CSC148 Exercise 4: Lists

(Due: May 31, 10:00 pm on Markus)

A Chain of People

In this task, you will practice modelling lists, using links between nodes. W willl start with a concrete example.

Consider the situation illustrated in the picture above: some people are forming a human chain. This chain has a `leader`, who is holding onto someone, who is holding onto someone else, etc. Note that the last person in the chain isn't holding onto anyone. For simplicity, we store only two pieces of information about each person: their name, and the "next" person in the chain, i.e., who they are holding onto.

We've provided a simple model of this situation in the starter code, consisting of two classes: `PeopleChain` and `Person`. The constructor is provided for you.

Implement the `PeopleChain` methods `get_leader`, `get_second`, `get_third`, and `get_nth`. Note that `get_nth` generalizes the other three methods, meaning all three of them could be implemented with `get_nth`. You can do this if you want, but we strongly recommend that you implement the methods in the order provided, to get a good feeling of what's going on.

Finally, one implementation restriction is that you must implement `get_nth` with a for or while loop. In particular, if you know what recursion is already, don't use it here! (You'll use recursion to your heart's content soon enough, my friend.)

Hint: if you need help, think about looping through the first n people in the chain. Have a variable store a reference to the "current person"; how would you advance from the current person to the next person? Take a look at the PeopleChain constructor for a bigger hint.