Use the [GridWorld Quick Reference](https://secure-media.collegeboard.org/apc/ap_comp_sci_a_quick_reference.pdf) if needed.Test the code out during and after each step to make sure it is error free

**\*\*\* Critters will NEVER override the ACT method \*\*\***

**Description:**

An attractive critter is a critter that processes other actors by attempting to relocate all of the other actors in the

grid, including other attractive critters. The attractive critter attempts to move each other actor one grid cell

closer to itself in the direction specified by getDirectionToward. An actor is relocated only if the location into which it would be relocated is empty. If an actor cannot be moved, it is left in its original position.

After trying to move all other actors, the attractive critter moves like a critter.

**/\*~~~~~~~~~~~~~~~~ AttractiveCritterRunner.java ~~~~~~~~~~~~~~~~\*/**

**Part 0:**

Create a new Class named AttractiveCritterRunner w/ a main method

In the main method create a new world that uses the code below.

Code to use in the main:

ActorWorld world = new ActorWorld( );

// add actors to the world here

world.show();

**\*\* Test part 0 out before moving on to part 1 \*\***

**/\*~~~~~~~~~~~~~~~~ AttractiveCritter.java ~~~~~~~~~~~~~~~~\*/**

**Part 1:**

Create a new Class named AttractiveCritter, a default AttractiveCritter is magenta.

Add a default AttractiveCritter to the center of the world. Add a Rock, a Flower, and an Actor to the world.

New Classes/Methods to use:

world.add( Location loc, Actor actor )

new Location( int row, int col )

new AttractiveCritter(), new Rock(), new Flower(), new Actor()

**\*\* Test Part 1 out before moving on to part 2 \*\***

**Part 2:**

Override the getActors method from the Critter class.

The getActors method in AttractiveCritter should return a list of all the Actors that are in the grid, except for this AttractiveCritter. Print the list before returning it, this is just to test that you are returning the correct list.

When you open GridWorld click step, read the output, make sure it only has a Rock, a Flower, and an Actor.

New Classes/Methods to use:

getGrid() /\*returns a reference to the current grid\*/

getOccupiedLocations() /\*returns a list of all the actors in the grid\*/

get( loc ) /\*returns the Actor at the location, or null\*/

**\*\* Test Part 2 out before moving on to part 3 \*\***

**Part 3:**

Override the processActors method from the Critter class. Loop through each actor in the list.

Make each of the actors in the given list move towards the AttractiveCritter.

Make sure the Location that the other actor is moving to does not contain another actor.

New Classes/Methods to use:

getLocation()

getDirectionToward( loc ) /\*returns the direction toward the given location \*/

getAdjacentLocation( dir ) /\*returns the next location in the direction given \*/

get( loc ) /\*returns the Actor, if there is one, at the given Location \*/

**\*\* Test part 3 out before turning it in \*\***