

CSE 143X Solutions to Section Handout #8

1. One possible solution appears below.

```
import java.awt.*;

public class Cat extends Critter {
    private int count;

    public Action getMove(CritterInfo info) {
        count++;
        if (info.getFront() == Neighbor.OTHER) {
            return Action.INFECT;
        } else if (info.getLeft() == Neighbor.OTHER ||
            info.getRight() == Neighbor.OTHER ||
            info.getBack() == Neighbor.OTHER) {
            return Action.HOP;
        } else {
            return Action.RIGHT;
        }
    }

    public Color getColor() {
        if (count % 2 == 0) {
            return Color.GREEN;
        } else {
            return Color.WHITE;
        }
    }

    public String toString() {
        return "C";
    }
}
```

2. One possible solution appears below.

```
import java.awt.*;

public class Dog extends Critter {
    private int spin;

    public Action getMove(CritterInfo info) {
        if (info.getFront() == Neighbor.OTHER) {
            return Action.INFECT;
        } else if (spin < 5) {
            spin++;
            return Action.LEFT;
        } else {
            spin = 0;
            return Action.HOP;
        }
    }

    public Color getColor() {
        return Color.PINK;
    }

    public String toString() {
        return "" + spin;
    }
}
```

3. One possible solution appears below.

```
import java.util.*;

public class Pigeon extends Critter {
    private String display;
    private Random r;

    public Pigeon() {
        display = "*";
        r = new Random();
    }

    public Action getMove(CritterInfo info) {
        if (info.getFront() == Neighbor.EMPTY) {
            display = "H";
            return Action.HOP;
        } else {
            int flip = r.nextInt(2);
            if (flip == 0) {
                display = "L";
                return Action.LEFT;
            } else {
                display = "R";
                return Action.RIGHT;
            }
        }
    }

    public String toString() {
        return display;
    }
}
```

4. One possible solution appears below.

```
public class Orca extends Critter {
    private int count;

    public Action getMove(CritterInfo info) {
        if (count % 6 == 4 || count % 6 == 5) {
            count++;
            return Action.LEFT;
        } else if (info.getFront() != Neighbor.EMPTY) {
            return Action.INFECT;
        } else {
            count++;
            return Action.HOP;
        }
    }

    public String toString() {
        if (count % 6 < 4) {
            return "M";
        } else {
            return "T";
        }
    }
}
```

5. One possible solution appears below.

```
import java.awt.*;

public class HighSchooler extends Critter {
    private static HighSchooler mostPopular;
    private static Action lastAction;

    public HighSchooler() {
        if(mostPopular == null) {
            mostPopular = this;
        }
    }

    public Action getMove(CritterInfo info) {
        if(mostPopular == this) {
            Random r = new Random();
            Action[] choices =
                { Action.INFECT, Action.HOP, Action.RIGHT, Action.LEFT };

            lastAction = choices[r.nextInt(choices.length)];
        }

        if(lastAction == null) {
            return Action.HOP;
        }

        return lastAction;
    }

    public Color getColor() {
        return Color.RED;
    }

    public String toString() {
        return "A";
    }
}
```