

# Assignment 4 – Linked Structures

## Goals-

Use appropriate simple data structures

Modify an existing parent abstract class (for use by the derived classes)

You will make a program to run a tournament. Start with your assignment 3 program for fighters. A user will enter the number of fighters both players will use. The first player supplies a lineup of creatures in order. The second player then supplies their lineup of creatures in order. At the end of the tournament your program will display: the first, second, and third place finishers as well as which side scored more points.

You should be using polymorphism and all creatures should inherit from the Creature class. The users should enter the type of creature and name. Each object should be instantiated and put into their lineup. You may only statically allocate Creature pointers or structures containing creature pointers for your lineup. Ask your TA if you're unclear on static vs dynamic memory allocation.

By lineup I mean something like a batting order in baseball or softball. The head of each lineup fights. The winner gets put at the back of his team's lineup; the other goes to the loser pile. The lineup order cannot be changed once the players are entered.

For this program you must use the FIFO and FILO structures from lab 7 to hold the lineups and the loser pile. The order of the lineups only changes when a fighter is taken from the front for a bout and the winner is put back at the end. The loser pile will have the first loser on the top and the last loser on the bottom.

Even if a creature wins she may have taken damage. You should restore some percentage of the damage when they get back in line. Make this simple. Some examples are: generate a random integer in some range, or on a role of 5 they recover 50% of the damage they lost, or some scheme of your own.

At the end of the tournament you should display the first, second, and third place finishers. You should also indicate their team(s), and which team won the tournament. This is another element you must define in the analysis and design stage. I recommend doing some research on different standings systems. You must have some method for determining the total points for each team. You do not need to base it only on wins and losses. If a weaker creature beats a stronger creature maybe they get more points? ☺ Or the winner gets points equal to the total strength of the opponent? Or the creature gets points based on how many hits they inflict? In your final design document clearly state the system you are using and how you will implement it.

This is a programming assignment and not a complete game! Other than entering the lineup the players just wait for the results. They should take no further actions. Make sure you test your program with instances of all creature types you've created. You may not be able to do an exhaustive test but make it as extensive as you can.

What to submit in your zip file:

- Your files that implement your class hierarchy

- Your file(s) that implement the tournament, including the main function

- Your file(s) with your FIFO and FILO structures

- Your makefile

- Your reflections, which can include your design document, which as usual discusses the design, assumptions you needed to make, the test plan, testing results, and changes necessary as a result.

NOTE: Testing will be more involved. The reflections has more weight so you can provide a more detailed description of how you tested your program. Remember to include the results of unit testing!

## Grading:

- programming style and documentation (10%)
- create tournament, getting the fighters, resolving the round, put the 2 fighters into the appropriate containers (25%)
- prompt the user and fill the two lineups based on the number of fighters entered by the user (10%)
- properly use your FIFO structure to hold the lineups (10%)
- properly use your FILO structure to hold the loser pile (10%)
- create a heal or recovery function for the winners of each round (5%)
- create and implement a system to determine the final three finishers as well as determine which side won the tournament (10%)
- reflections document to include the design description, test plan, test results, and comments about how you resolved problems during the assignment (20%)