/* CRITTERS README.pdf
 * EE422C Project 4 submission by
 * Varun Prabhu
 * vp6793
 * 15465
 * Kelby Erickson
 * kde528
 * 15495
 * Spring 2018
 */

Outside of the specified classes, we did not create any extra new classes. We edited the Critter.java and Main.java files to include all of the necessary functionality within. The 4 Critter classes that we created are named Critter1, Critter2, Critter3, and Critter4. Each Critter(number) class has methods doTimeStep, fight, and toString that all override the Critter super class' methods. Each Critter class also has distinct behavior that we have accounted for within the instance and static methods within the class. Within the Critter.java, we implemented the moving of Critters in a slightly different way than most likely intended. We implemented a helper function called walkHelper, which takes in a direction and advances the critter 1 space in the specified direction. For a critter to walk, walkHelper is called once and for it to run, walkHelper is called twice. Essentially, walkHelper is called whenever a critter is moving, regardless of the number of spaces. This made the implementation of walk and run specific restrictions much easier for us throughout the rest of the project.

As for storing the Critters, we did not change the default implementation. All of the currently alive Critters are stored in an ArrayList of Critters. If a Critter dies, they are removed from the ArrayList at the end of the time step. Additionally, any newly reproduced Critters are added to a separate ArrayList and moved into the population ArrayList at the end of the timestep (removes them from babies ArrayList as well).

On Piazza, it stated that we do not need to have a team_plan.pdf. Just in case, I will include the information here. Me and my partner have worked on the entirety of the project together excluding the 4 created Critters, which split up and individually uploaded 2 each to the project. In addition, here is the link to our github repository:

https://github.com/kelbster123/Critter-Simulator