Lab 6: Inheritance

The labs are getting gradually more difficult, so this fully relies on your capacities as a group, not as an individual! Work together.

Some groups have good programmers in them. Make sure that they do not simply implement everything and others in the group are just watching. Tricks to avoid this (use all!): 1) when working together, have the students who are less comfortable with the programming do the typing; 2) divide tasks and then return to the group to discuss each other's progress; 3) do not quickly implement everything before you meet as a group, but work together.

Part A: Individuals

By the end of Part A of the lab, this code should work:

```
from simulation import Simulation

sim = Simulation(persons=10, edges=10)

sim.setup()

sim.print()
sim.run(10)
sim.print()
sim.run(10)
sim.print()
```

1. Create a Person class which inherits from the Agent class. Change it so that agents have an x position, but only persons have an ε and τ variable.
2. Make sure the Agent class contains the method *calculateMu()*, while only the Person class contains *communicate()*.
3. Add an instance variable *neighbours* to the Agent class, which will be a list of agents, initially empty.
4. Add an *addNeighbour()* method to the Agent class.
5. Add a *getRandomNeighbour()* method to the Agent class.

6. In the Simulation class, add a method to randomly generate edges (ties) in the network by randomly selecting two persons and then adding each as a neighbour to the other. Make the number of edges an input parameter of the constructor of the Simulation class.
7. Update the *calculateMu()* method, such that m represents the number of neighbours.
8. Add a *setup()* method to the Simulation class and move some of the basic functionality from the constructor to this method, so that persons and edges are created in this separate method.
9. Update the *run()* method so that instead of randomly selecting two persons, it always first randomly selects one person, and then randomly selects a neighbour of that person, and then those two communicate with each other.

Part B: Media

By the end of Part B of the lab, this code should work:

```
from simulation import Simulation

sim = Simulation(persons=10, media=3, edges=10)

sim.setup()

sim.print()
sim.run(10)
sim.print()
sim.run(10)
sim.print()
```

1. Add a Medium class which inherits from the Agent class.
2. Update the constructor of the Person class by adding an instance variable η to store the horizon within which individual persons are willing to consume media.
3. Update the Simulation class so that it holds separately a list of persons and a list of media. The number of media should be an additional parameter to the constructor of the Simulation class.
4. Add a *print()* method to the Medium class. Update the *print()* method in the

Simulation class so that it prints both individuals and media.

5. Update the *setup()* method in the Simulation class to also add one medium to each individual person. Use the *addNeighbour()* method to do so.
6. Add one media consumption instance to each agent in each time step in the *run()* method. Discuss first in the group what this might look like and should do. Don't forget about η.