

Activity:

A) Look at the following class file using **Geany**:

```
GridWorldCode/framework/info/gridworld/actor/Critter.java
```

Open the following file in a **web browser**:

```
GridWorldCode/javadoc/info/gridworld/actor/Critter.html
```

Read: Page 29 in the GridWorld Student Manual

Questions:

- 1) What methods are implemented in `Critter`?

- 2) What are the five basic actions common to all critters when they act?

- 3) Should subclasses of `Critter` override the `getActors` method? Explain.

- 4) Describe three ways that a critter could process actors.

- 5) What three methods must be invoked to make a critter move? Explain each of these methods.

- 6) Why is there no `Critter` constructor?

Activity:

B) Look at the following class file using **Geany**:

```
GridWorldCode/projects/critters/ChameleonCritter.java
```

Read: Pages 30 to 33 in the GridWorld Student Manual

Questions:

- 7) Why does `act` cause a `ChameleonCritter` to act differently from a `Critter` even though `ChameleonCritter` does not override `act`?

8) Why does the `makeMove` method of `ChameleonCriticter` call `super.makeMove`?

9) How would you make the `ChameleonCriticter` drop flowers in its old location when it moves?

10) Why doesn't `ChameleonCriticter` override the `getActors` method?

11) Which class contains the `getLocation` method?

12) How can a `Criticter` access its own grid?

Activity:

C) Look at the following class file using **Geany**:

```
GridWorldCode/projects/critters/CrabCriticter.java
```

Read: Pages 34 and 35 in the GridWorld Student Manual

Questions:

13) Why doesn't `CrabCriticter` override the `processActors` method?

14) Describe the process a `CrabCriticter` uses to find and eat other actors. Does it always eat all neighboring actors? Explain.

15) Why is the `getLocationsInDirections` method used in `CrabCriticter`?

16) If a `CrabCriticter` has location (3, 4) and faces south, what are the possible locations for actors that are returned by a call to the `getActors` method?

17) What are the similarities and differences between the movements of a `CrabCritic` and a `Critic`?

18) How does a `CrabCritic` determine when it turns instead of moving?

19) Why don't the `CrabCritic` objects eat each other?

Activity:

D) Create a directory called **Activity4** inside your **GridWorldCode** directory (**GridWorldCode/Activity4**). Use that directory for all your work because it will be used for grading your project. Copy or move all of the files in your **GridWorldCode/projects/critters** to the new **Activity4**. Download the bug runners for each of these from Mr Greenstein's web site.

On pages **35** and **36** in the **GridWorld Student Manual**, do the following **Exercises** in the **Activity4** directory. Refer to the manual for details.

1. Modify the `processActors` method in `ChameleonCritic`. (p. 35) Use **ChameleonRunner** that came with `GridWorld`.
2. Create a **ChameleonKid** class. (p. 35) Use **ChameleonKidRunner** from Mr Greenstein's web site.
3. Create a **RockHound** class. (p. 36) Use **RockHoundRunner** from Mr Greenstein's web site.
4. Create a **BlusterCritic** class. (p. 36) Use **BlusterCriticRunner** from Mr Greenstein's web site.
5. Create a **QuickCrab** class (p. 36) Use **QuickCrabRunner** from Mr Greenstein's web site.
6. Create a **KingCrab** class (p. 36) Use **KingCrabRunner** from Mr Greenstein's web site. Other actors must move radially outward, away from the **KingCrab**.