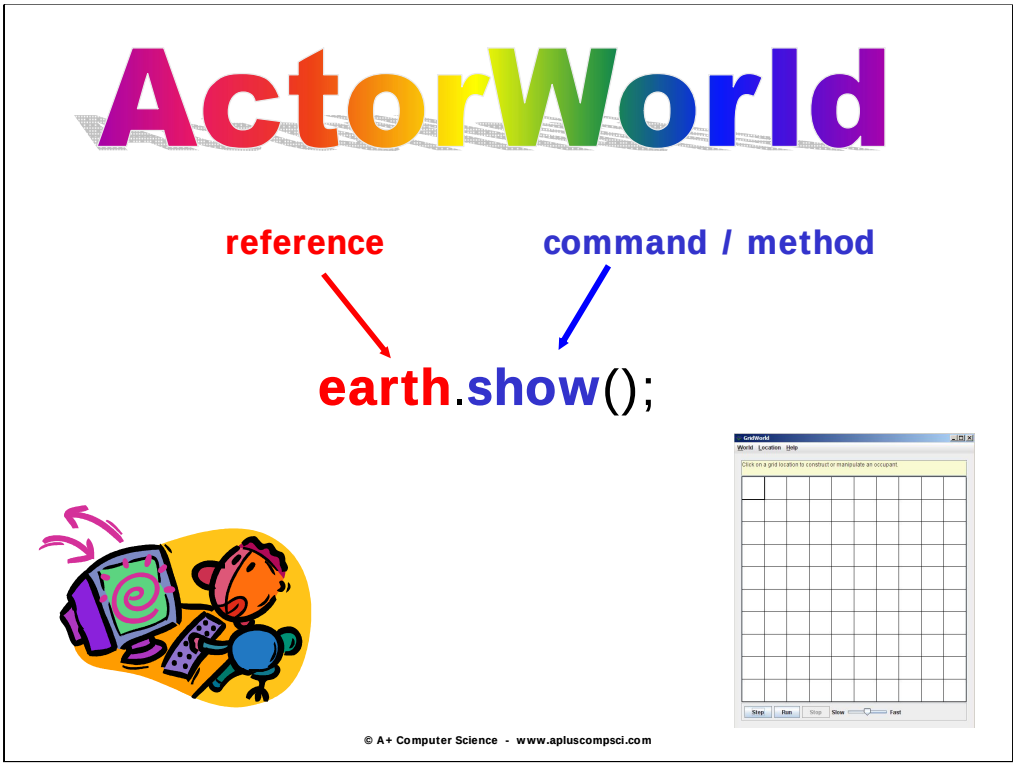
A reference variable is used to store the location of an Object. In most situations, a reference stores the actual memory address of an Object.

`earth` stores the location / memory address of a new ActorWorld object.



A reference variable is used to store the location of an Object. In most situations, a reference stores the actual memory address of an Object.

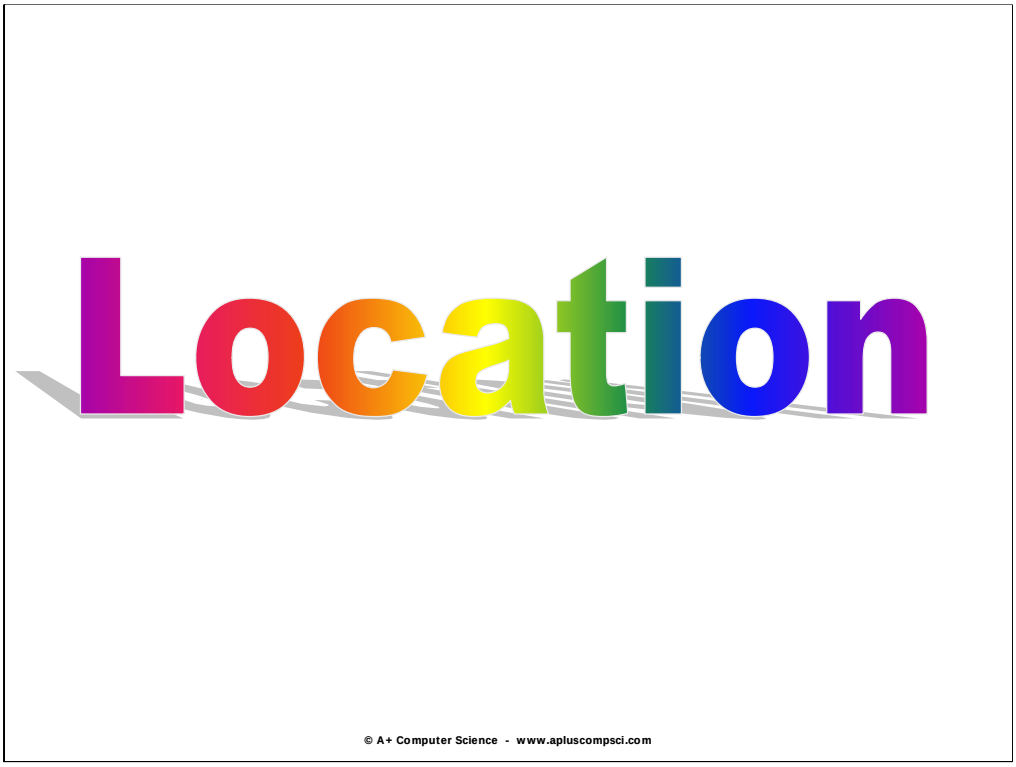
`earth` stores the location / memory address of a new ActorWorld.



earth is a reference to an ActorWorld.

`earth.show()` calls the ActorWorld show method to show the grid of actors graphically.

open gridworldone.java


Location
© A+ Computer Science - www.apluscompsci.com

Location

frequently used methods

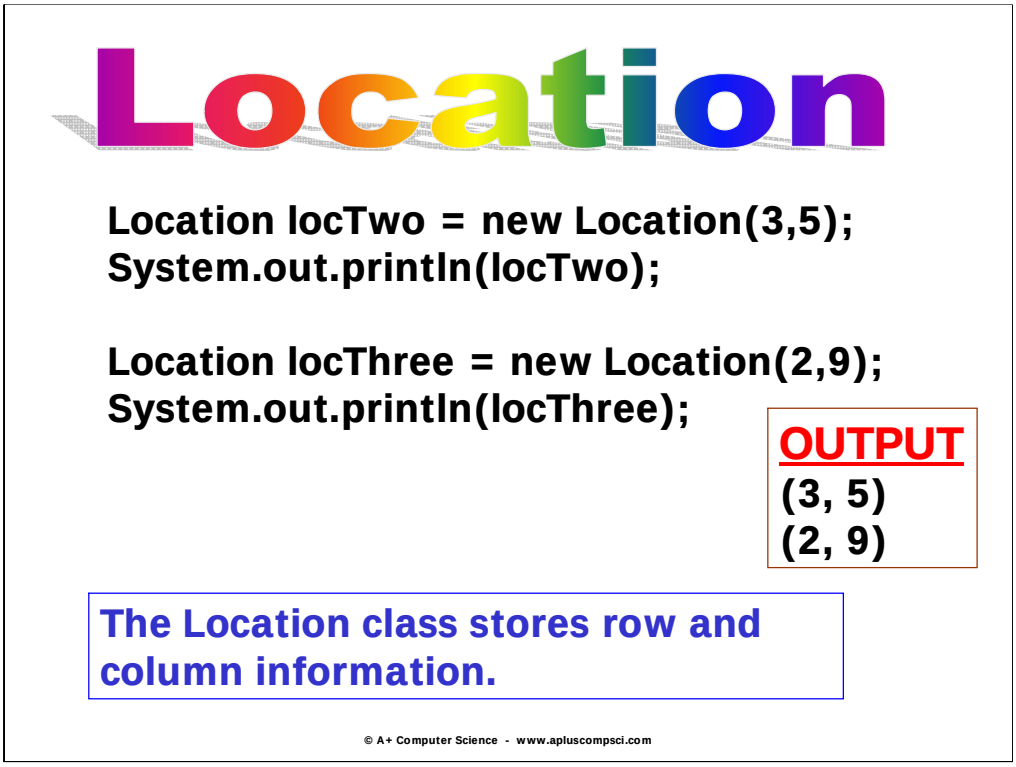
Name	Use
Location(row, col)	creates a new row,col Location
getCol()	gets the column value for this location
getRow()	gets the row value for this location

```
import info.gridworld.grid.Location;
```


Location is a class that stores row and column information.

```
Location spot = new Location(4,5);
```

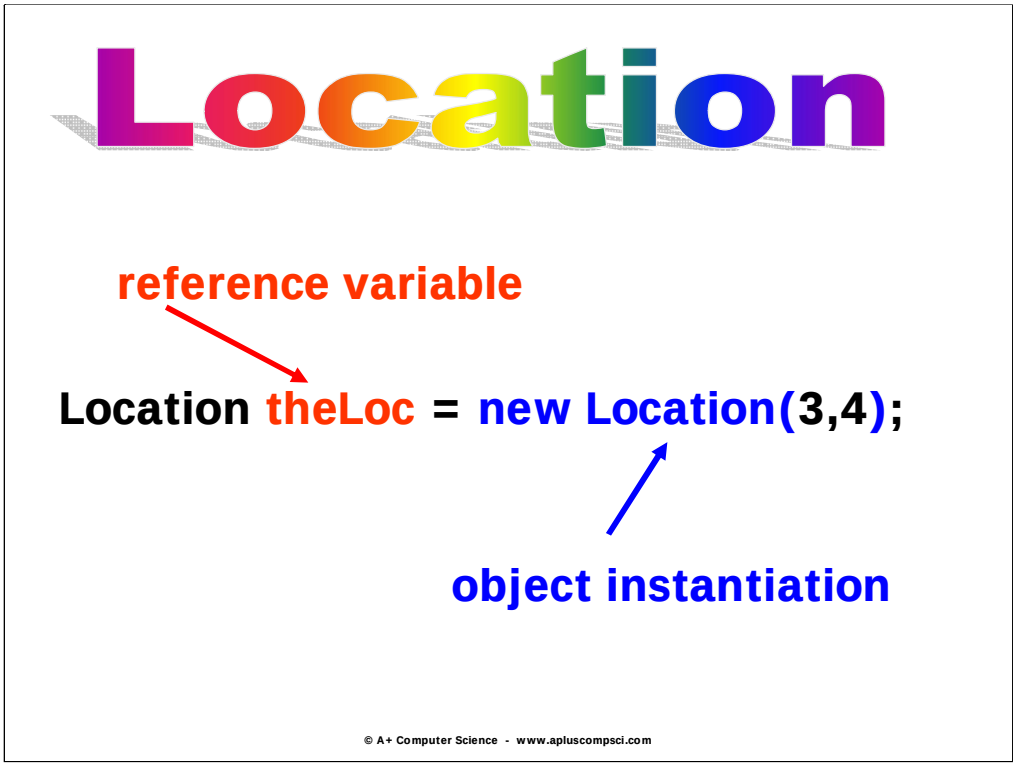
spot has a row value of 4 and a column value of 5.



Location is a class that stores row and column information.

```
Location spot = new Location(4,5);
```

spot has a row value of 4 and a column value of 5.



`Location` is a class which must be instantiated before it can be used. In other words, you must make a new `Location` if you want to use a `Location`. A reference must be used to store the location in memory of the `Location` object created.

A row value and a column value must be passed as parameters to the `Location` constructor.

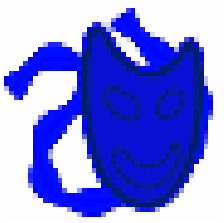
`theLoc` is a reference that will store the location/memory address of newly created `Location` object.

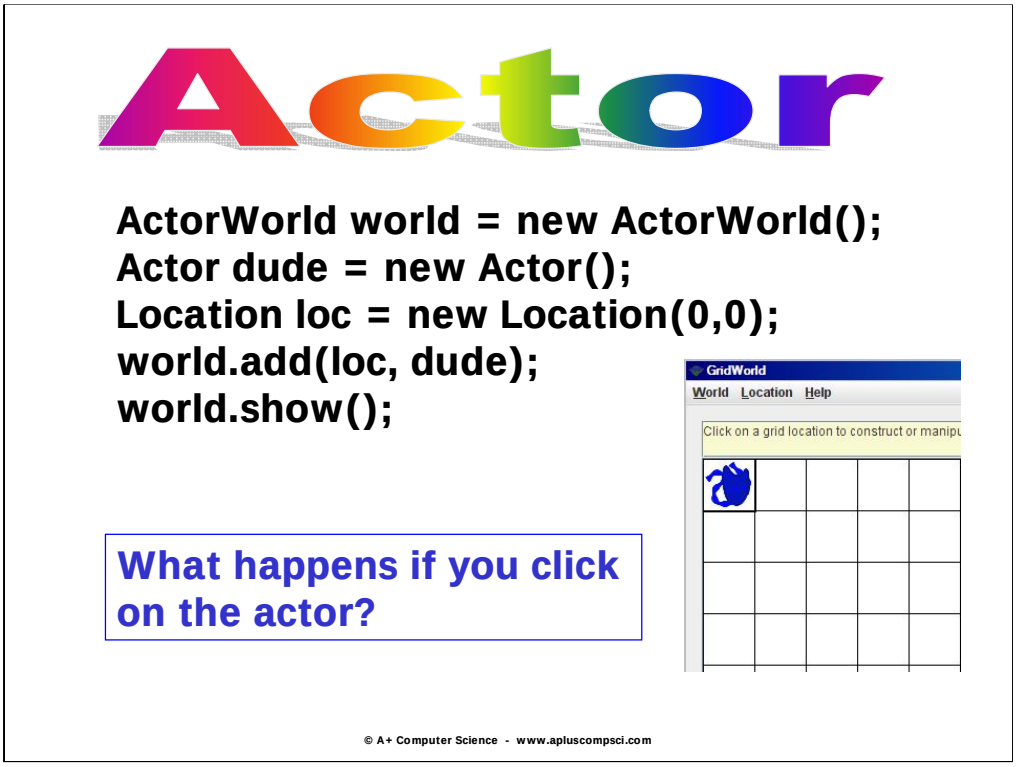
open locationone.java

Actor

Actor is the basic object from which all other GridWorld actors will be built.

Each of the new actors created will extend the original actor class.





The add method of class World receives a Location parameter and an Actor reference.

The Actor reference is stored in the grid at the specified Location.

The default color of an Actor is BLUE.

```
ActorWorld world = new ActorWorld();
Actor dude = new Actor();
Location loc = new Location(0,0);
world.add(loc, dude);
world.show();
```

Methods

reference

command / method

```
world.add(loc, dude);
```

Methods are used to tell an object what to do.

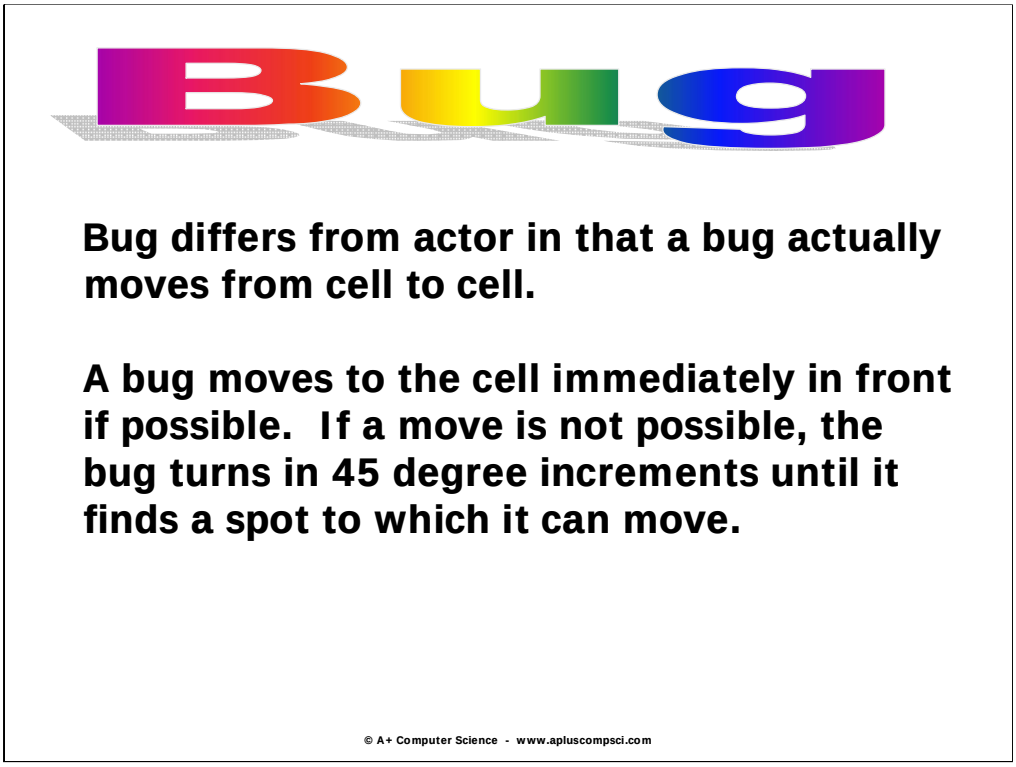
World is the reference to an world.

Add is a method that adds a reference at a particular location.

The add method is called to tell world to add dude at location loc.

open actorone.java


Bug
© A+ Computer Science - www.apluscompsci.com

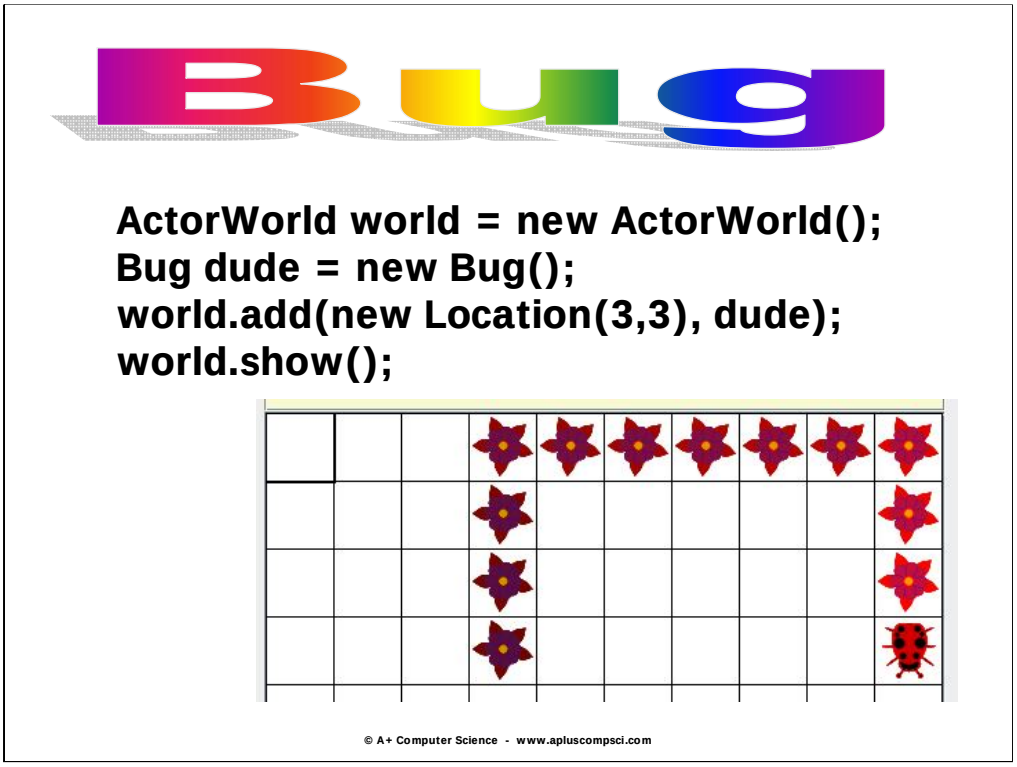


Bug is a suped up Actor.

Bug extends Actor.

Bug has two constructors, one of which takes a Color parameter.

Bug will move when its act method is called.

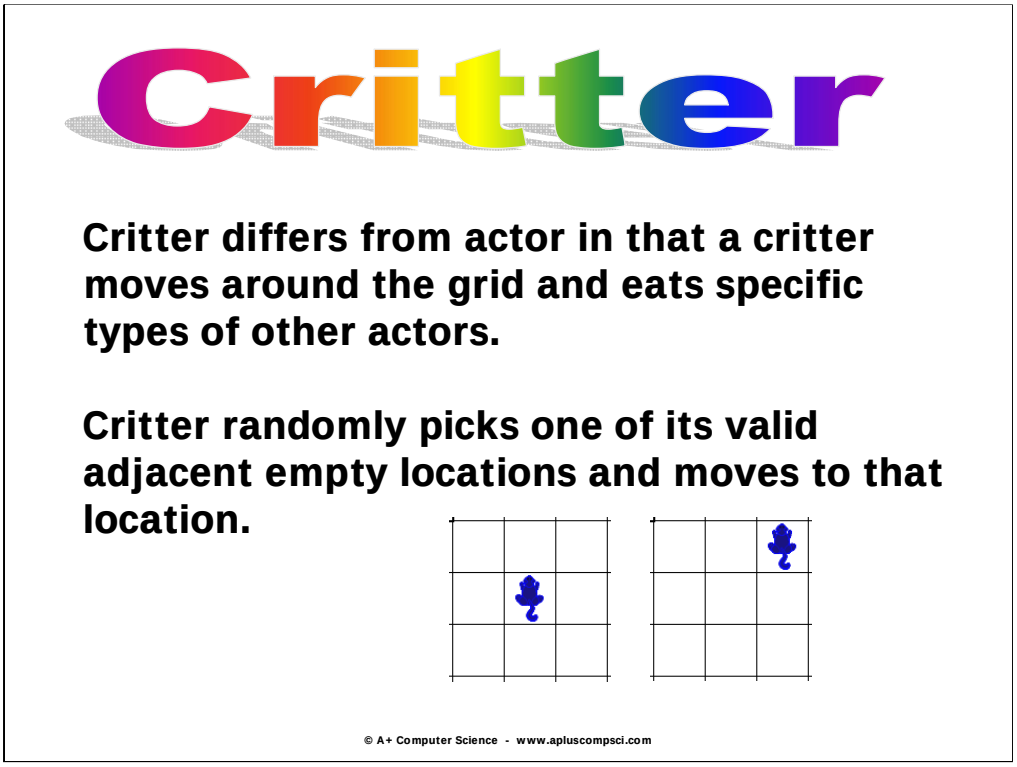


In this example, a default Bug is created.

A default bug is red and facing NORTH.

The bug moves NORTH until it reaches the top of the grid. The bug them turns in 45 degree increments until it finds an empty location to which to move.

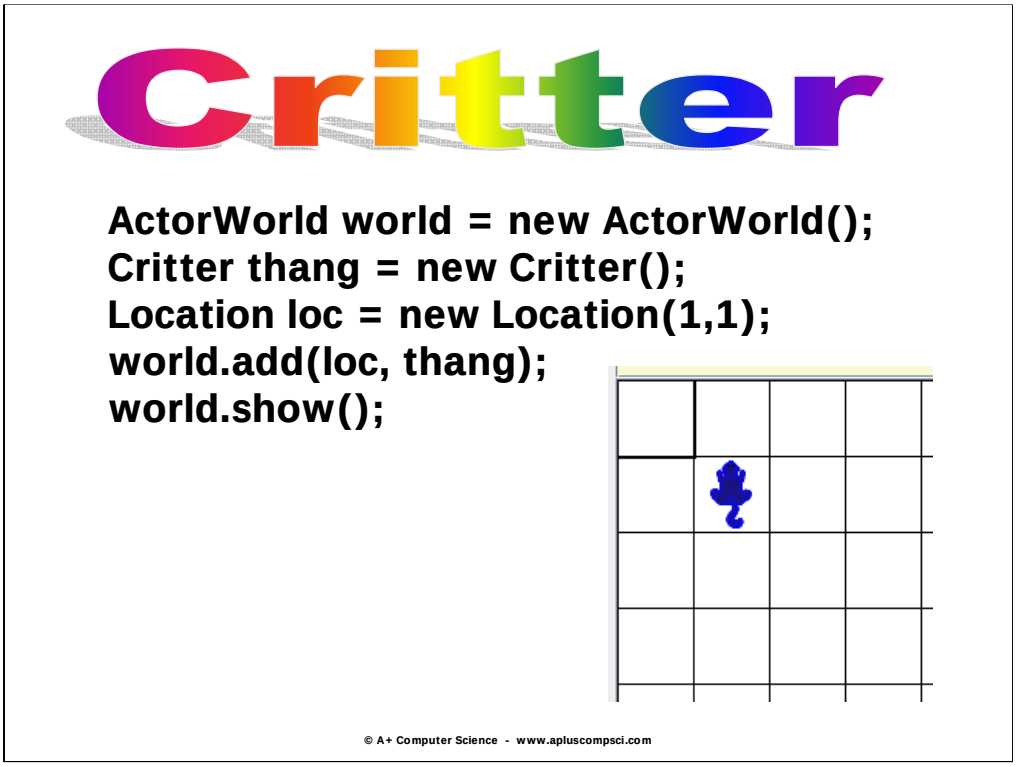
open bugone.java



Critter is a suped up actor that moves around the grid and eats other actors.

Critter will eat all neighbors that are not critters and not rocks.

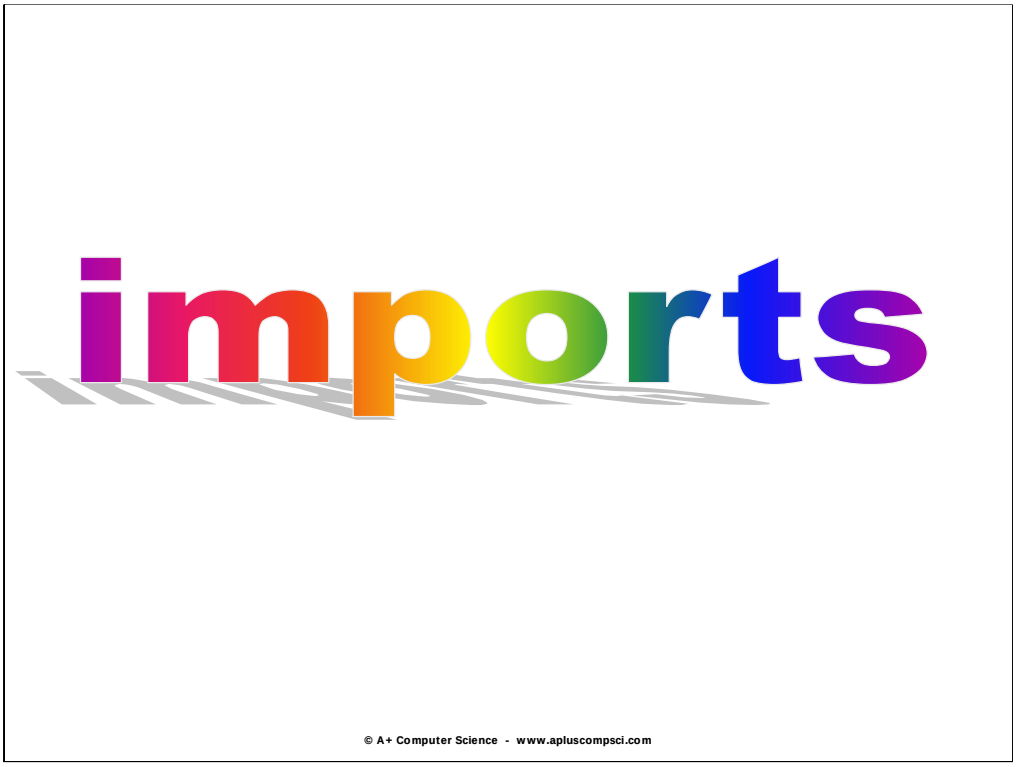
Critter randomly selects one of its valid empty adjacent locations. Critter moves to the new location.



//shortcut version of the code above

```
ActorWorld world = new ActorWorld();
world.add(new Location(1,1), new Critter());
world.show();
```

open critterone.java


imports
© A+ Computer Science - www.apluscompsci.com

Imports are used to tell Java where to find a class.

```
import info.gridworld.grid.Location;
```

```
import info.gridworld.actor.Rock;
```

```
import info.gridworld.actor.Flower;
```

```
import java.awt.Color;
```

open gridworldtwo.java


Start work
on ActorBox
© A+ Computer Science - www.apluscompsci.com