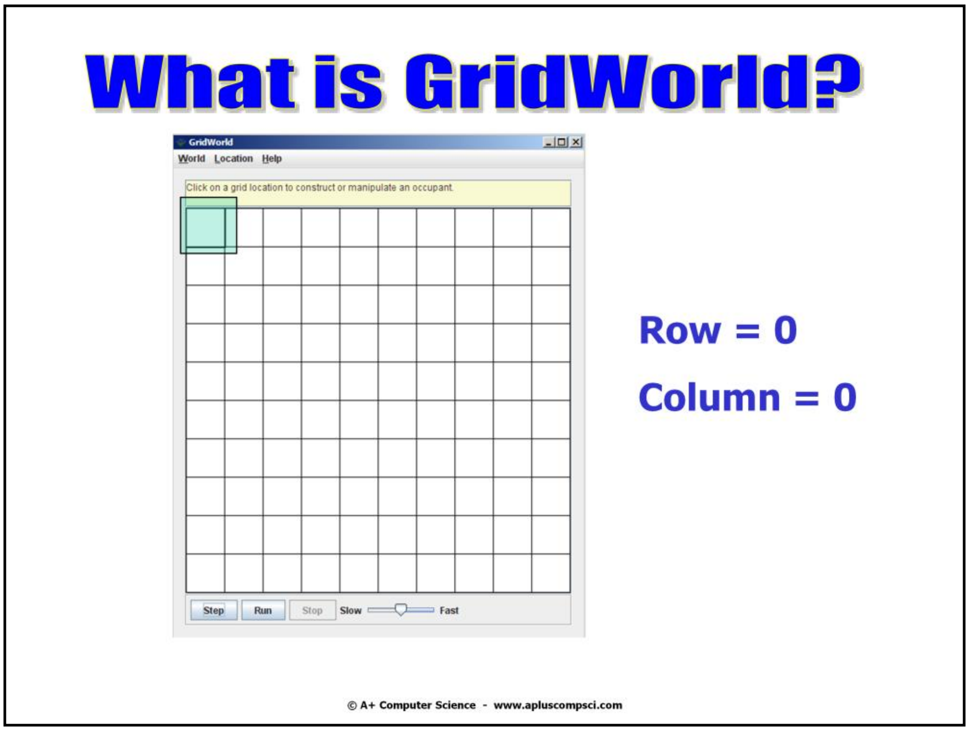
GridWorld



What is GridWorld?

GridWorld
World Location Help
Click on a grid location to construct or manipulate an occupant.
Step Run Stop Slow Fast

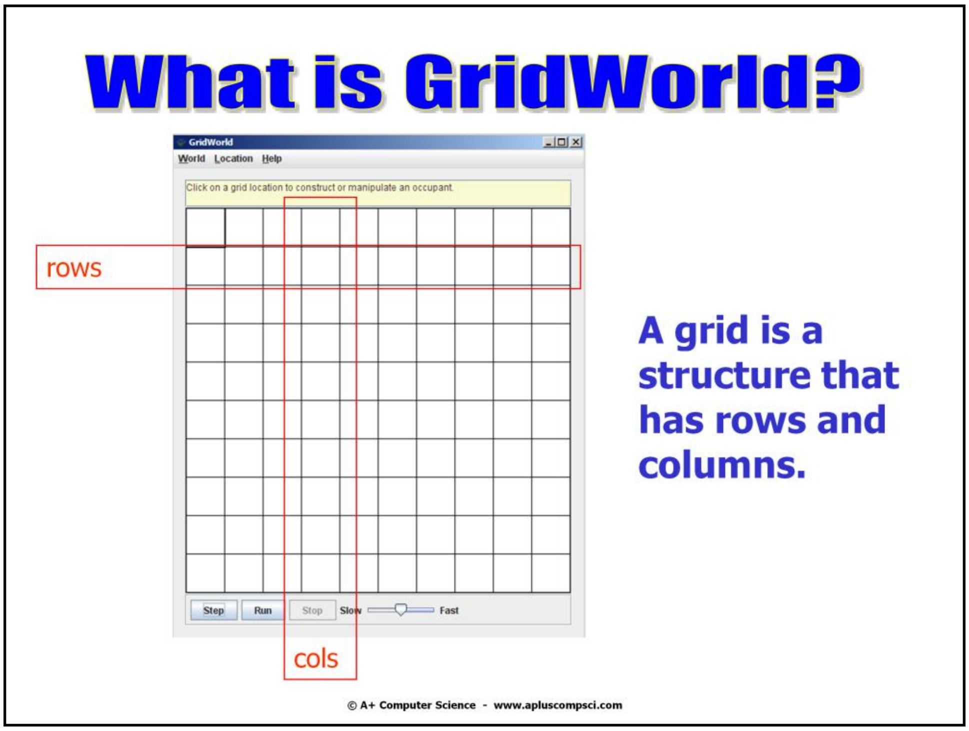
Row = 0

Column = 0

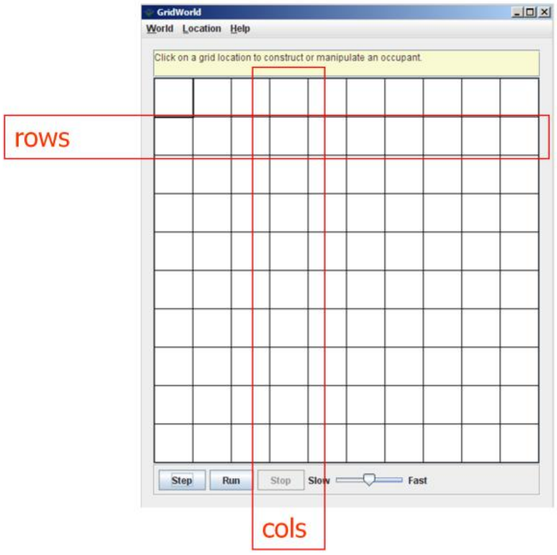
A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.



What is GridWorld?


GridWorld
World Location Help
Click on a grid location to construct or manipulate an occupant.
rows
cols
Step
Run
Stop
Slow
Fast

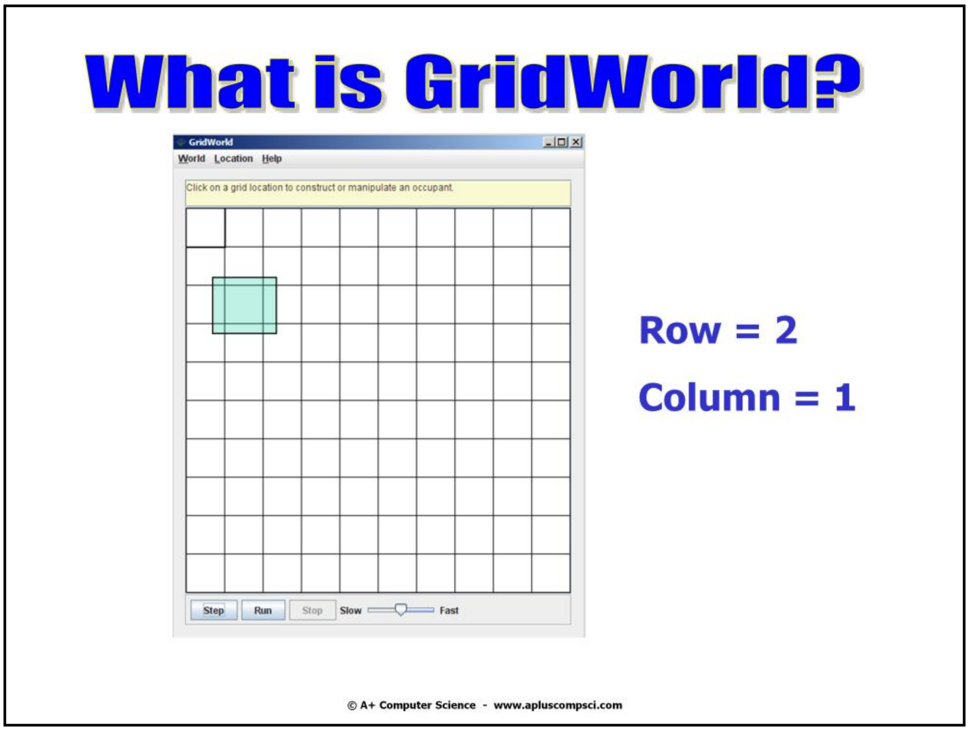
A grid is a structure that has rows and columns.

© A+ Computer Science - www.apluscompsci.com

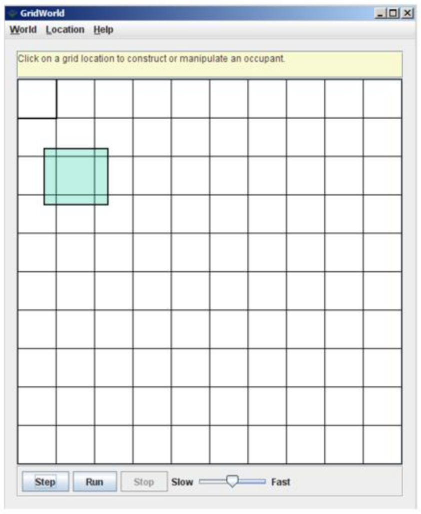
A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.



What is GridWorld?



Click on a grid location to construct or manipulate an occupant.

Row = 2

Column = 1

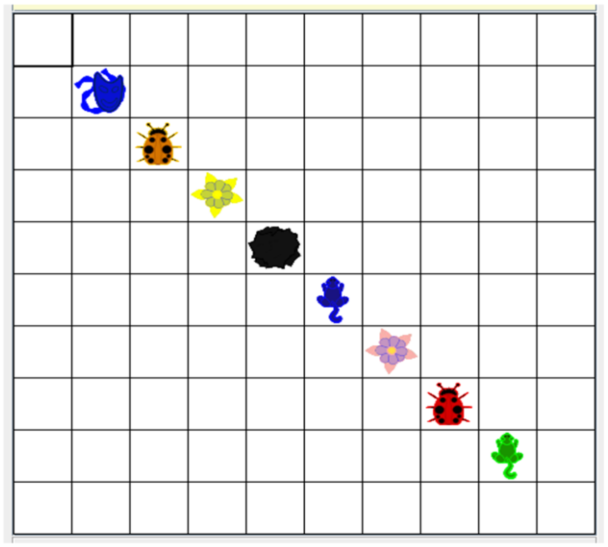
© A+ Computer Science - www.apluscompsci.com

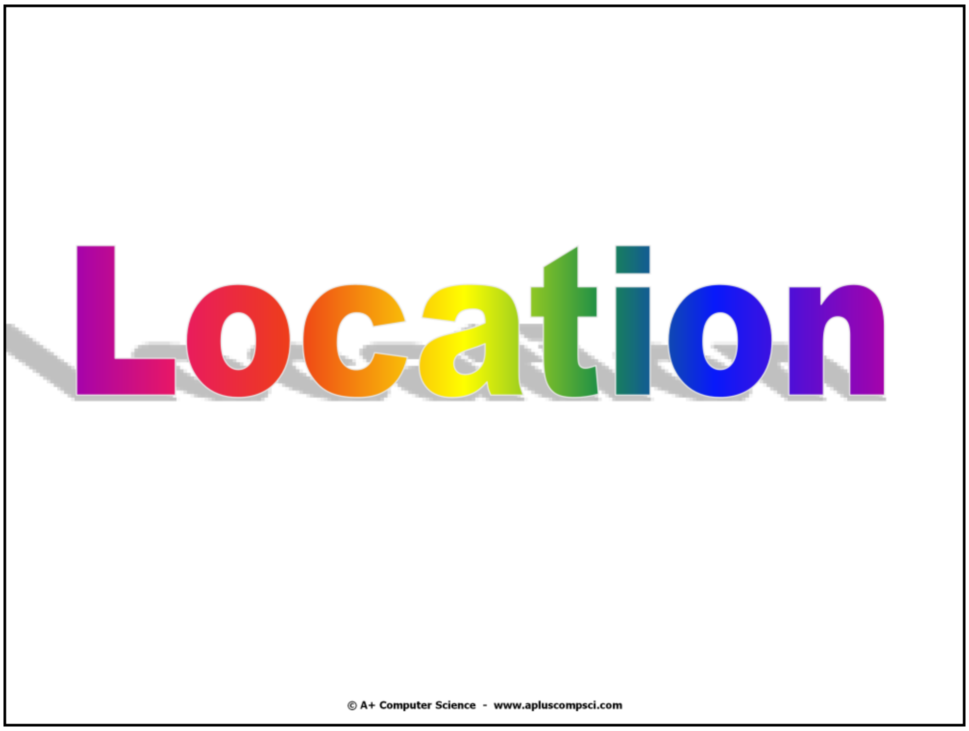
A grid is a structure that has rows and columns.

A spreadsheet is a grid.

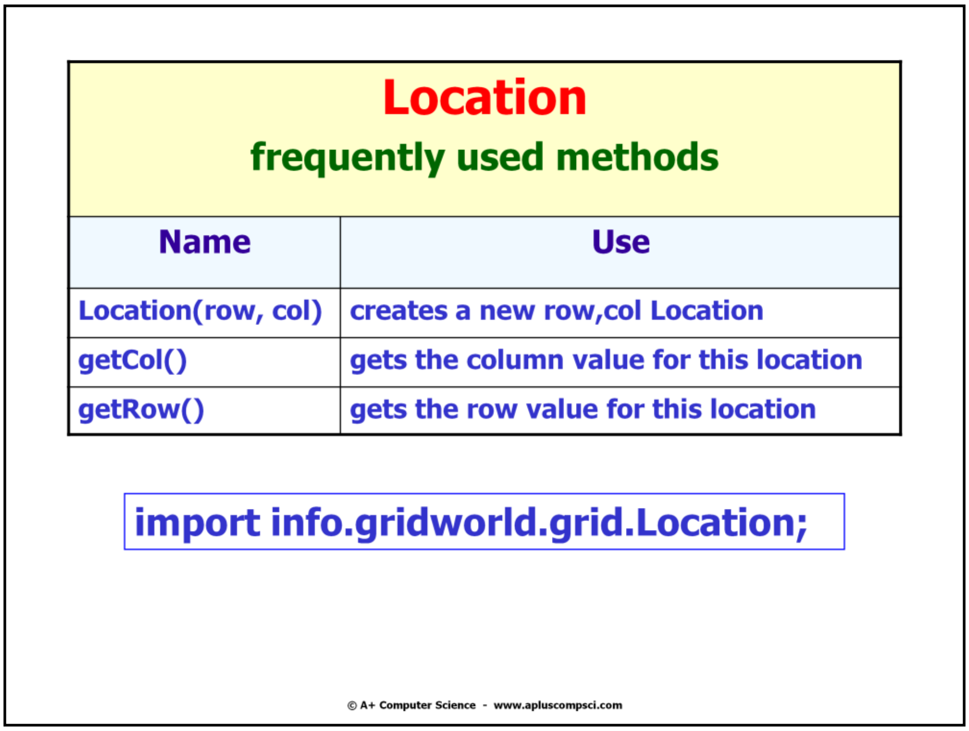
A checker board is a grid.

What is GridWorld?





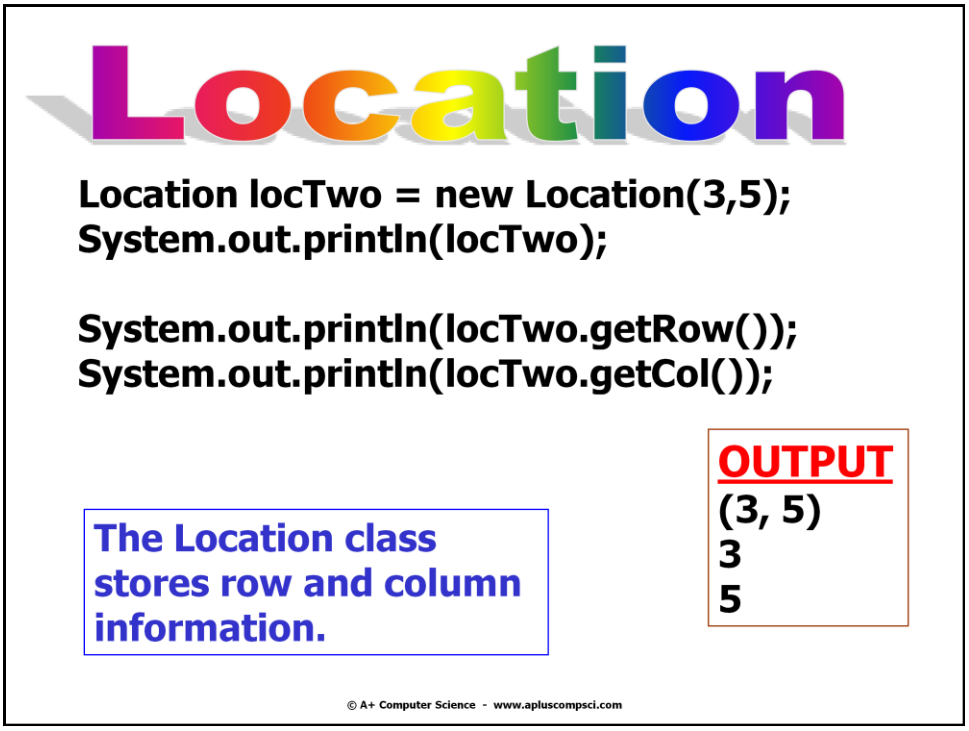
Location



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

```
Location locTwo = new Location(3,5);
System.out.println(locTwo);

System.out.println(locTwo.getRow());
System.out.println(locTwo.getCol());
```

OUTPUT

```
(3, 5)
3
5
```

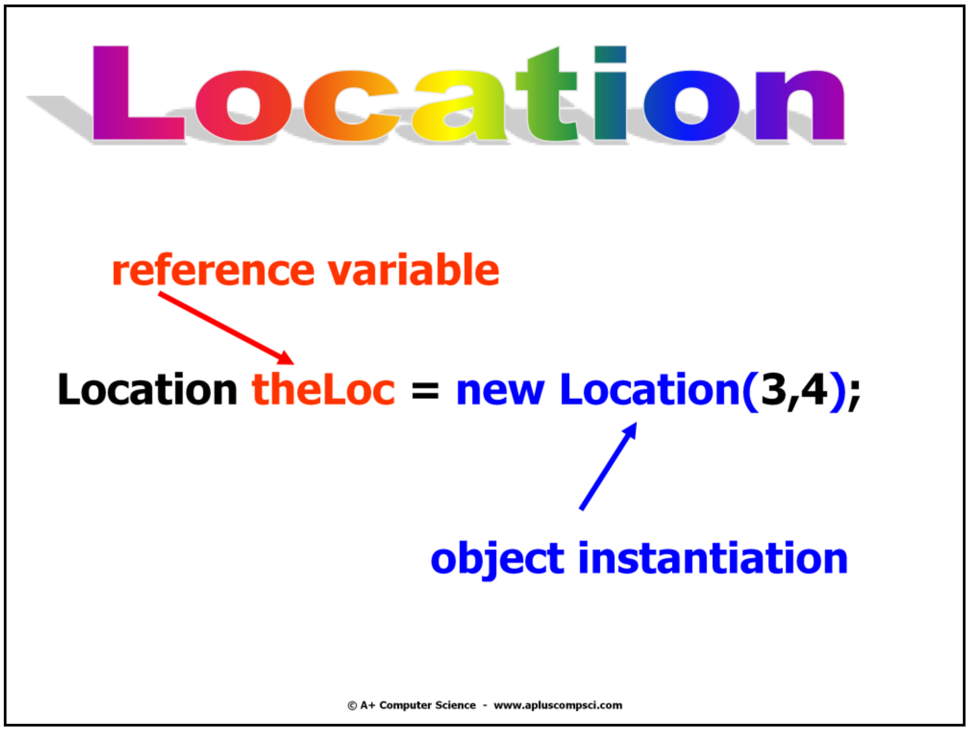
The Location class stores row and column information.

© A+ Computer Science - www.apluscompsci.com

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

reference variable

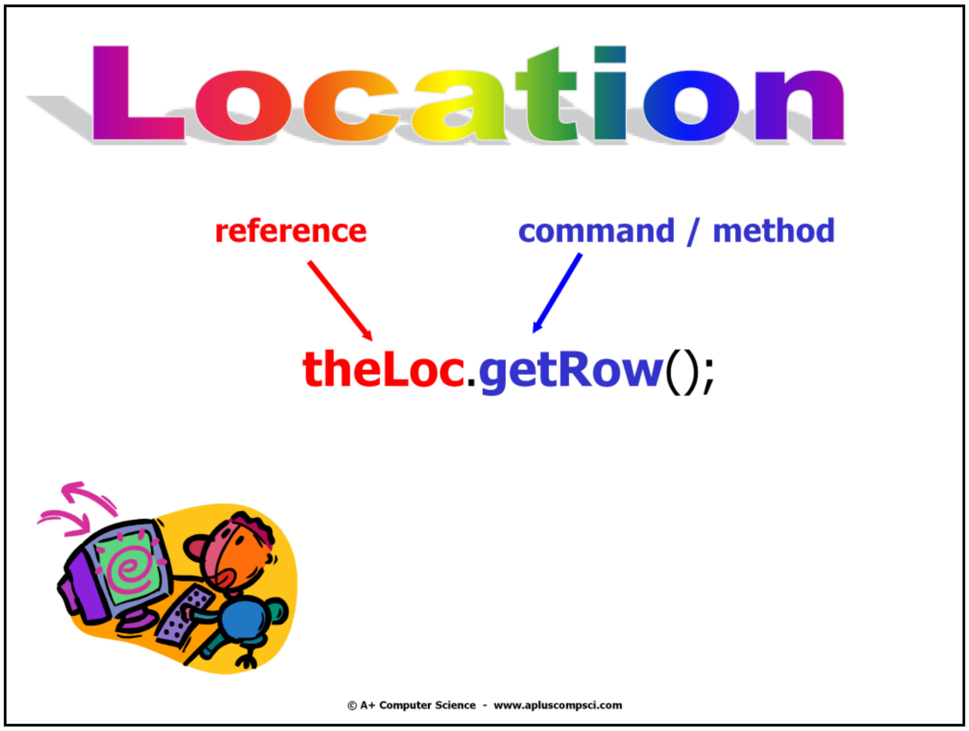
```
Location theLoc = new Location(3,4);
```

object instantiation

`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.

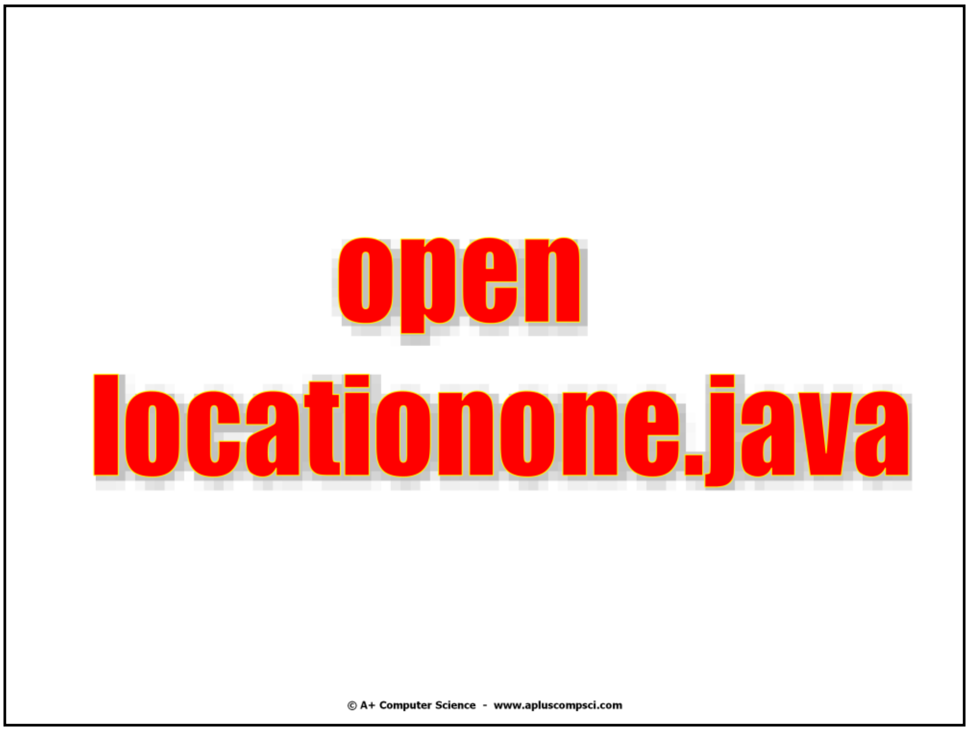
`theLoc` is a reference that will store the location/memory address of newly created `Location` object.



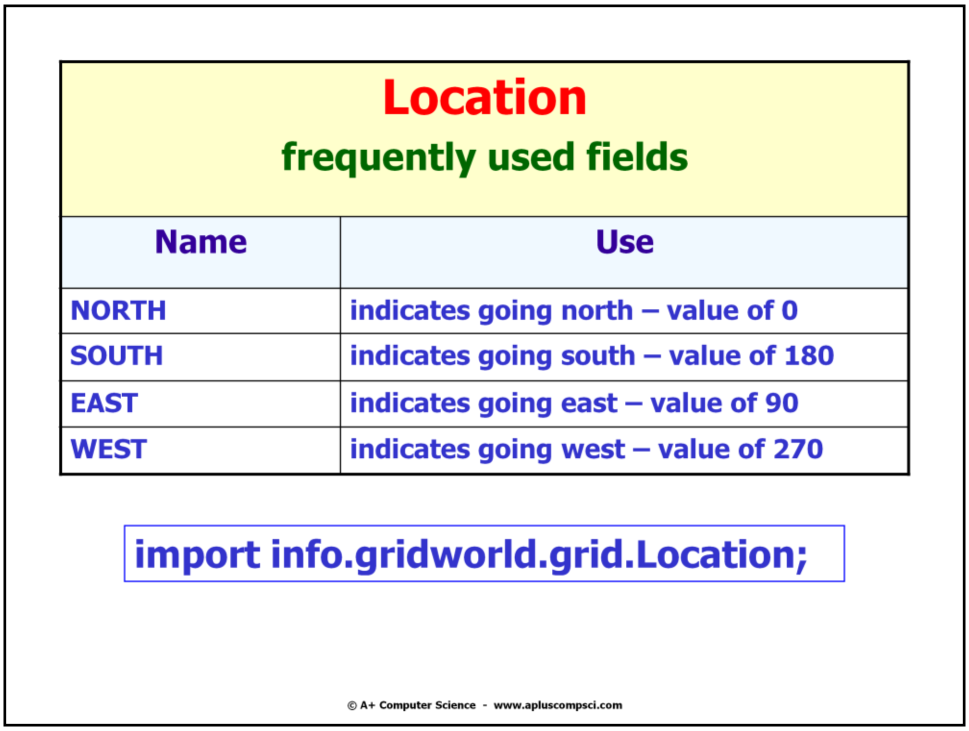
Location

reference   command / method

```
theLoc.getRow();
```



open locationone.java



Location
frequently used fields

Name	Use
NORTH	indicates going north – value of 0
SOUTH	indicates going south – value of 180
EAST	indicates going east – value of 90
WEST	indicates going west – value of 270

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

The location class contains the following 8 directions :

Location.NORTH

Location.SOUTH

Location.EAST

Location.WEST

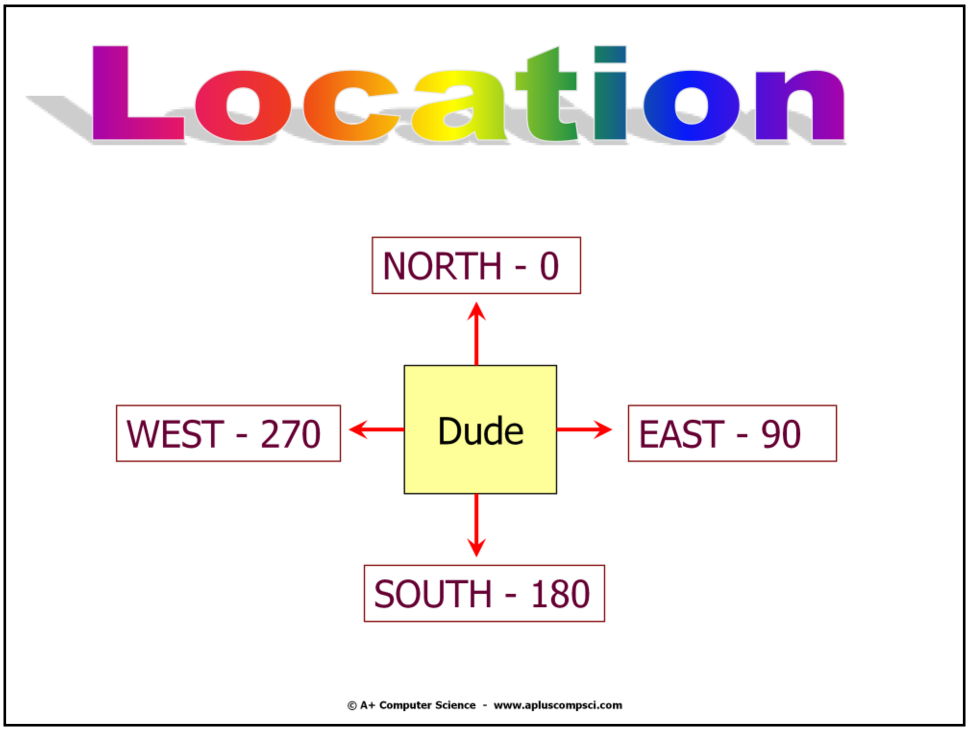
Location.NORTHWEST

Location.SOUTHWEST

Location.NORTHEAST

Location.NORTHWEST

All of the direction fields in the location class are final integers.



The location class contains the following 8 directions :

Location.NORTH

Location.SOUTH

Location.EAST

Location.WEST

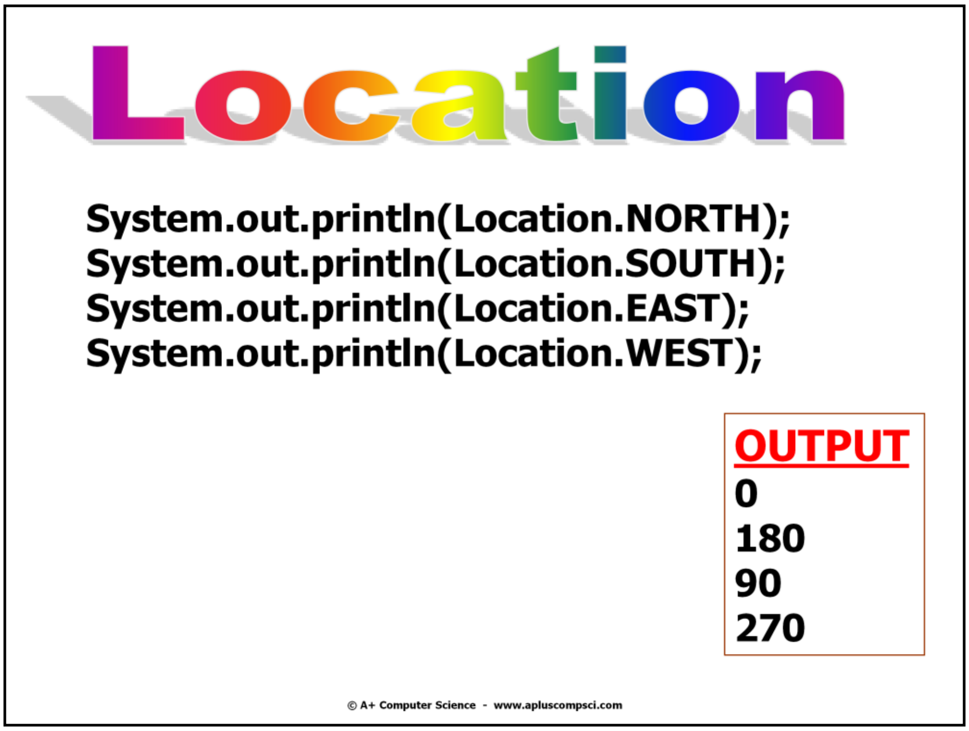
Location.NORTHWEST

Location.SOUTHWEST

Location.NORTHEAST

Location.NORTHWEST

All of the direction fields in the location class are final integers.



Location

```
System.out.println(Location.NORTH);
System.out.println(Location.SOUTH);
System.out.println(Location.EAST);
System.out.println(Location.WEST);
```

OUTPUT

```
0
180
90
270
```

© A+ Computer Science - www.apluscompsci.com

The location class contains the following 8 directions :

Location.NORTH

Location.SOUTH

Location.EAST

Location.WEST

Location.NORTHWEST

Location.SOUTHWEST

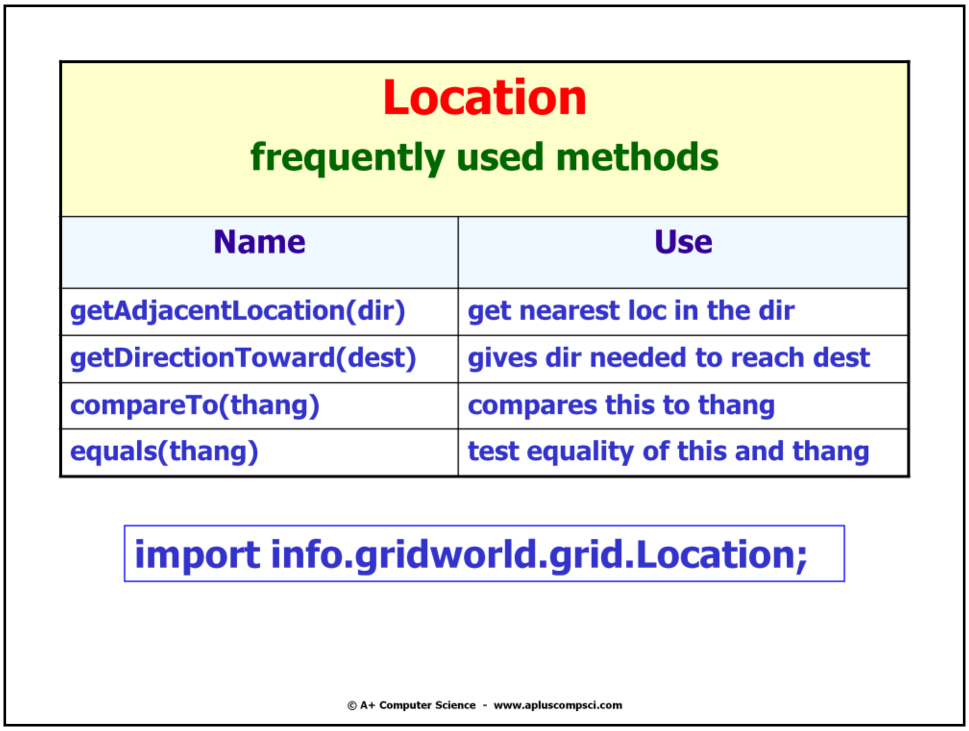
Location.NORTHEAST

Location.NORTHWEST

All of the direction fields in the location class are final integers.

open locationtwo.java

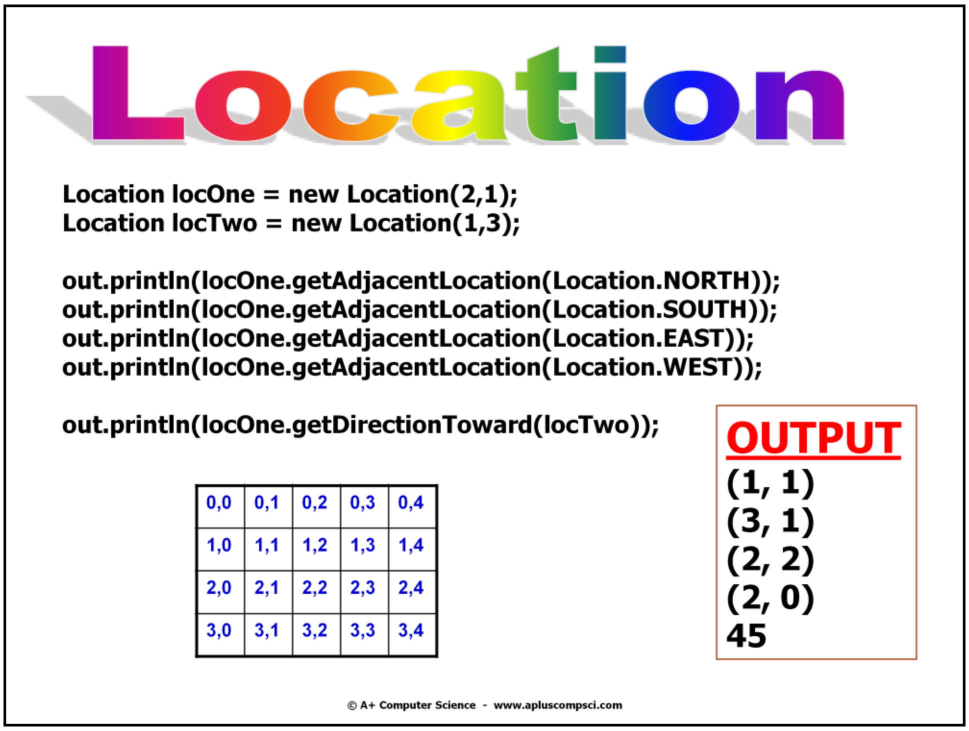
© A+ Computer Science - www.apluscompsci.com



Location
frequently used methods

Name	Use
getAdjacentLocation(dir)	get nearest loc in the dir
getDirectionToward(dest)	gives dir needed to reach dest
compareTo(thang)	compares this to thang
equals(thang)	test equality of this and thang

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



Location

```
Location locOne = new Location(2,1);
Location locTwo = new Location(1,3);

out.println(locOne.getAdjacentLocation(Location.NORTH));
out.println(locOne.getAdjacentLocation(Location.SOUTH));
out.println(locOne.getAdjacentLocation(Location.EAST));
out.println(locOne.getAdjacentLocation(Location.WEST));

out.println(locOne.getDirectionToward(locTwo));
```

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4

OUTPUT

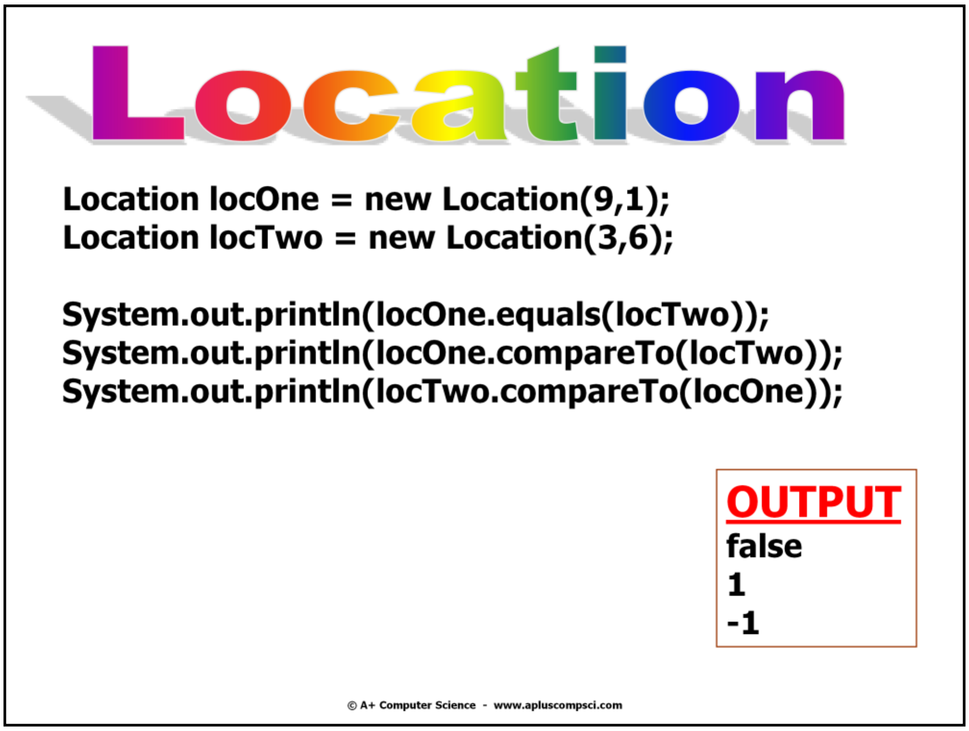
```
(1, 1)
(3, 1)
(2, 2)
(2, 0)
45
```

© A+ Computer Science - www.apluscompsci.com

getAdjacentLocation will return a nearby location based on a provided direction.

getDirectionToward will look at two locations and return the direction required to go from a to b.

open locationthree.java



Location

```
Location locOne = new Location(9,1);
Location locTwo = new Location(3,6);

System.out.println(locOne.equals(locTwo));
System.out.println(locOne.compareTo(locTwo));
System.out.println(locTwo.compareTo(locOne));
```

OUTPUT

```
false
1
-1
```

© A+ Computer Science - www.apluscompsci.com

The equals method compares two locations to see if both have the same row and column values.

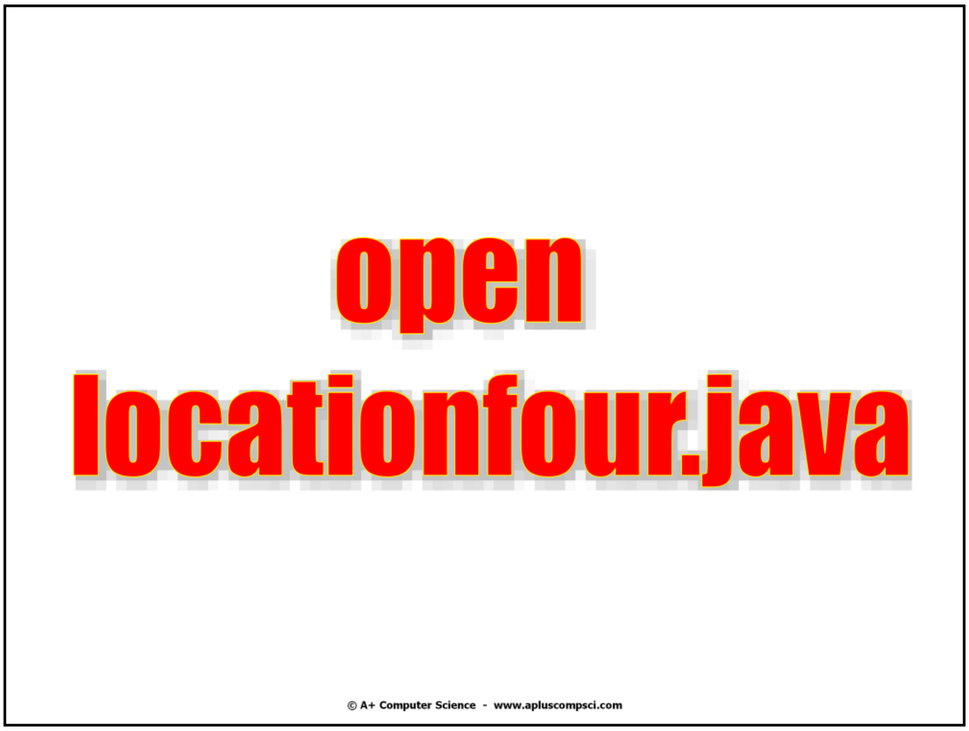
The compareTo method compares two locations for equality, greater than, and less than.

If A>B, compareTo returns a value >0.

If A<B, compareTo returns a value <0;

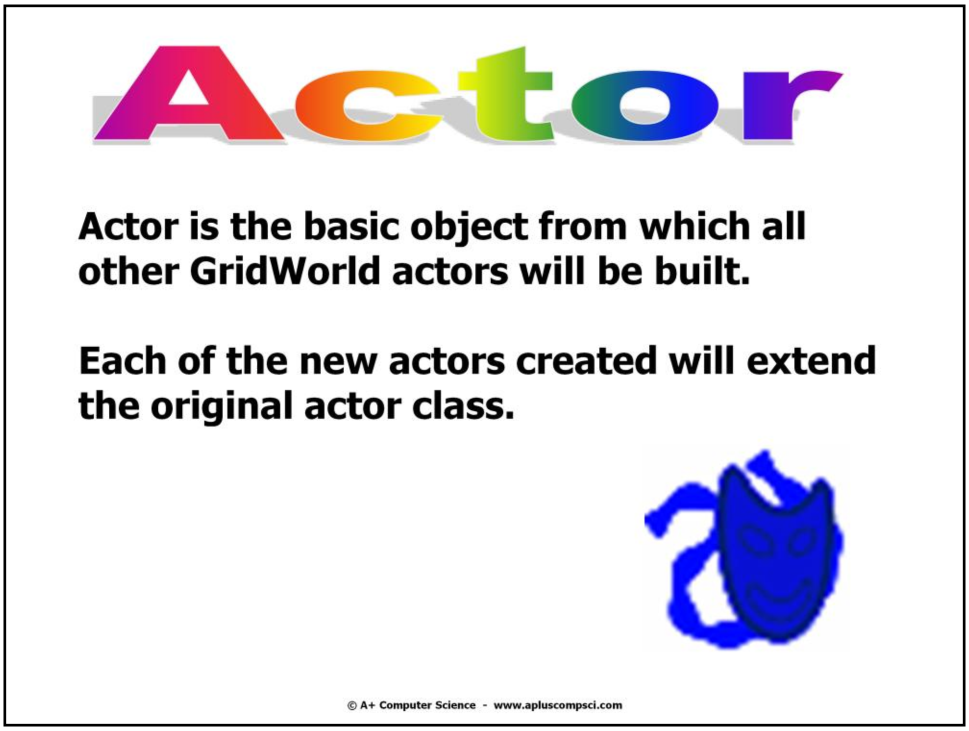
If A==B, compareTo returns 0.

When comparing locations, compareTo first compares the row value. If the row values are the same, the column value is compared.



open locationfour.java

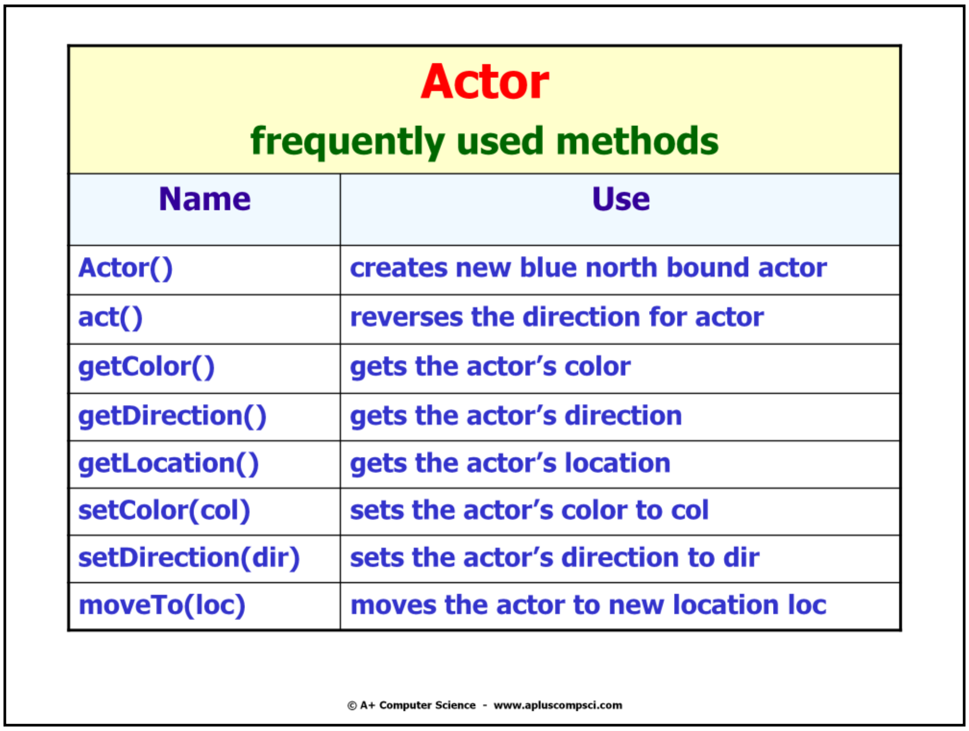
Actor



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.

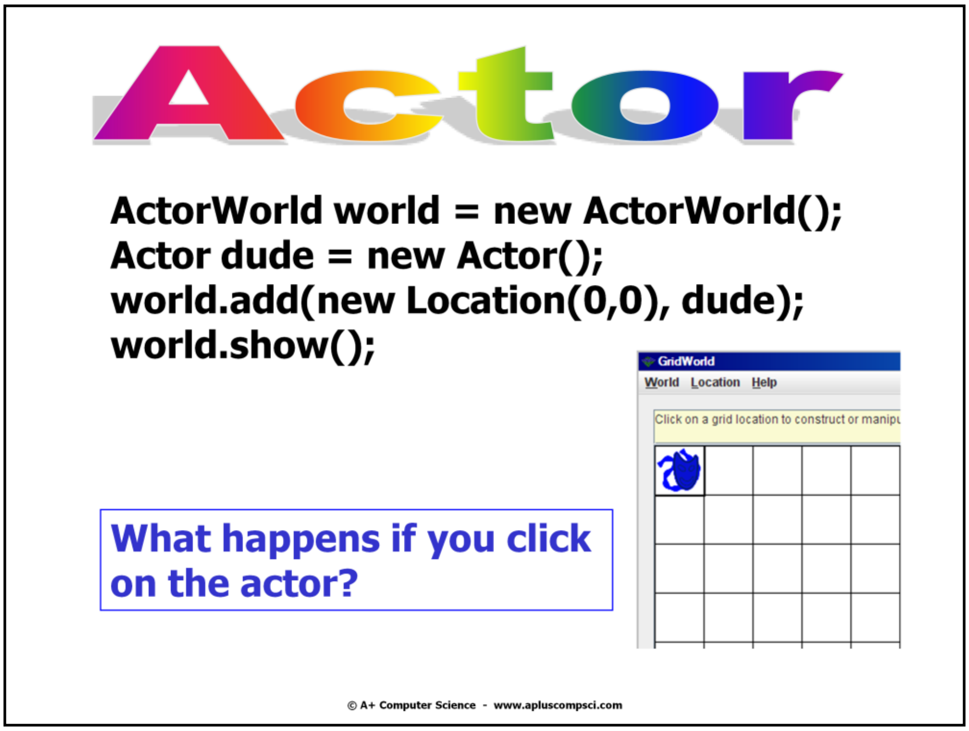


Actor

frequently used methods

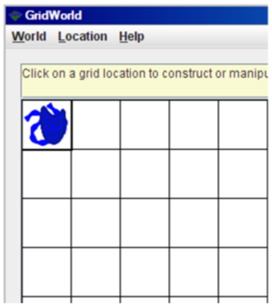
Name	Use
Actor()	creates new blue north bound actor
act()	reverses the direction for actor
getColor()	gets the actor's color
getDirection()	gets the actor's direction
getLocation()	gets the actor's location
setColor(col)	sets the actor's color to col
setDirection(dir)	sets the actor's direction to dir
moveTo(loc)	moves the actor to new location loc

Notice that actor has only one constructor and that that constructor takes no parameters.



Actor

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



What happens if you click on the actor?

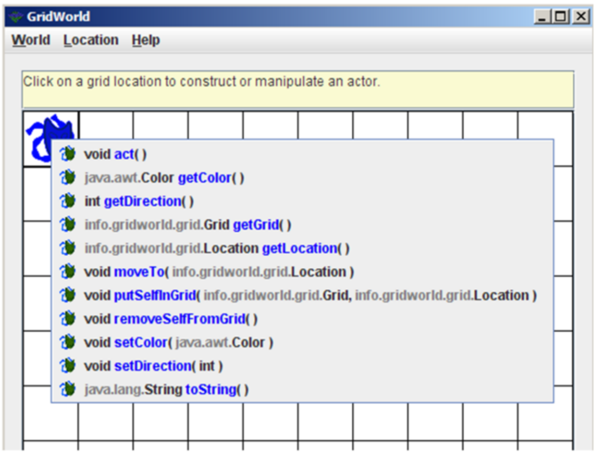
The add method of class world receives a location parameter and an actor.

The actor reference is stored in the grid at the specified location.

The default color of an actor is BLUE.

Actor

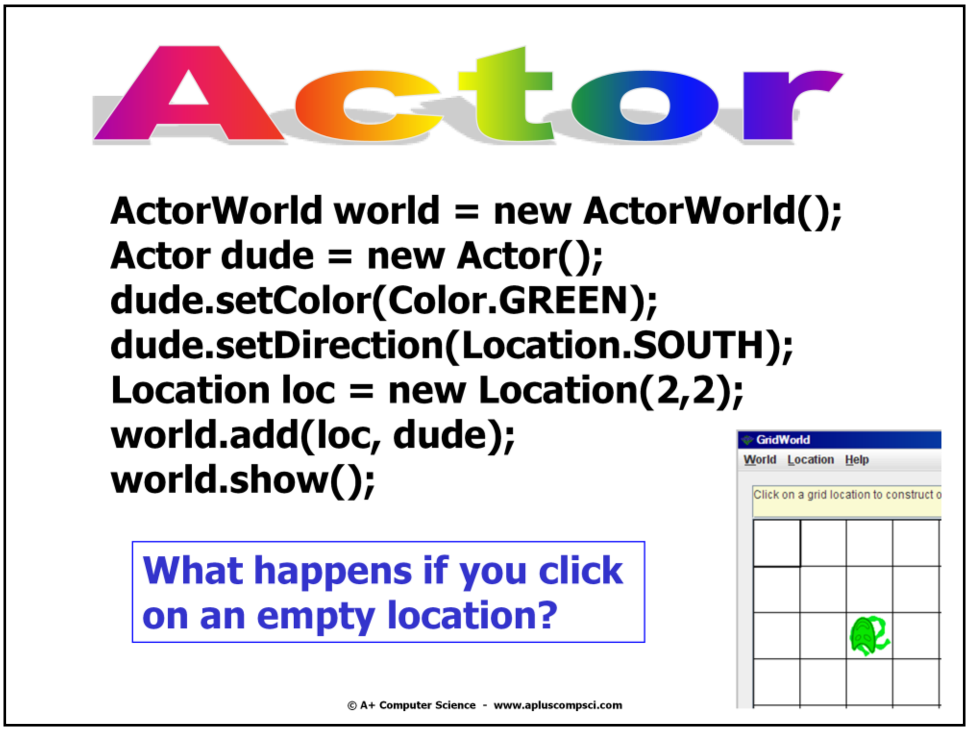
When you click on an actor, a list of methods is shown.



Click on a grid location to construct or manipulate an actor.

```
void act( )
java.awt.Color getColor( )
int getDirection( )
info.gridworld.grid.Grid getGrid( )
info.gridworld.grid.Location getLocation( )
void moveTo( info.gridworld.grid.Location )
void putSelfInGrid( info.gridworld.grid.Grid, info.gridworld.grid.Location )
void removeSelfFromGrid( )
void setColor( java.awt.Color )
void setDirection( int )
java.lang.String toString( )
```

open
actorone.java



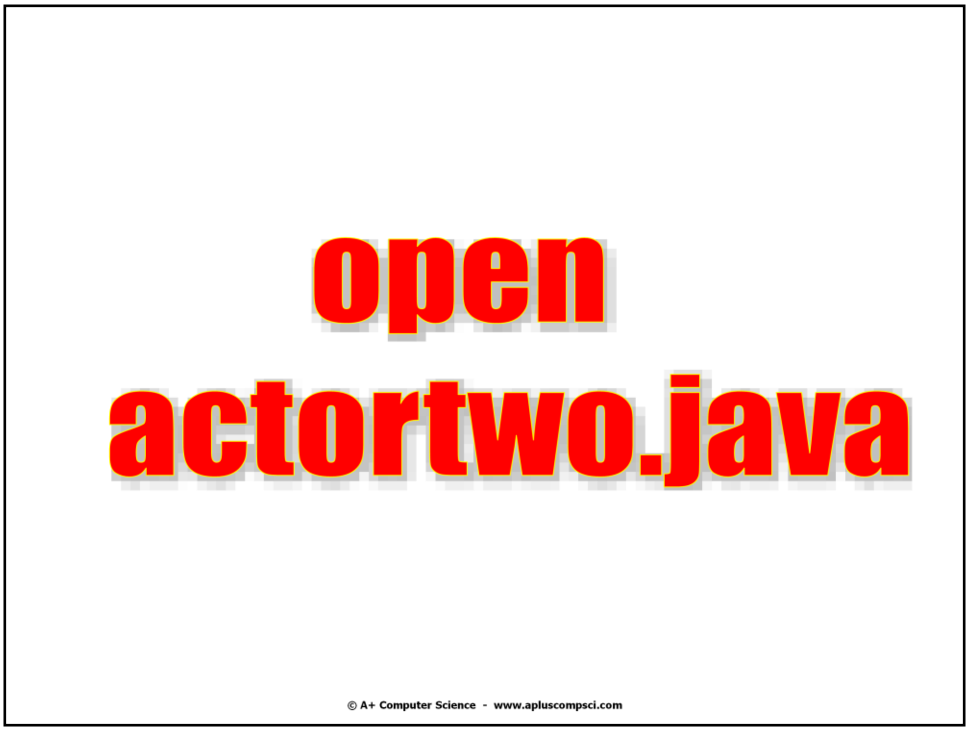
Actor

```
ActorWorld world = new ActorWorld();
Actor dude = new Actor();
dude.setColor(Color.GREEN);
dude.setDirection(Location.SOUTH);
Location loc = new Location(2,2);
world.add(loc, dude);
world.show();
```

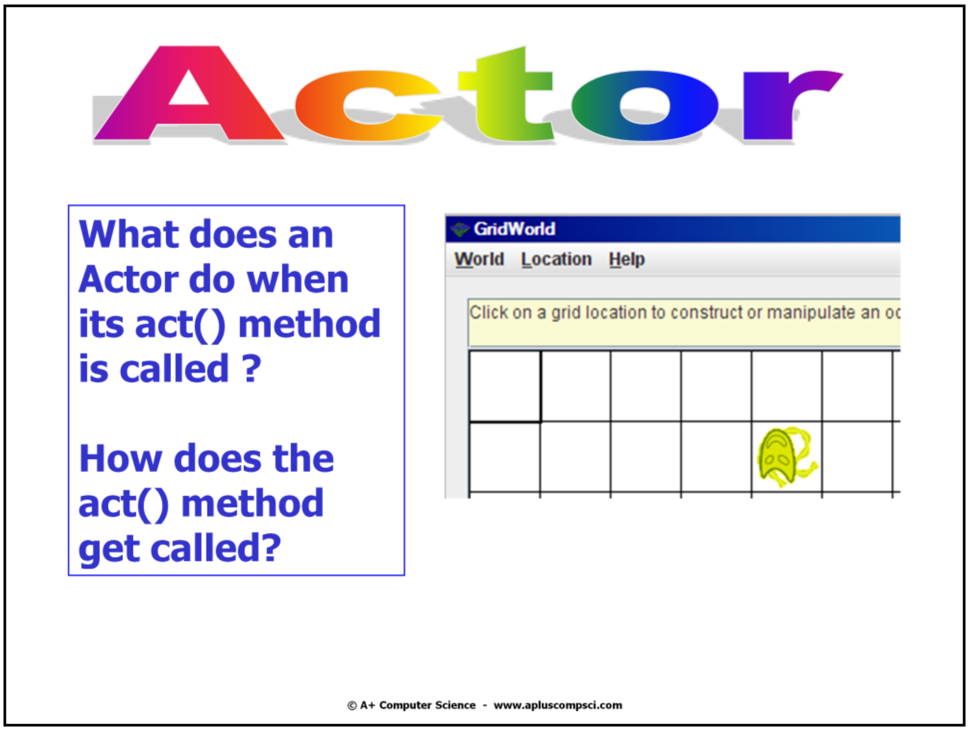
What happens if you click on an empty location?

GridWorld
World Location Help
Click on a grid location to construct o

If you want an actor have a color other than BLUE, simply call the setColor method and provide the color of your choice.



open actortwo.java



Actor

What does an Actor do when its act() method is called ?

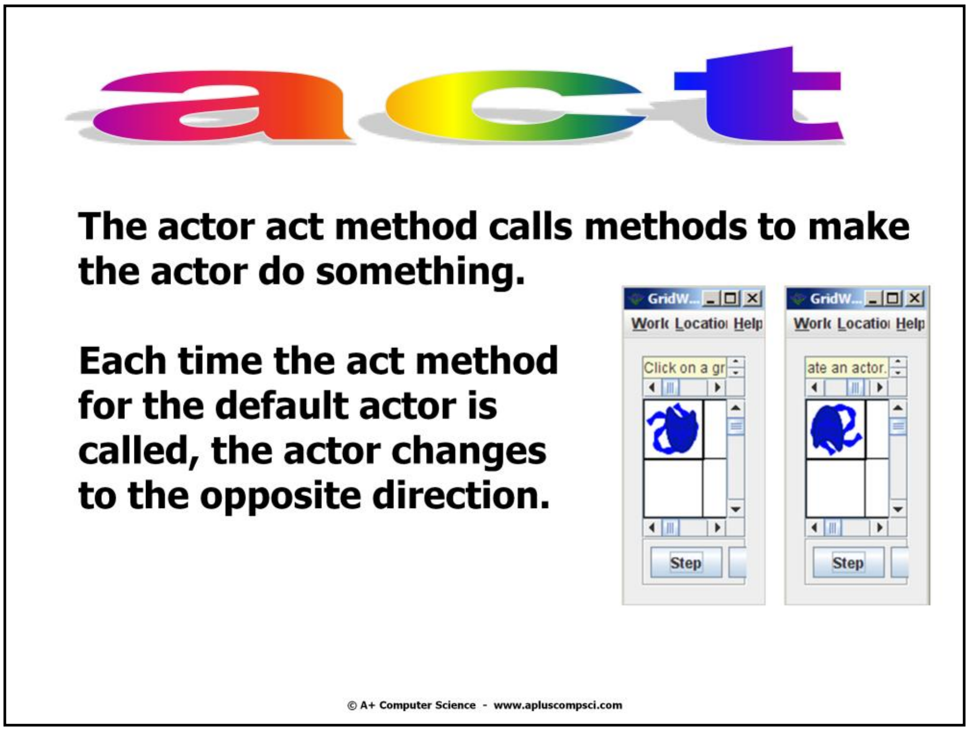
How does the act() method get called?

© A+ Computer Science - www.apluscompsci.com

An actor does not move from cell to cell, but it does flip from side to side within its cell.

An actor's act method is called when the step or run button is called.

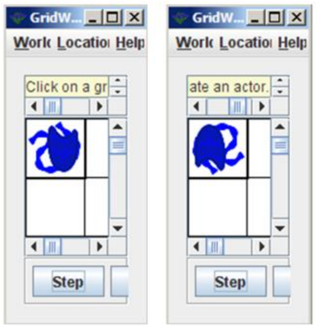
The act method contains the code that determines the behavior for each type of actor.



act

The actor act method calls methods to make the actor do something.

Each time the act method for the default actor is called, the actor changes to the opposite direction.



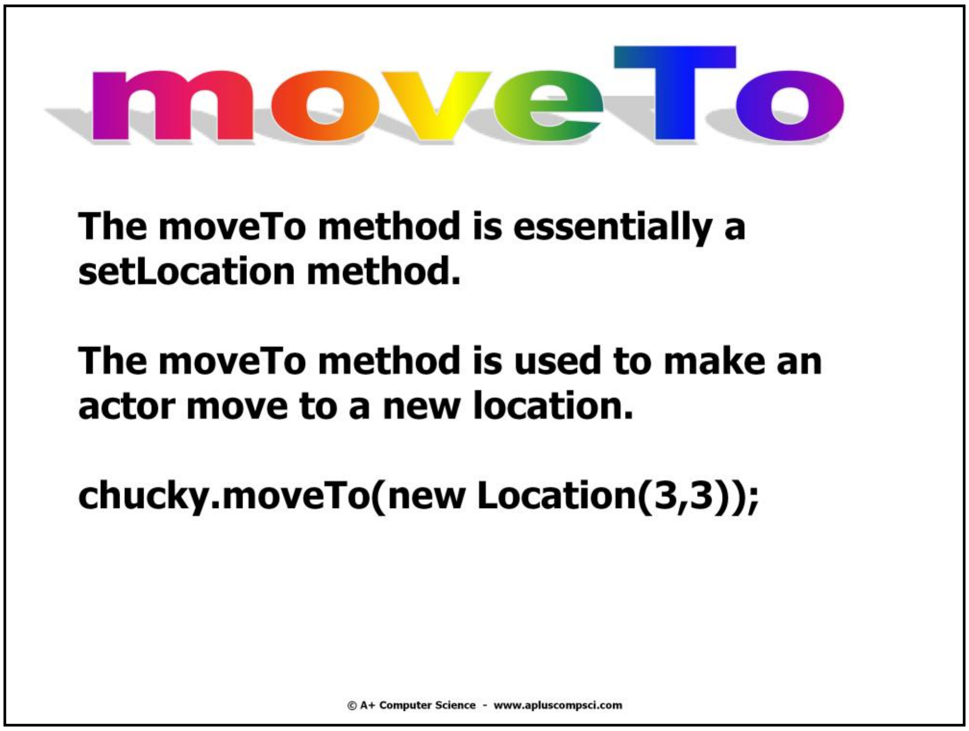
© A+ Computer Science - www.apluscompsci.com

An actor does not move from cell to cell, but it does flip from side to side within its cell.

An actor's act method is called when the step or run button is called.

The act method contains the code that determines the behavior for each type of actor.

The act method will be overridden when new types of actors are created.


moveTo

The moveTo method is essentially a setLocation method.

The moveTo method is used to make an actor move to a new location.

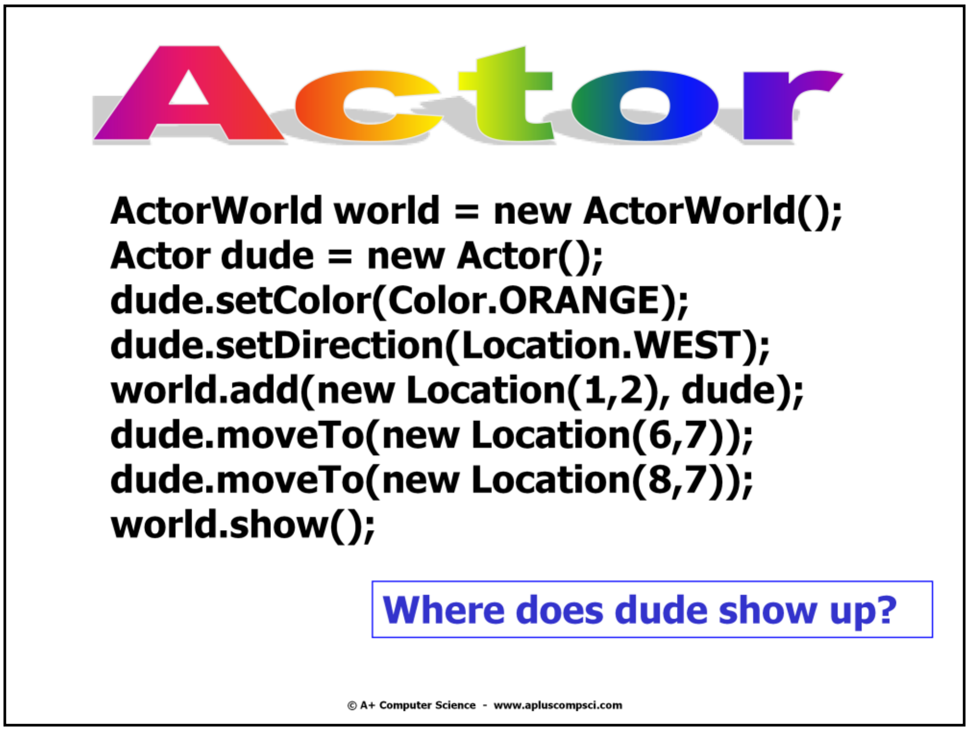
```
chucky.moveTo(new Location(3,3));
```

© A+ Computer Science - www.apluscompsci.com

The actor moveTo method can be used to move an actor to another location.

The actor moveTo method functions like a setLocation method.

Actor does not have a setLocation method.



Actor

```
ActorWorld world = new ActorWorld();
Actor dude = new Actor();
dude.setColor(Color.ORANGE);
dude.setDirection(Location.WEST);
world.add(new Location(1,2), dude);
dude.moveTo(new Location(6,7));
dude.moveTo(new Location(8,7));
world.show();
```

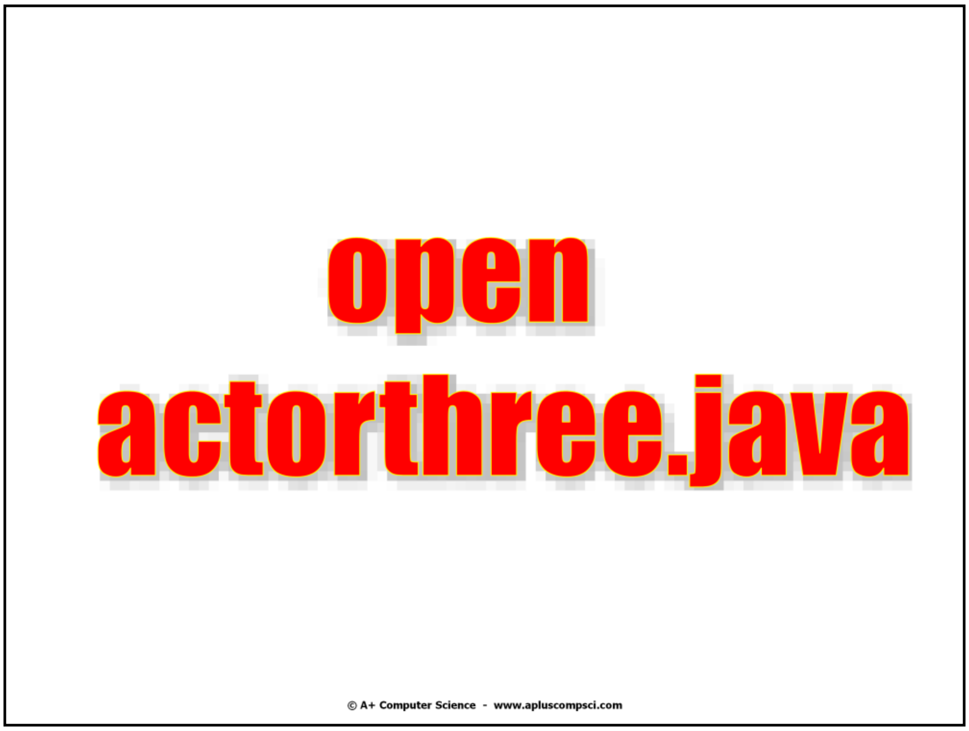
Where does dude show up?

© A+ Computer Science - www.apluscompsci.com

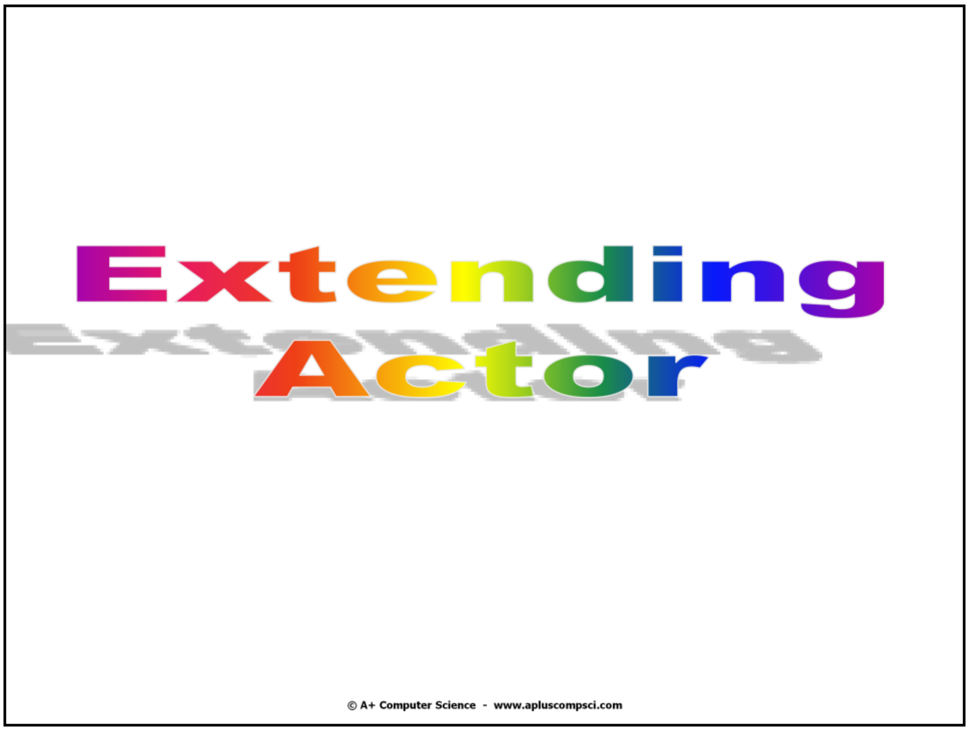
You can place an actor in the grid at a specified location and then that actor's location may change as the program runs.

You can also place the actor in the grid in the main and then for testing and understanding, change the location with code statements to see the effect.

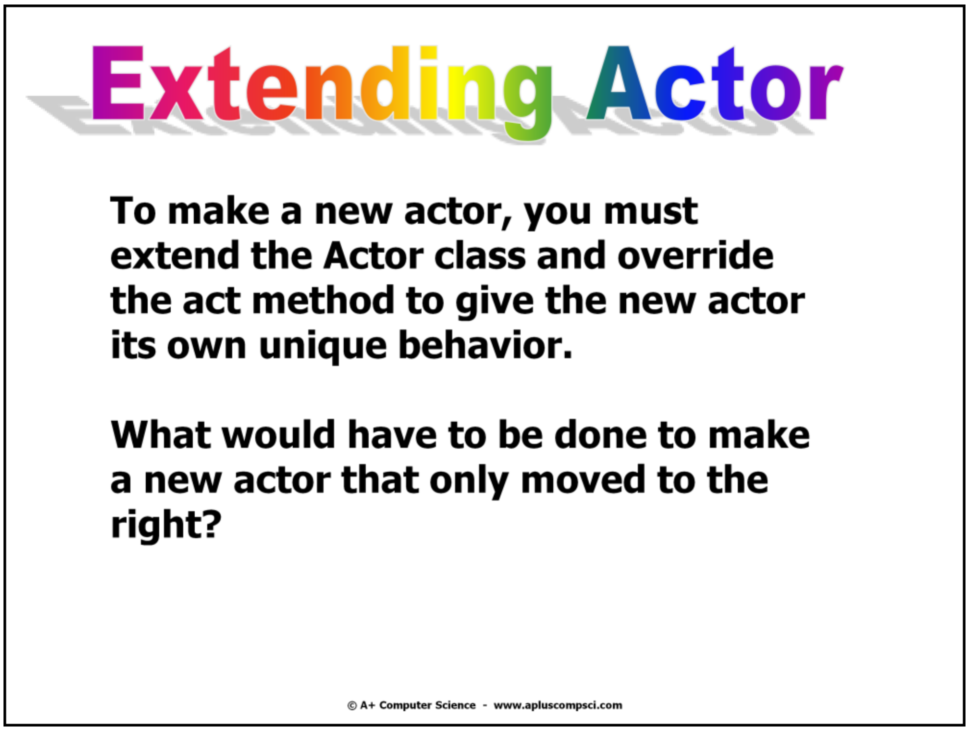
For the code above, the actor will appear at position 8,7. The actor had an initial location of 1,2 that was changed to 6,7 and then finally set as 8,7.



open actorthree.java



Extending Actor



Extending Actor

To make a new actor, you must extend the Actor class and override the act method to give the new actor its own unique behavior.

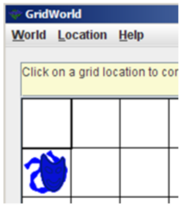
What would have to be done to make a new actor that only moved to the right?

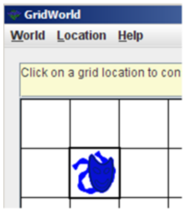
When extending actor, the act method will most always be overridden to create some new type of actor with new behaviors.

Extending Actor

```
public class SideWaysActor extends Actor
{
 public void act()
 {
   //move to the right

 }
}
```



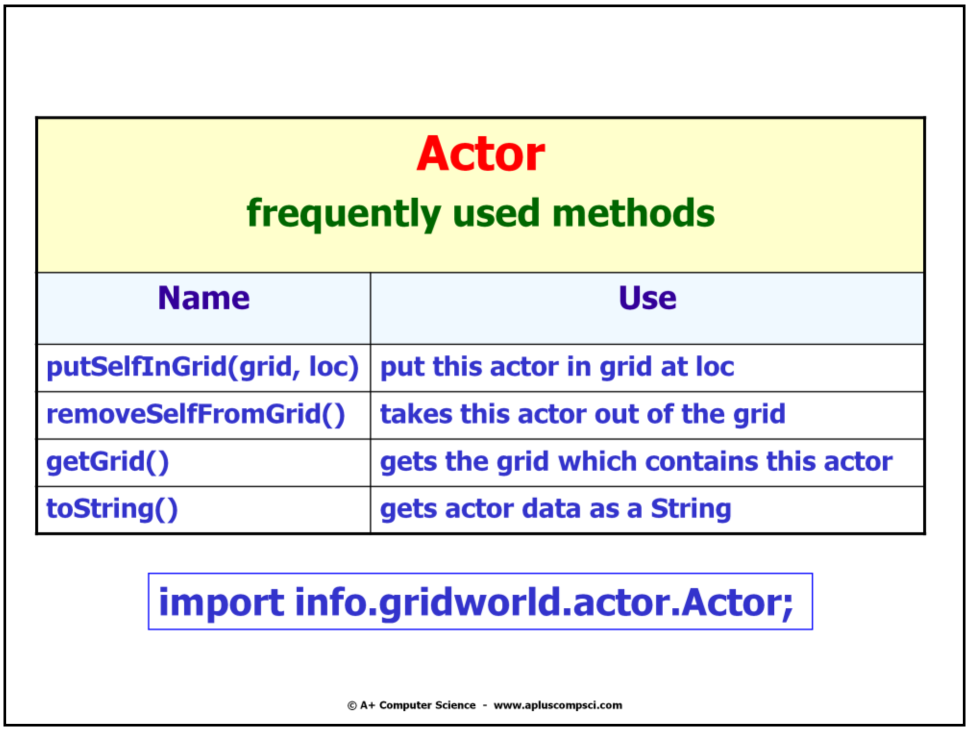


© A+ Computer Science - www.apluscompsci.com

When extending actor, the act method will most always be overridden to create some new type of actor with new behaviors.

open
sidewaysactor.java
sidewaysactorrunner.java

Start work on Actor Labs

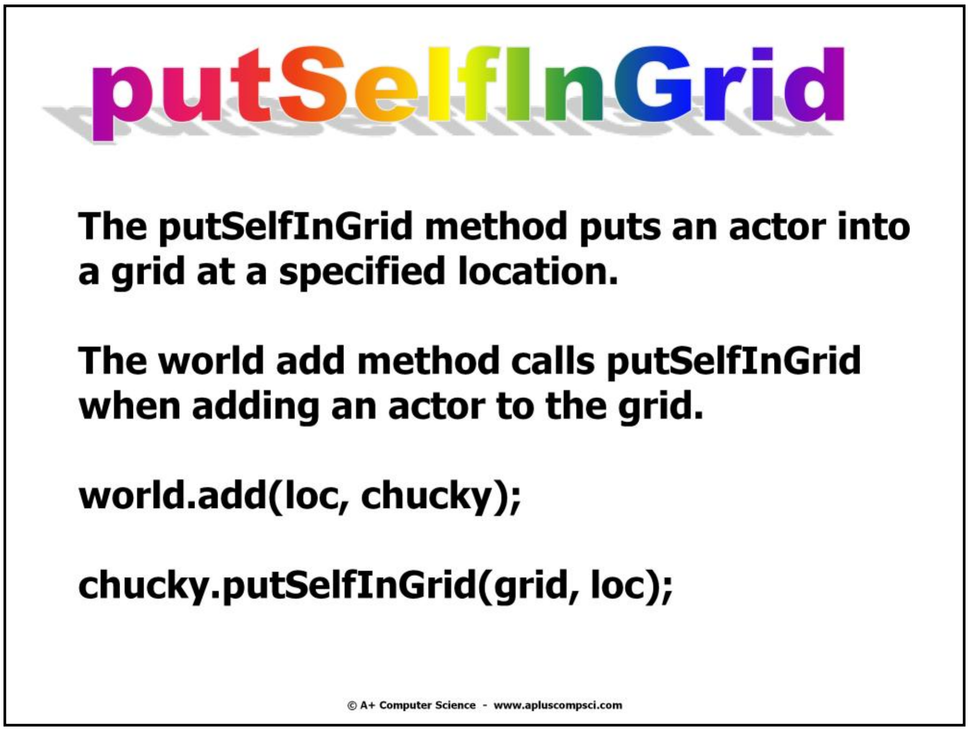


Actor

frequently used methods

Name	Use
putSelfInGrid(grid, loc)	put this actor in grid at loc
removeSelfFromGrid()	takes this actor out of the grid
getGrid()	gets the grid which contains this actor
toString()	gets actor data as a String

```
import info.gridworld.actor.Actor;
```



putSelfInGrid

The putSelfInGrid method puts an actor into a grid at a specified location.

The world add method calls putSelfInGrid when adding an actor to the grid.

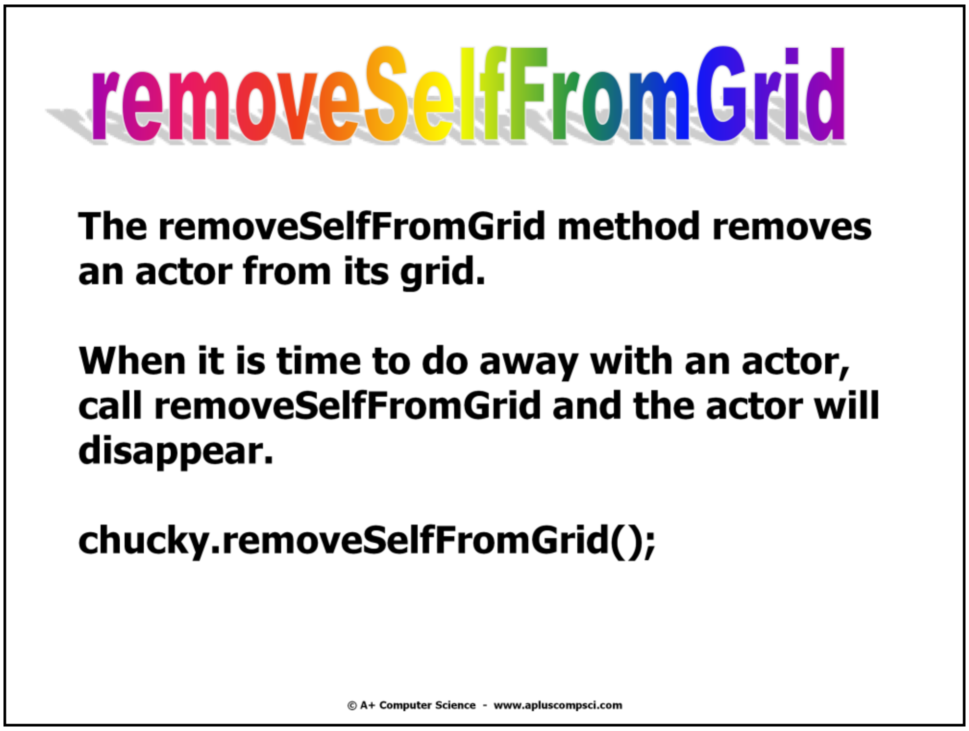
```
world.add(loc, chucky);

chucky.putSelfInGrid(grid, loc);
```

© A+ Computer Science - www.apluscompsci.com

When placing an actor in the grid, the world add method or actor putSelfInGrid method ***must*** be called.

The world add method calls the actor putSelfInGrid method automatically.


removeSelfFromGrid

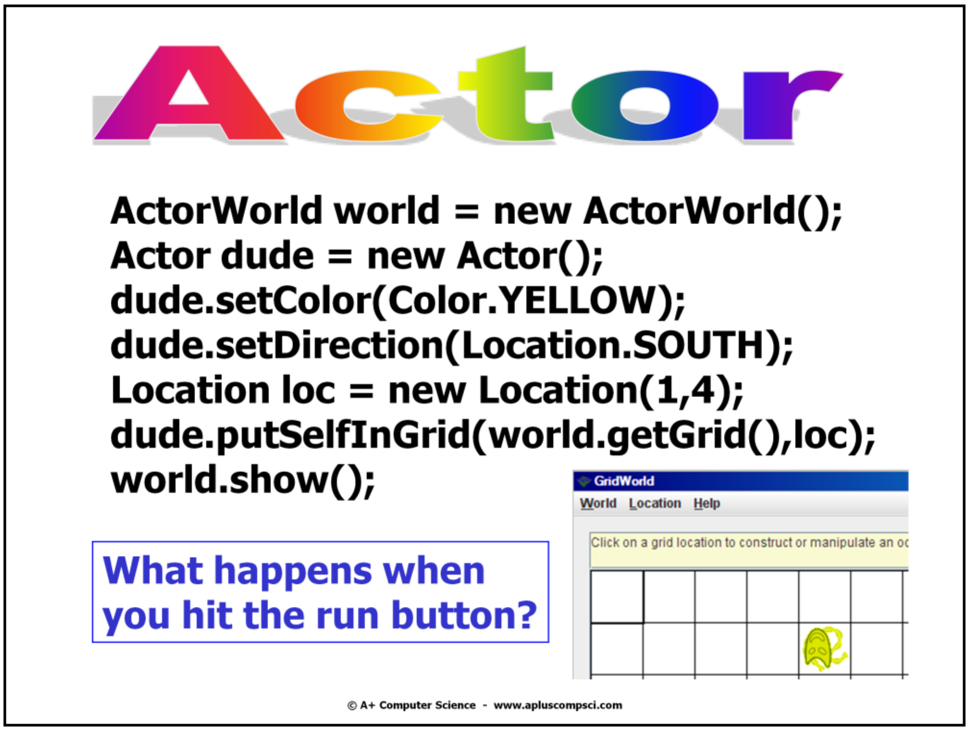
The removeSelfFromGrid method removes an actor from its grid.

When it is time to do away with an actor, call removeSelfFromGrid and the actor will disappear.

```
chucky.removeSelfFromGrid();
```

© A+ Computer Science - www.apluscompsci.com

The removeSelfFromGrid is useful to remove an actor from a grid.



Actor

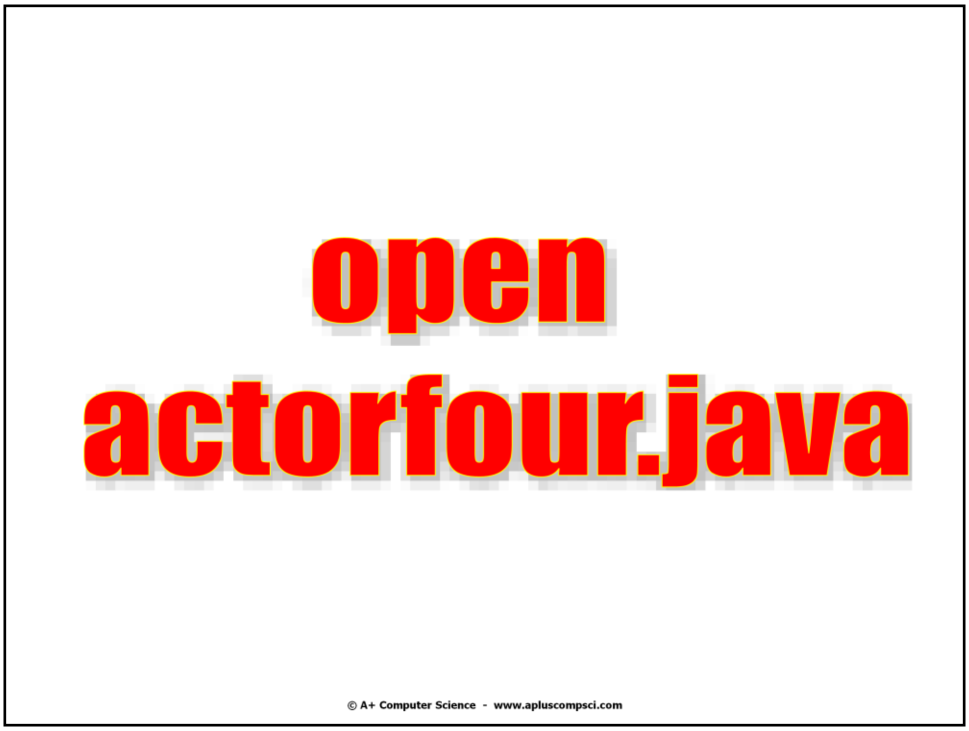
```
ActorWorld world = new ActorWorld();
Actor dude = new Actor();
dude.setColor(Color.YELLOW);
dude.setDirection(Location.SOUTH);
Location loc = new Location(1,4);
dude.putSelfInGrid(world.getGrid(),loc);
world.show();
```

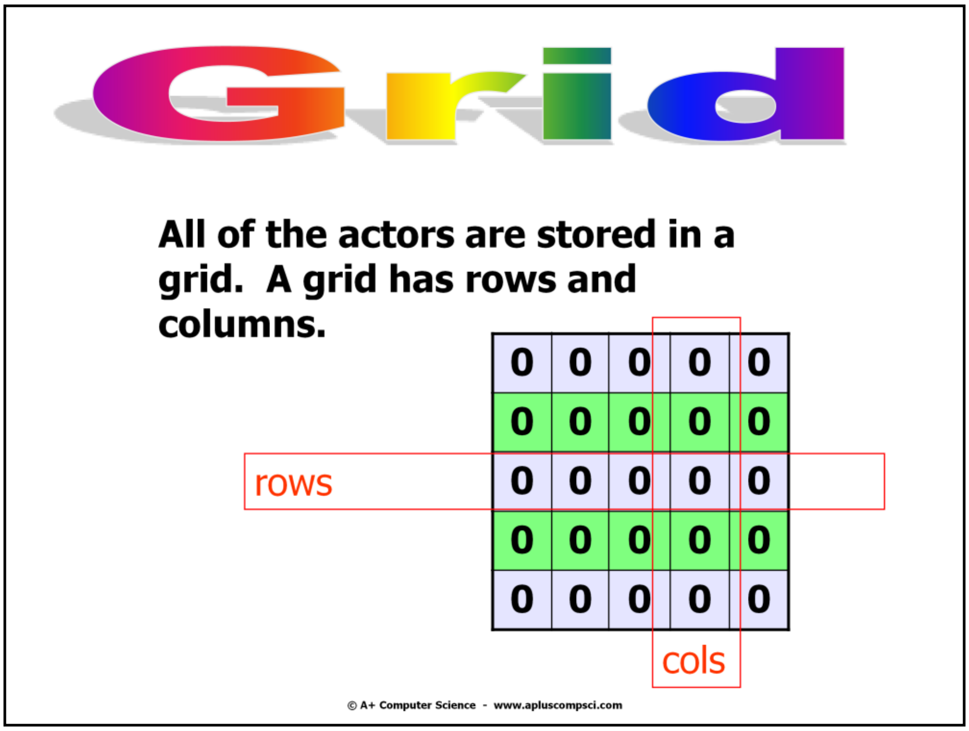
What happens when you hit the run button?

GridWorld
World Location Help
Click on a grid location to construct or manipulate an o

When placing an actor in the grid, the world add method or actor putSelfInGrid method ***<u>must</u>*** be called.

The world add method calls the actor putSelfInGrid method automatically.


open actorfour.java



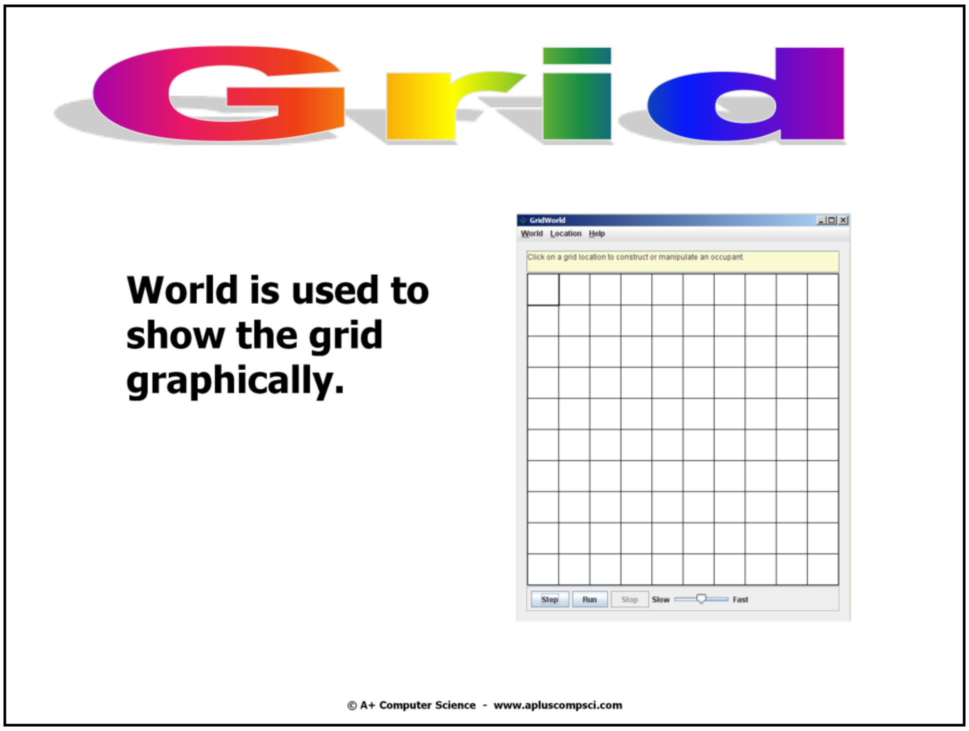
Grid

All of the actors are stored in a grid. A grid has rows and columns.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

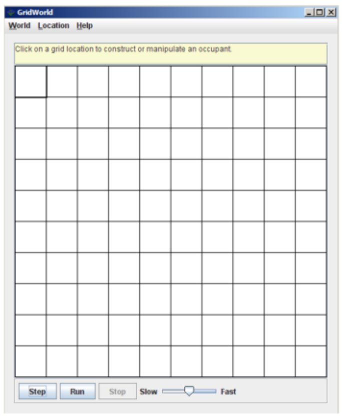
Grid is a row / column structure that stores Objects.

The location of each Object is determined by the Location provided when putting the Object in the grid.



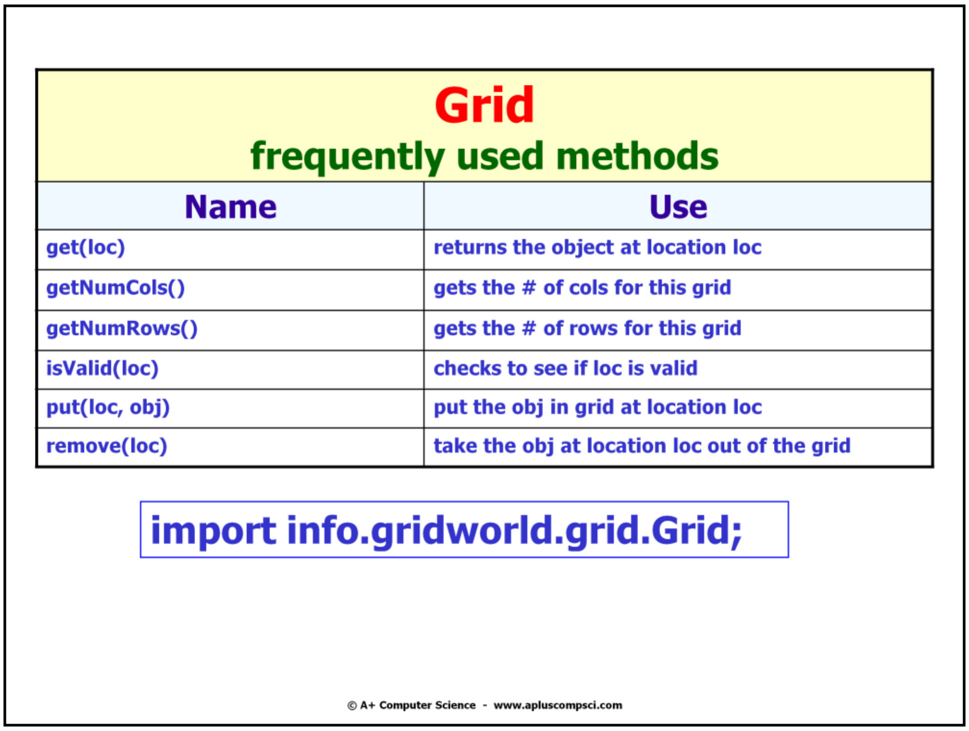
Grid

World is used to show the grid graphically.


GridWorld
World Location Help
Click on a grid location to construct or manipulate an occupant.
Step
Run
Stop
Slow
Fast

World stores the grid of Objects.

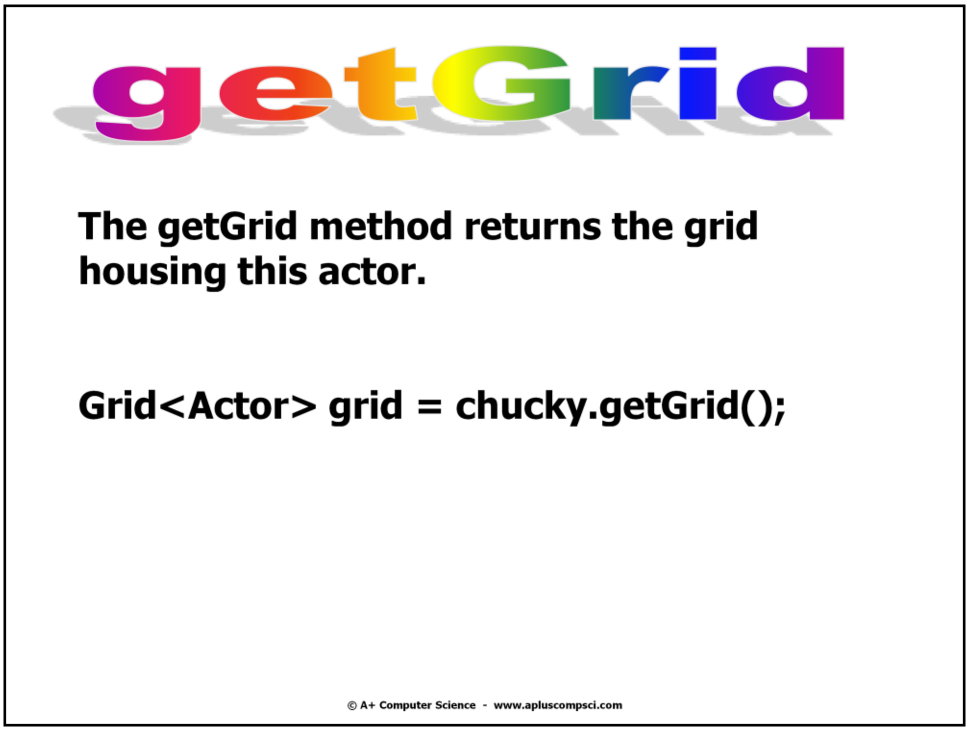
World is a graphical environment that displays the grid and its contents graphically.



Grid
frequently used methods

Name	Use
get(loc)	returns the object at location loc
getNumCols()	gets the # of cols for this grid
getNumRows()	gets the # of rows for this grid
isValid(loc)	checks to see if loc is valid
put(loc, obj)	put the obj in grid at location loc
remove(loc)	take the obj at location loc out of the grid

```
import info.gridworld.grid.Grid;
```



getGrid

The getGrid method returns the grid housing this actor.

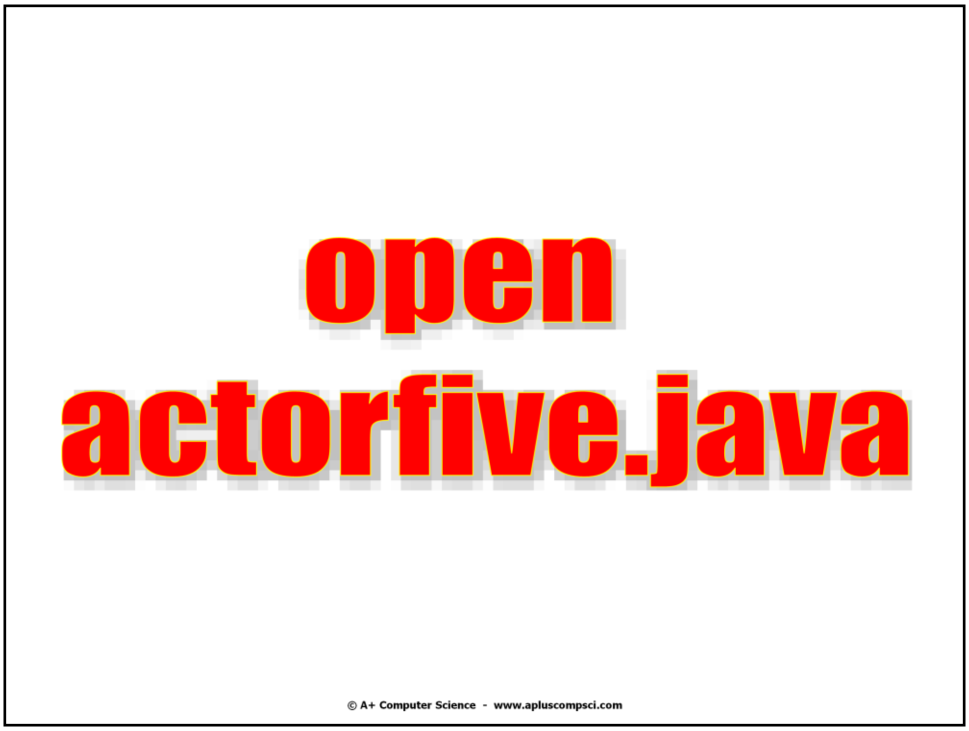
```
Grid<Actor> grid = chucky.getGrid();
```

© A+ Computer Science - www.apluscompsci.com

The getGrid method returns a reference to a grid in which this actor is stored.

If you need an actor's grid, call getGrid.

getGrid will become more useful as new types of actors are created.



open
actorfive.java

Continue work on Actor Labs