In this lab, you will extend the example of object-oriented programming in the Week 11 lecture slides, by changing the representation of box objects, and adding a new “plus” object.

The functions from the lecture slides are available at

http://www.cs.utoronto.ca/~radford/csc121/draw.r

and the script that was used (in slightly different form) to produce the results in the lecture slides is at

http://www.cs.utoronto.ca/~radford/csc121/drawscript.r

As a first exercise, modify the functions for the “box” class to represent a box by a list whose elements are the same as the arguments of the new_box function. This may require changing new_box, draw.box, rescale.box, and translate.box. But it should not require any change to the R code that uses boxes, in the script file.

As a second exercise, try to add a new “plus” object, which looks like a plus sign. You should write a new_plus function that creates an object of class "plus", given the $x$ and $y$ coordinates of the centre of the plus (where the two lines cross) and how far the ends of the plus extend away from the centre (the same for the horizontal and vertical lines). You should also write a draw.plus function to provide a draw method. You should then be able to try drawing plusses. Once you have this working, you can write rescale.plus and translate.plus methods for objects of this class, and try them out. You should also be able to call the smaller function from the lecture with a plus object as its argument, without making any changes to smaller.

Finally, you can try to implement a new generic function for these objects, called rotate90, that rotates an object by 90 degrees counterclockwise. You may find it convenient to define a default method for this generic function (which would need to be called rotate90.default). For which classes of objects could you just let the default method be used? For which will you need to write a special method for rotate90?