The rules:

- Problem sets are due two weeks after they are released (at midnight of last day). For example, for the problem set released on Sep. 4, the deadline is 23:59 PM on Sep. 18 (CST).
- We will wait for the late submissions for 72 more hours. For example, if you submit your assignment between 00:01 AM Sep. 19 and 23:59 PM Sep. 21, it is regarded as a late submission. If you miss both deadlines, you get NO score.
- Late problem sets will be graded with an upper bound equalling the average grade of all on-time submissions. In other words, taking extra time probably won't help your grade.
- You are free to collaborate with other students, but you should submit your own solutions
- If you copy someone else's solutions, we will have a delayed talk about it at the end of the semester (i.e. you won't have time to drop the class)
- Each problem set will have an extra credit solution. It is not mandatory to complete this, but it is the only way by which you can get an A+ in the course. If you get an A+ I will be happy to write a recommendation letter for you in the future
- Unless explicitly mentioned, you are not supposed to use functions from any toolboxes. You should write all your code using build-in MATLAB functions. When in doubt ask for clarifications

The format:

In most problems in this class we will ask you to replicate figures that were shown in the class. We would strongly prefer it if you use the latest version of MATLAB (you can download an academic version from http://webstore.illinois.edu for free). Your submissions will be in terms of code which we will run upon submission. We will inspect that we get the desired figure outputs. If your code does not run, your submission will be incomplete (so to be safe submit your solutions a few days early).

Send to us all the files you need to replicate the figures. The main file to run should be called **`main.m`**. Assume that all the necessary data that we provide are in the same directory. PLEASE DO NOT ZIP YOUR FILES. Submit your .m files as they are.

You will use compass2g.illinois.edu to submit your homework. After logging into the system and clicking on the class, you'll see the assignment items in the front page. If you are not registered yet for the class and you are doing the homework, please talk to Minje on details for how to submit your homework.

Problem Set 1

You need to provide code that recreates the following plots from the lecture slides:

Lecture 1: Recreate the figure in slide 47. Use a color picture of yourself instead. In addition to removing the green channel you should flip/mirror the horizontal dimension, and swap the top with the bottom half. This needs to be done with a linear algebra operation, not with reshuffling the indexes. Note that the equation shown in that slide returns the vectorized form of the transformed image. You should use the **`reshape()`** function to transform that vector back to a tensor. You can the necessary functions from the image processing toolbox to read and potentially resize the picture.

Lecture 2: You got lucky, nothing to do here

Lecture 3: Recreate the figures in:

- Slide 18, do not use the build in spectrogram commands, write your own
- Slide 21, use matrix operations to perform the DFT
- Slide 39, feel free to use **`fir1()`** to design filters, plot using your spectrogram code. Note that in these examples the frequency axis is somewhat warped, don't replicate that part.
- Slide 46, filters have to be functionally the same, don't worry about the exact values. You need to generate the noisy versions on the right
- Slide 49, feel free to use the **`fir1()`** function, but do not use **`decimate()`** or **`resample()`**. Resample by a factor of 1/4. Just as before the frequency axis is a little warped in my slides, don't worry about replicating that.

Lecture 4: Recreate the figures in:

- Slide 50, try to generate the input data approximately
- Slide 52, don't worry about getting the exact same filters as in the figure, we just want to see that you can get peaks at the intended parts.

Data: The necessary input data is provided in the class web page. The media is titled with the corresponding slide number (e.g. the sound for slide 18 is file **`18.wav`**).

Extra credit: Describe how you could obtain the spectrogram of a vector using only a matrix product. How will that matrix look like? What are the equations that describe the whole operation?

Enjoy!