**Set 7.**

1. **What methods are implemented in Critter?**

act(), getActors(), processActors(ArrayList<Actor> actors), selectMoveLocation(ArrayList<Actor> actors), makeMove(Location loc)

1. **What are the five basic actions common to all critters when they act?**

* Getting a list of other actors
* Processing the list
* Getting locations to move to
* Selecting one of them
* Moving to selected location

1. **Should the subclasses of Critter override the getActors method? Explain.**

Yes, because they need to look elsewhere for actors to process.

1. **Describe three ways that a critter could process actors.**

* Only look at actors north and south of the critter
* Remove neighboring actors that are an instance of a specific actor, such as Rocks
* Changes the colors of neighboring actors

1. **What three methods must be invoked to make a critter move? Explain each of these methods.**

* getMoveLocations(): gets a list of possible locations for the next move
* selectMoveLocation(ArrayList<Actor> actors): selects the location for the next move
* makeMove(Location loc): moves the critter to the given location

1. **Why is there no Critter constructor?**

* Has no private variables, Java creates a constructor from the super class

**Set 8.**

1. **Why does act cause a ChameleonCritter to act differently from a Critter even though ChameleonCritter does not override act?**

The makeMove() method of the ChameleonCritter is different, and act() calls the makeMove() method.

1. **Why does the makeMove method of ChameleonCritter call super.makeMove()?**

Because after setting the direction, the makeMove() method of ChameleonCritter does the same thing that the makeMove() method of Critter does.

1. **How would you make the ChameleonCritter drop flowers in its old location when it moves?**

By adding the lines:

“Flower flower = new Flower(getColor());

flower.putSelfInGrid(grid, loc);”

to the end of the makeMove(Location loc) method.

1. **Why doesn’t ChameleonCritter override the getActors method?**

It looks for actors in the same place as Critter.

1. **Which class contains the getLocation() method?**

Actor.java

1. **How can a Critter access its own grid?**

It is a subclass of Actor and Actor has the method getGrid().

**Set 9.**

1. **Why doesn’t CrabCritter override the processActors method?.**

Because it processes actors the same as Critter.

1. **Describe the process a CrabCritter uses to find and eat other actors.**

It gets the actors in the three locations immediately in front, to its front-right, and its front-left.

1. **Does it always eat all neighboring actors? Explain.**

No, because it only looks at the actors to its front, front-right, and front-left.

1. **Why is the getLocationsInDirections() method used in CrabCritter?**

It is used to find valid adjacent locations of the critter in different directions.

1. **If a CrabCritter has location(3,4) and faces south, what are the possible locations for actors that are returned by a call to the getActors() method?**

(4, 4), (4, 3), (4, 5).

1. **What are the similarities and differences between movements of a CrabCritter and a Critter?**

CrabCritter moves like critter if it has a place to move. Otherwise, if it doesn’t move, it randomly turns left or right.

1. **How does a CrabCritter deteremine when it turns instead of moving?**

It turns and doesn’t move if there are no valid locations to turn to.

1. **Why doen’t the CrabCritter objects eat each other?**

Because it exclusively does not eat other CrabCritters.