

Project 4, Critters Part 1, FAQ

1. The `make` command is in STAGE 3. How can I add Critters to the world without the `make` command for testing STAGE1 functions?

In a temporary version of `main()`, you can construct a `Critter` using a call to its constructor. You can write a temporary `public static addCritter(...)` method in the `Critter` class to initialize this new `Critter` and add it to `population`. Then test by calling other [STAGE1] commands – `step` and `show`. Before submission, you will have to comment out test features.

Remember also that during testing, you can make methods and fields in `Critter` `public`, and test your code by calling `population.get(0).walk()`, for example, from `main()`, printing out the position of the `Critter` before and after walking. This method is even more preliminary than [STAGE1], since it does not require the `step` command. Restore everything to `private/protected` etc. before submission.

2. Do I have to do the work in the order of the STAGEs? How do I use the STAGE suggestions without getting confused?

You may finish the work in any order that you like. Read the entire document twice first. We then suggest that you look through the document searching for [STAGE1], finish those parts, and then move on to [STAGE2].

3. How are we supposed to use the jar files provided to us?

Non-executable jar files are like zip files, and may be opened using `unzip`. They can also be loaded into Eclipse using `import`, and the files therein modified.

4. How am I supposed to test my files on Linux?

Before the submission date, we will provide instructions.

5. How are we supposed to access the private fields like "x_coord" and "y_coord" in `Critter.java` when we are not provided getter/setters and we are not allowed to add non-private methods?

You can access private fields from inside the class itself. That should be enough for this project. In this case, all the methods inside `Critter` have access to `x_coord` and `y_coord`. Same deal with `reproduce`.

If you want your derived Critters to access the fields, make them subclasses of `TestCritter`. Remember that the Critters you submit must be sub-classes of `Critter`, not `TestCritter`.

6. What does the static block do?

Static blocks are initialization blocks. These static blocks will be called when JVM loads the class into memory i.e. once, initially.

7. What is a protected final method?

A `final` method is one that cannot be overwritten by a subclass. A `protected` method can, however, be executed by a subclass.

`Protected` is an access level modifier, such as `public` or `private`.

| | Class | Package | Subclass (same pkg) | Subclass (diff pkg) | World |
|--------------------------|-------|---------|-------------------------|-------------------------|-------|
| <code>public</code> | + | + | + | + | + |
| <code>protected</code> | + | + | + | + | o |
| <code>no modifier</code> | + | + | + | o | o |
| <code>private</code> | + | o | o | o | o |

+ : accessible
 o : not accessible

8. If our simulator randomly spawns multiple algae to the same grid location, how should it be handled?

The confusion here is, what happens at the end of the cycle when encounters are resolved, and `fight` is called, and both algae return 0. This is fine, because even algae fight, just with a power level of zero. One of the algae will be killed (can be picked arbitrarily, doesn't matter) and the other will live on with the dead one's life.

9. What is the exact order of events of everything that happens in `worldTimeStep()`?

```
// 1. increment timestep; timestep++;
// 2. doTimeSteps();
// 3. Do the fights. doEncounters();
// 4. updateRestEnergy();
// 5. Generate Algae genAlgae();
// 6. Move babies to general population. population.addAll(babies); babies.clear();
```

10. Should rest energy be subtracted from every critter? Even the ones who moved?

Yes, rest energy should be subtracted from every critter in step 4. There is no need to keep track of which ones moved.

11. Using `Class.forName()` and `newInstance()` seems to require that we need to catch some exceptions. Should these be handled individually? How does the `InvalidCriticException` interact with these?

The exceptions the methods throw (`ClassNotFoundException`, `InstantiationException`, and `IllegalAccessException`) as all of them except the `IllegalAccessException` can be prompted by the user requesting to create a critter that does not exist. `InvalidCriticException` is one that should be thrown, not caught, and since the aforementioned exceptions will be triggered by inputting an invalid critter, an `InvalidCriticException` can be thrown from the catch block for the aforementioned Exceptions.

12. `getInstances()` in `Critter.java` is already partially filled in, but it **doesn't do everything the doc says it should do. Can we edit it?**

Yes, you should edit it. What is provided is just a skeleton, and you have to flesh it out.

13. **What exactly are the desired outputs for stage 3? Should there be an output to the screen when an `InvalidCritterException` is thrown?**

The desired outputs fall into two categories:

1) Invalid Command: Invalid Commands are commands that are not one of the following (quit, show, step, seed, make, or stats). In this instance of an invalid command, print "invalid command:" followed by text entered

2) Exception occurs during the command (regardless of exception type, including `InvalidCritterException`): Print "error processing: <command_here>" when this occurs.

For example:

"make Craig 10-" is an invalid Integer, so print out "error processing: make Craig 10-"

"make Critter main" main will cause an `InvalidCritterException`. You do not need to print out "Invalid Critter Class:" you simply need to print out "error processing: make Critter main"

14. **Overlapping critters when display world is called: Which one should be printed?**

It does not matter which one is printed, as long as something is printed.

15. `Class.forName("Craig")` **throws a `ClassNotFoundException`.** This is because the input to `Class.forName()` has to be fully qualified. So the method call should be changed to `Class.forName("assignment4.Craig");`

16. **Are multiple commands in the same line considered invalid commands?**

Yes, commands must be entered one at a time, and must be on separate lines.

17. **Should movement energy cost be subtracted even if the move is failed? For example, if the critter tries to run away during fight into a square that is already occupied?**

Yes, every time a critter tries to move, the movement cost should be subtracted.

18. **How to call a specific Critter's static `runStats` instead of calling the**

`runStats` in `Critter`?

This can be solved by using reflection. In brief, you have to get the `Class` of the `Critter` whose stats you want (say, `Craig`), create a `List` of instances of `Craig`, and then use reflection to invoke the `runStats` method of `Craig` with the already-obtained list as the parameter. We might post an addendum later giving you more details if you are still stuck.

19. How does the rounding work when a critter loses a fight?

Assign to the winner half of the loser's energy, rounded down.

20. Should a critter move to a square that is occupied by a dead critter?

This only matters in the `fight` method, when critters are trying to run away. And yes, they should be able to.

21. Should we check energy before making an action?

Only for `reproduce()`. Critters can kill themselves trying to walk.

22. What is expected for custom critters?

They don't have to be super interesting, but they have to be unique. Don't copy-paste code from `Craig` for example.

23. How to generate Javadoc for the assignment?

After adding all of your javadoc style comments, go to toolbar at top:
Project -> Generate Javadoc

Choose options appropriate for this assignment. After you finish, you should have javadoc html files in the destination folder that you specified in the options.

24. What should the output for `stats Critter` be?

This is not a valid command since `Critter` is not a concrete type of `Critter`. That is, it cannot be instantiated. Therefore the output for this command should be error processing: `stats Critter`

25. When running, can a critter run through/over another `Critter`?

Yes, critters can pass through other `Critters`.

26. What does the direction field do in `reproduce`?

It specifies the direction in which to place the child after being born, relative to its parent.

27. How can a program continue execution if an exception is thrown?

You can simply catch the exception, do something with it, and then execution will continue normally.

28. Should rest energy be subtracted from `Algae`?

Yes, rest energy should be subtracted from every `Critter`, including `Algae`.