6.00 Handout, Lecture 4
(Not intended to make sense outside of lecture)

```
low = 0.0
high = x
ans = (high + low)/2.0
while abs(ans**2 - x) >= epsilon:
    #print 'ans =', ans, 'low =', low, 'high =', high
    if ans**2 < x:
        low = ans
    else:
        high = ans
    ans = (high + low)/2.0
print ans, 'is close to square root of', x
----------------------------

def withinEpsilon(x, y, epsilon):
    """x, y, epsilon ints or floats. epsilon > 0.0
       returns true if x is within epsilon of y"""
    return abs(x - y) <= epsilon

def f(x):
   x = x + 1
   print 'x =', x
   return x
x = 3
z = f(x)
print 'z =', z
print 'x =', x

def f1(x):
   def g():
      x = 'abc'
   x = x + 1
   print 'x =', x
   g()
   assert False
   return x

x = 3
z = f1(x)

def isEven(i):
    """assumes i a positive int
       returns True if i is even, otherwise False"""
    return i%2 == 0

def findRoot(pwr, val, epsilon):
    """assumes pwr an int; val, epsilon floats
       pwr and epsilon > 0
       if it exists,
          returns a value within epsilon of val**pwr
          otherwise returns None"""
    assert type(pwr) == int and type(val) == float\
           and type(epsilon) == float
    assert pwr > 0 and epsilon > 0
    if isEven(pwr) and val < 0:
```

```
          return None
    low = -abs(val)
    high = max(abs(val), 1.0)
    ans = (high + low)/2.0
    while not withinEpsilon(ans**pwr, val, epsilon):
        #print 'ans =', ans, 'low =', low, 'high =', high
        if ans**pwr < val:
           low = ans
        else:
           high = ans
        ans = (high + low)/2.0
    return ans

def testFindRoot():
    """x float, epsilon float, pwr positive int"""
    for x in (-1.0, 1.0, 3456.0):
        for pwr in (1, 2, 3):
            ans = findRoot(pwr, x, 0.001)
            if ans == None:
                print 'The answer is imaginary'
            else: print ans, 'to the power', pwr,'is close to', x

sumDigits = 0
for c in str(1952):
    sumDigits += int(c)
print sumDigits

x = 100
divisors = ()
for i in range(1,x):
    if x%i == 0:
        divisors = divisors+(i,)

print divisors[0] + divisors[1]
print divisors[2:4]
```

6.00SC Introduction to Computer Science and Programming
Spring 2011