HOMEWORK #1
Due at 5 pm on Friday 10/9/15

Homework policy: Homeworks can be turned in during class or to the TA's mailbox in the Graduate Student Lounge. Late homeworks will be marked down by 20% per day. If you know that you need to turn in a homework late because of an emergency or academic travel, please let the TA know ahead of time. Collaboration is encouraged on homework assignments, however, the homework that you submit should reflect your own understanding of the material.

Readings:
AAPM/RSNA Physics Tutorial for Residents: X-Ray Production
http://radiographics.rsna.org/content/17/4/967.full.pdf

AAPM/RSNA Physics Tutorial for Residents: X-Ray Tubes
http://radiographics.rsna.org/content/17/5/1259.full.pdf

AAPM/RSNA Physics Tutorial for Residents: X-Ray Interactions
http://radiographics.rsna.org/content/18/2/457.full.pdf

AAPM/RSNA Physics Tutorial for Residents: X-Ray Attenuation
http://radiographics.rsna.org/content/18/1/151.full.pdf

Problems:

1. Explain how and why the photoelectric effect and Compton scattering depend on X-ray energy and atomic number.
2. Consider the projection radiograph of an off-axis hollow plastic cylinder. Consider the distortion in the image. Think about how the cylinder must be positioned and oriented with respect to the source and detector to cause this distortion. Sketch the position and orientation of the cylinder and explain the distortion.   (This is adapted from problem 5.17 from Prince and Links; Originally Figure 4.10 from Macovski).
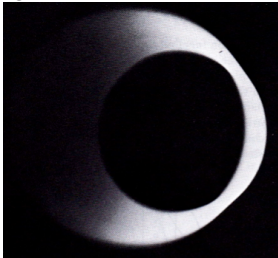
3. $h_1(x) = \exp(-x^2 / 7)$ is the point spread function (PSF) at a range $z = 5d / 8$, where d is the source-to-detector distance. Write a general expression for the PSF as a function of z. (Adapted from Prince and Links Problem 5.12).

4. Consider the object below, with attenuation coefficients of $\mu_1 = 9$, $\mu_2 = 4$, $\mu_3 = 6$, $\mu_4 = 5$ where the coefficients have units of cm^{-1}. Plot the intensity that would be detected by detectors along the x and y axes (i.e. plot two intensity profiles).

y | x

μ_1 | μ_2
μ_3 | μ_4

2 cm

4 cm
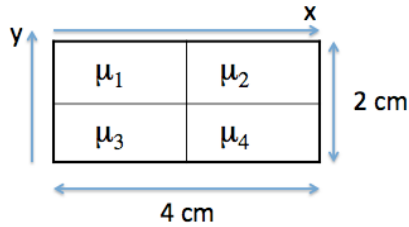
Matlab Exercise:

We will be using MATLAB throughout the course. The purpose of this exercise is to familiarize you with some of the MATLAB commands used to displaying and manipulating images. Minimize the use of **for** loops!

1. Use the MATLAB command *zeros* to create a 128x128 matrix of zeros.
2. Now fill in your matrix with 1's in the appropriate positions to create a cross pattern similar to that shown in the X-Rays/CT Lecture 1 slides.
3. Use the *imagesc* function to look at your image. Play around with the *colorbar* and *colormap* functions to see what these do. To find out how to use a function, just type help function-name (e.g. *help imagesc*).
4. Now define a 11x11 matrix that is a Gaussian function of the form
   $\exp\left(-((x - x_0)^2 + (y - y_0)^2)/w\right)$ where x_0 and y_0 denote the center of the matrix and w is a width parameter. Play around with different values of w and see what the resulting images look like.
5. Pick a value of w that will create some noticeable blurring. Use the *conv2* command to convolve your cross image with the Gaussian. Experiment with different values of w.
6. Turn in images of the cross image you created in part 2 convolved with Gaussians of different widths. Also, turn in a copy of your code.

HINTS ON USING MATLAB

When using MATLAB, it's important to keep in mind that MATLAB really likes vector and matrices. So as much as possible, you want to use vectorized operations (see below). In general, it's a good idea to minimize the use of looping structures (e.g. for loops, while loops). You will need to use some loops for the project, but try to avoid them where possible.

Here are a few examples of how to best use MATLAB.

Defining a RECT function

BAD:
```
i = 1;
for t = -10:0.1:10;
  if t < -1 | t >1
    rect(i)=0;
  else
   rect(i) = 1;
  end
i = i+1;
end
```

Good:
```
t = -10:0.1:10;
rect = ((t <= 1) & (t >=-1);
```

Defining a two-dimensional Gaussian function;

Bad:
```
i = 1 ;j = 1;
for x = -10:.1:10;
  for y = -10:.1:10;
    g(i,j) = exp(-x^2 –y^2);
    j = j + 1;
  end
 i = i + 1;
end
```

Good:
```
[x,y] = meshgrid(-10:.1:10,-10:.1:10);
g = exp(-x.^2 –y.^2);
```

You can find much more information on the web, such as:
http://web.cecs.pdx.edu/~gerry/MATLAB/programming/performance.html