6.00 Introduction to Computer Science and Programming
Fall 2008

Lecture 7 handout
6.00 Fall Term 2008

```
import math

#Get base
inputOK = False
while not inputOK:
    base = input('Enter base: ')
    if type(base) == type(1.0): inputOK = True
    else: print('Error.  Base must be floating point number.')

#Get Height
inputOK = False
while not inputOK:
    height = input('Enter height: ')
    if type(height) == type(1.0): inputOK = True
    else: print('Error.  Height must be floating point number.')

hyp = math.sqrt(base*base + height*height)

print 'Base: '+str(base)+',height: '+str(height)+', hyp: '+
str(hyp)


def getFloat(requestMsg, errorMsg):
    inputOK = False
    while not inputOK:
        val = input(requestMsg)
        if type(val) == type(1.0): inputOK = True
        else: print(errorMsg)
    return val

base = getFloat('Enter base: ', 'Error: base must be a float')
height = getFloat('Enter height: ', 'Error: height must be a
float')

hyp = math.sqrt(base*base + height*height)

print 'Base: ' + str(base) + ',height: ' + str(height) + ', hyp:
' + str(hyp)




def exp1(a,b):
    ans = 1
    while (b>0):
        ans *= a
        b -= 1
    return ans

def exp2(a,b):
    if b == 1:
        return a
    else: return a*exp2(a,b-1)

def exp3(a,b):
```

```
    if b == 1:
        return a
    if (b%2)*2 == b:
        return exp3(a*a, b/2)
    else: return a*exp3(a,b-1)

def g(n):
   x = 0
   for i in range(n):
      for j in range(n):
         x += 1
   return x



def Towers(size,fromStack,toStack,spareStack):
    if size == 1:
        print 'Move disk from ',fromStack, 'to ',toStack
    else:
        Towers(size-1,fromStack,spareStack,toStack)
        Towers(1,fromStack,toStack,spareStack)
        Towers(size-1,spareStack,toStack,fromStack)
```