The purpose of this lab is to allow you to create a model of evolutionary dynamics from the ground up based on what you have learned in NetLogo. I would like you to create a model to show how cooperative behaviour can evolve when populations are not well mixed. Here are the design parameters:

1. Make your world about 50 by 50 (ie going from –25 to 25 in the x and y directions).
2. Create a population of turtles with one on each patch in the NetLogo environment, with a random mixture of cooperators (blue) and defectors (red). Since you are creating turtles on each patch rather than at random locations you will most likely want to use ask each patch to `sprout` one turtle and then randomly assign this turtle a color (or breed). I suggest you have a slider so that you can choose the proportion (or density) of initial cooperators be a variable between 0% to 100%. Hint the reporter `random-float 100` will give a number between 0 and 100, so for example if you want 35% of the new turtles to be cooperators you could ask each newly sprouted turtle to check if random-float 100 is less or equal to 35. If it is, set the turtles color to blue, otherwise set it to red. You might want to also choose different shapes for your turtles – that's up to you.
3. Inside a `go` procedure ask the turtles to play a game of prisoner's dilemma with each turtle that they find on a neighboring batch, recording its total fitness from playing all the games. There is an easy way of doing this. Lets take cost c=1 and benefit, *b*, be a slider variable ranging from 1 up to 10, in increments of 0.1. Each turtle gets *b* for every turtle on a neighbouring patch that is a cooperator. In addition, if a turtle is a cooperator it incurs a cost of 1 for every turtle on a neighbouring patch. So the fitness of a defector is *bn* and the fitness of a cooperator is *bn-m*, where *n* is the number of cooperating turtles on neighbouring patches and *m* is the total number of turtles on neighbouring patches. Note: if turtles don't move then *m* is always 8, but later in this assignment you will get turtles to move and then the number of neighbours will change.
4. After all the turtles have played their games let each turtle identify which of the turtles on the neighbouring patches has the maximum fitness (you need the primitives `max-one-of` and `turtles-on neighbors` in order to do this – so look them up). You might want to create a turtles-own variable called `fittest-neighbor`, to keep track of this turtle. All turtles should then compare their fitness to the fitness of this `fittest-neighbor` and if the `fittest-neighbor` is fitter than they are they should adopt its strategy as their own. Cautionary note: You need to do this in two steps, otherwise some turtles will change their strategy before others have had a chance to check the strategy of their `fittest-neighbor`. First have all the turtles record the strategy of the `fittest-neighbour`. (For example, if you defined a turtles-own variable such as `new-color` you ask turtles who are adopting the strategy of the fittest-neighbor to set `new-color` to the color (or breed) of the `fittest-neighbor`, otherwise if they are fitter they would set `new-color` to their own color (or breed). Then once all turtles have recorded the `new-strategy` they will adopt, in a new `ask turtles` command block get the turtles to set their color (or breed) to the new-color (or breed).
5. Plot the total number of cooperators and defectors at each time step.
6. Experiment with your model and determine the different types of behaviour it exhibits. For which value of *b* is cooperative behaviour dominant, for which values is defective behaviour dominant. For which range of values for *b* do you find an equilibrium? Put your answers in the information tab.
7. Now allow your turtles to move randomly about by wiggling and taking a step size between 0 and 2 using a slider variable. Choose step size of 0.5. Is there a value of *b>1* for which there is an equilibrium between cooperators and defectors?
8. Submit your completed file to the drop box with naming convention Lastname_Firstname_Lab_5.nlogo when you are finished.