

CSE 143X Section Handout #8

1. Define a `Critter` class called `Cat`. It should always infect if an enemy is in front of it. Otherwise it should hop if there is a critter of another species either to its left, its right, or behind it. Otherwise it should turn right. Its color should switch between green and white and should be always display as a "C".
2. Define a `Critter` class called `Dog`. It should always infect if an enemy is in front of it. All other turns should have the following pattern: turn left five times and then hop once (e.g. your critter may turn left twice, then infect, then turn left three times, then hop). It should be colored pink and it should display how many left turns it has made since it last tried to hop (initially 0, then 1, then 2, ..., then 5, then back to 0, 1, 2, ...).
3. Define a `Critter` class called `Pigeon`. The instances of the `Pigeon` class always hop when possible and otherwise randomly choose between turning left and turning right, with each choice being equally likely. Their appearance changes over time. Each `Pigeon` initially displays as an asterisk ("*"). Then as each `Pigeon` chooses a move, it changes its appearance to match that move. If its most recent move was a hop, it displays as "H". If its most recent move was to turn left, it displays as "L". And if its most recent move was to turn right, it displays as "R". Its color should be the default color for critters.
4. Define a `Critter` class called `Orca`. The instances of the `Orca` class follow a pattern of moving forward four times, then turning around, then moving back four times and turning around again so that they return to their original position and direction. Each `Orca` is always either in moving-mode or in turning-mode. They start out in moving-mode. While in moving-mode, they try to hop forward if possible until they have hopped four times, at which point they switch into turning-mode. If it is not possible to hop while in moving-mode, an `Orca` instead infects whatever is in front of it. When in turning-mode, the `Orca` turns left twice and then switches back to moving-mode. Don't worry about the fact that if the `Orca` encounters a wall while in moving-mode, it gets stuck trying to infect the wall indefinitely. The `Orca` displays itself as "M" while in moving-mode and as "T" while in turning-mode. Its color should be the default color for critters.
5. A static variable exists once through an entire class. That means that all instances of an object can share one variable if it is static. For instance, consider if this variable is declared:

```
public static String example = "A";
```

All instances of the class will be able to refer to this variable just by using the variable name 'example'. In the case of objects, a placeholder value that can be used is null.

```
public static Point exampleTwo = null;
```

You can check for null, but you can't ask a null variable to run a method or retrieve a field. So this is alright:

```
if(exampleTwo == null) { .... }
```

Write a class called `HighSchooler` that extends the `Critter` class. The first instance `HighSchooler` should be considered the most popular `HighSchooler`. All other instances of `HighSchooler` will try to clone the most recent action of the most popular `HighSchooler`. The most popular `HighSchooler` always randomly chooses between all four actions. If the most popular `HighSchooler` is removed from the Board, then all remaining and future `HighSchoolers` will repeat the

Constant	Description
<code>Direction.NORTH</code>	facing north
<code>Direction.SOUTH</code>	facing south
<code>Direction.EAST</code>	facing east
<code>Direction.WEST</code>	facing west

most popular `HighSchooler`'s final action. If (at the beginning of a game) the most popular `HighSchooler` has not made a move yet, then `HighSchoolers` should hop. All `HighSchoolers` are Red and have an appearance of "A".