6.00 Handout, Lecture 8
(Not intended to make sense outside of lecture)

```
def squareRootBi(x, epsilon):
    """Assumes x a non-negative float,
               epsilon a non negative float > 0
    Return y s.t. y*y is within epsilon of x"""
    assert float(x) and float(epsilon) and epsilon > 0.0
    low = 0
    high = x
    guess = (low + high)/2.0
    ctr = 1
    while abs(guess**2 - x) > epsilon and ctr <= 100:
        #print 'low:', low, 'high:', high, 'guess:', guess
        if guess**2 < x:
            low = guess
        else:
            high = guess
        guess = (low + high)/2.0
        ctr += 1
    assert ctr <= 100, 'Iteration count exceeded'
    print 'Iteration:', ctr, 'guess:', guess
    return guess

def squareRootNR(x, epsilon):
    """Assumes x a non-negative float,
               epsilon a non negative float > 0
    Return y s.t. y*y is within epsilon of x"""
    x = float(x)
    guess = x/2.0
    diff = guess**2 - x
    ctr = 1
    while abs(diff) > epsilon and ctr <= 100:
        diff = guess**2 - x
        #print 'Estimate:', guess, 'Error:', diff
        guess = guess - diff/(2.0*guess)
        ctr += 1
    assert ctr <= 100, 'Iteration count exceeded'
    print 'Iteration:', ctr, 'Estimate:', guess

def cubeRoot(x, epsilon):
    """Assumes x a float,
               epsilon a non negative float > 0
    Return y s.t. y*y*y is within epsilon of x"""
    x = float(x)
    guess = x/3.0
    diff = guess**3 - x
    ctr = 1
    while abs(diff) > epsilon and ctr <= 100:
        diff = guess**3 - x
        print 'Estimate:', guess, 'Error:', diff
        guess = guess - diff/(3.0*guess**2)
        ctr += 1
    assert ctr <= 100, 'Iteration count exceeded'
    print 'Iteration:', ctr, 'Estimate:', guess
```