

**Always** check the class message board on the blackboard site from [home.nyu.edu](http://home.nyu.edu) before doing any work on the assignment.

## Coding requirements

This is a list of computing and coding specifications. Please adhere to these for coding assignments. This list may grow during the course, so specifications are dated. You may lose points on programming exercises if you do not follow these guidelines.

(Sept. 29) Do not *hard wire* parameters or constants. If you have a formula  $u = (i/m)p$  and  $p = .3$ , do not implement it as `u = (i/m)*.3`. Instead, define a variable and set it to `.3`, as ...

```
p = .3          # red ball probability
u = (i/m)*p     # up probability
```

This makes the code easier to read. It protects against certain mistakes, particularly if  $p$  occurs more than once, as in  $v = ((m - i)/m)(1 - p)$ . It allows you to change the value of  $p$  more easily.

(Sept. 29) Comment any piece of code you add. It does not have to be as thick as the prototype codes posted, but any functionality you add or change should be commented at least a little. This applies to something as small as putting another curve in a plot figure.

(Sept. 29) When you modify a plot, also modify the legend and title to keep everything consistent. If you go from Brownian motion to Ornstein Uhlenbeck, you have to change the plot title. If you put curves from multiple times, the times must be indicated either in the title, the subtitle, or legend.

(Sept. 29) You should hand in graphs or output numbers to illustrate whatever remarks you make. If you say that curves are converging, or agree, there should be plots of enough of these curves to see that.