6.00 Handout, Lecture 23
(Not intended to make sense outside of lecture)

```
class Stock(object):
    def __init__(self, price, distribution):
        self.price = price
        self.history = [price]
        self.distribution = distribution
        self.lastChange = 0
    def setPrice(self, price):
        self.price = price
        self.history.append(price)
    def getPrice(self):
        return self.price
    def makeMove(self, mktBias, mo):
        oldPrice = self.price
        baseMove = self.distribution() + mktBias
        self.price = self.price * (1.0 + baseMove)
        self.price += mo*random.gauss(.25,.25)*self.lastChange
        if self.price < 0.01:
            self.price = 0.0
        self.history.append(self.price)
        self.lastChange = self.price - oldPrice
    def showHistory(self, figNum):
        pylab.figure(figNum)
        pylab.plot(self.history)
        pylab.title('Closing Price, Test ' + str(figNum))
        pylab.xlabel('Day')
        pylab.ylabel('Price')

def unitTestStock():
    def runSim(stks, fig, mo):
        mean = 0.0
        for s in stks:
            for d in range(numDays):
                s.makeMove(bias, mo)
            s.showHistory(fig)
            mean += s.getPrice()
        mean = mean/float(numStks)
        pylab.axhline(mean)
    numStks = 20
    numDays = 200
    stks1 = []
    stks2 = []
    bias = 0.0
    mo = 0
    for i in range(numStks):
        volatility = random.uniform(0, 0.2)
        d1 = lambda: random.uniform(-volatility, volatility)
        d2 = lambda: random.gauss(0.0, volatility/2.0)
        stks1.append(Stock(100.0, d1))
        stks2.append(Stock(100.0, d2))
    runSim(stks1, 1, mo)
    runSim(stks2, 2, mo)
```