Ways to plan programs

- **STEP 0.1**: ALWAYS get ABSOLUTELY clear what behavior is SPECIFIED by the professor, your boss, customer, or your own decisions.

- **STEP 0.2**: Determine the algorithm or algorithms you will use to make the computer solve the problem from STEP 0.1
  - many beginning exercises have obvious algorithms.
  - some do not—like the histogram computation and predicting a loan payment for a given term, when the interest rate grows.
Alternatives for Planning a Programming Project after the requirements and algorithm are known clearly.

- Pseudo-Code
- Incremental Development
- Flowchart or UML Activity Diagram
  (Professionals today use a host of different Universal Modeling Language diagrams. That language is formalized. Some software translates diagrams into programs!)
Pseudo code for image processing

1. Obtain the input picture.
   - pict = input picture or use function parameter

2. Extract sizes
   - width = width(pict)
   - height = height(pict)

3. Construct new picture
   - newpict = makePicture(width, height)

4. Loop for each pixel position (x,y):
   a) Determine the RGB intensities for (x,y) in the new picture from RGB intensities in pict.
   b) Set the RGB intensities for (x,y) in the new picture to those determined from the previous step

5. Return or output newpict

The steps must be written sequentially. The sequence expresses the order in which the computer will do them!
Incremental Development

from myro import *
def filter( pict ): #Prof Chaiken wants this
    show( pict ) #Scaffolding code
from myro import *
def filter( pict ): #Prof Chaiken wants this
    W = getWidth(pict)
    H = getHeight(pict)
    print 'Width', W, 'Height' H
from myro import *
def filter( pict ): #Prof Chaiken wants this
    W = getWidth(pict)
    H = getHeight(pict)
    newpict = makePicture(W,H)
    show(newpict)
from myro import *
def filter( pict ):
    W = getWidth(pict)
    H = getHeight(pict)
    newpict = makePicture(W,H)
    tempCount = 0    #scaffolding
    for x in range(W):
        for y in range(H):
            tempCount = tempCount + 1  #scaffolding
    print tempCount, W*H
from myro import *
def filter( pict ):
    W = getWidth(pict)
    H = getHeight(pict)
    newpict = makePicture(W,H)

    for x in range(W):
        for y in range(H):
            pixel = getPixel(pict, x, y)
            R, G, B = getRGB(pixel)
            print 'Last R, G, B gotten are', R, G, B #scaffolding
from myro import *

def filter( pict ):
    W = getWidth(pict)
    H = getHeight(pict)
    newpict = makePicture(W,H)

    for x in range(W):
        for y in range(H):
            pixel = getPixel(pict, x, y)
            R, G, B = getRGB(pixel)
            setPixel(newpict, x, y, makeColor(R,G,B)
    show(newpict)  #scaffolding
from myro import *
def filter(pict):
    W = getWidth(pict)
    H = getHeight(pict)
    newpict = makePicture(W,H)

    for x in range(W):
        for y in range(H):
            pixel = getPixel(pict, x, y)
            R, G, B = getRGB(pixel)
            R = 255 - R  # insert new intensity calculation
            setPixel(newpict, x, y, makeColor(R, G, B))
    show(newpict)  # scaffolding
from myro import *

def filter( pict ):
    W = getWidth(pict)
    H = getHeight(pict)
    newpict = makePicture(W,H)
    for x in range(W):
        for y in range(H):
            pixel = getPixel(pict, x, y)
            R, G, B = getRGB(pixel)
            R = 255 - R  # insert new intensity calculation
            setPixel(newpict, x, y, makeColor(R,G,B))

    return(newpict)  # Prof Chaiken wants new picture
                     # to be returned, not be shown
                   # WE'LL BE DONE.. when we TEST INTEGRATION..